“It’s a Small World After All”: NoC Performance Optimization Via Long-Range Link Insertion

Umit Y. Ogras, Student Member, IEEE, and Radu Marculescu, Member, IEEE

Abstract—Networks-on-chip (NoCs) represent a promising solution to complex on-chip communication problems. The NoC communication architectures considered so far are based on either completely regular or fully customized topologies. In this paper, we present a methodology to automatically synthesize an architecture which is neither regular nor fully customized. Instead, the communication architecture we propose is a superposition of a few long-range links and a standard mesh network. The few application-specific long-range links we insert significantly increase the critical traffic workload at which the network transitions from a free to a congested state. This way, we can exploit the benefits offered by both complete regularity and partial topology customization. Indeed, our experimental results demonstrate a significant reduction in the average packet latency and a major improvement in the achievable network through with minimal impact on network topology.

Index Terms—Design automation, multiprocessor system-on-chip (MP-SoC), network-on-chip (NoC), performance analysis.

I. INTRODUCTION

CONTINUOUS scaling of CMOS technology makes it possible to integrate a large number of heterogeneous devices that need to communicate efficiently on a single chip. Large-scale integration of these diverse blocks calls for truly scalable communication architectures. While the legacy bus-based communication architecture is the standard solution for on-chip communication, its poor scalability, both in terms of performance and power efficiency, makes it a poor choice for future systems-on-chip (SoCs). Therefore, it has been recently proposed to replace the custom (global) wires with structured networks-on-chips (NoCs) [4], [12], [18], [21].

Regular NoC architectures based on grid-like (or two-dimensional (2-D) lattice) topologies as shown in Fig. 1 provide structured global interconnects. This ensures well-controlled electrical parameters and reduced power consumption on the global wires. However, such architectures may suffer from long packet latencies due to the lack of fast paths between remotely situated nodes. Indeed, having to traverse many hops between any two remotely communicating nodes increases the message blocking probability. This makes the message latencies unpredictable and guaranteed service operation hard to achieve. Moreover, since most of the real-life applications have widely varying communication requirements, such general-purpose platforms may become less attractive for application-specific designs that need to guarantee a certain level of performance.

On the other hand, fully customized topologies [32], [35], [38] improve the overall system performance at the expense of altering the regularity of the grid structure. This results in global wires with widely varying lengths, performance, and power consumption. Consequently, better logical connectivity comes at the expense of a penalty in the structured wiring. Hence, usual problems like crosstalk, timing closure, and wire routing may undermine the advantages expected from customization. Besides these issues, the customized topologies require specific routing algorithms, which can be difficult to implement.

Fortunately, these two extreme points in the design space (i.e., purely regular or completely customized topologies) are not the only possible solutions for NoC architectures. In fact, many technological, biological, and social networks are neither completely regular nor completely irregular [22], [41], [42]. One can view these networks as a superposition of clustered nodes with many short-range links and a few long-range links that produce shortcuts among different regions of the network. The existence of short paths between such remotely situated nodes lies at the heart of the small-world phenomenon, popularly known as six degrees of separation [30], [41]. A useful feature of these small-world networks (e.g., WWW, electrical power grid, and collaboration networks) is the logarithmic relation between the mean internode distance and network size.

Starting from this idea, this paper explores the potential of using standard mesh topologies in conjunction with a few additional long-range links, to improve the performance of NoCs. Inserting a few long-range links to the basic regular architecture clearly reduces the average distance between remotely situated nodes. Furthermore, the node/edge connectivity, hence the network reliability, is also improved. However, long-range link insertion has to be done judiciously as it has a more pronounced, yet barely studied, impact on the dynamic properties of the network characterized by traffic congestion. At low traffic loads, the average packet latency exhibits a weak dependence on the traffic injection rate. However, when the traffic injection rate exceeds a critical value, the packet delivery latency rises abruptly and the network throughput starts collapsing (see Fig. 2). The state before the congestion (i.e., the area at the left-hand side of the critical value) represents the free state, while the state beyond the critical value is the congested state. Finally, the transition from a free to the congested state is known as phase-transition region.

The emergence of congestion in mesh networks can be significantly delayed by introducing a few additional long-range links (see Fig. 2) [15]. It is important to note that, due to the abrupt
rise of the latency values beyond criticality, even a small right shift of the network critical value results in a huge reduction of the average packet latency. Similarly, the achievable network throughput grows significantly with the right shift of the critical traffic value.

This phenomenon is at the very heart of the optimization technique presented in this paper. Our main objective is to optimize the network performance (i.e., reduce the average packet latency and increase the network throughput) by maximizing the value of the critical traffic load through smart insertion of long-range links. To this end, our contribution is threefold.

• First, for a given application, we propose an algorithm that determines the long-range links that need to be inserted on a regular mesh network.
• Second, we present a deadlock-free decentralized routing algorithm that exploits the long-range links in order to achieve the desired performance level.
• Third, we evaluate the proposed approach from a complex (i.e., performance, energy and practical use) viewpoint and support our claims by experimental data obtained from a cycle accurate NoC simulator, as well as a field-programmable gate array (FPGA) prototype.

This paper is organized as follows. In Section II, we review related work and then, in Section III, propose an algorithm for long-range link insertion. The routing algorithm for the long-range links is given in Section IV. In Section V, we discuss the practical considerations related to the implementation of long-range links. Section VI discusses some energy-related issues, while Section VII outlines possible applications of the proposed technique. The experimental results appear in Section VIII. Section IX concludes the paper and indicates possible directions for future work.

II. RELATED WORK

The need for scalable communication architectures is discussed in [4], [12], [18], [21], and [36], while several concrete NoC architectures are presented in [1], [3], [5], [6], [17], [24], and [26]. Design methodologies for application-specific NoCs are addressed in [2], [19], [20], [27], [28], [32], [35], [38], and [39]. More precisely, the studies in [2], [19], and [27] consider regular network topologies and present algorithms for application mapping under different routing strategies. Topology selection for application-specific NoCs is discussed in [28]. The authors evaluate different standard topologies in terms of power, performance, and area, and select the one giving the best results. In [39], the authors analyze the impact of process technologies on network energy for 2-D meshes/torus, their higher dimensional variants, multiple hierarchies, and express channels.

The design of fully customized communication architecture, for a given application, is addressed in [32], [35], and [38]. The work in [32] is based on decomposing the communication requirements of the target application as a combination of frequently encountered communication primitives, such as gossiping and broadcasting. In [35], the authors present a constraint-driven communication architecture synthesis approach based on point-to-point communication specifications. Linear programming-based techniques for synthesizing custom NoC architectures are discussed in [38].

Inserting express channels to $k$-ary $n$-cube networks in a systematic way is discussed in [10] and [25]. These channels are used to bypass the intermediate nodes in the network. As a result, average internode distance and message latency become smaller.
To the best of our knowledge, the idea of optimizing generic network topologies with application-specific long-range links is first addressed in this paper. Previous work on links addition comes from the network theory side and is based on idealistic assumptions. For instance, the authors of [15] investigate the effect of adding random links to 2-D mesh and torus networks, under the assumption of uniform traffic. The packets in the network consist of a single atomic entity containing the address information only. Moreover, due to the infinite buffer assumption, the authors do not deal with deadlock states explicitly.

In contrast to this prior work, we consider wormhole routing and routers with bounded input buffers. Most importantly, instead of uniform traffic, we assume application-specific traffic patterns and present an algorithm which inserts the long-range links by considering the traffic patterns specificity. Due to the bounded input buffers, the additional long-range links may cause deadlock, so we also present a deadlock-free routing algorithm which exploits the long-range links in order to achieve the desired performance boost.

III. LONG-RANGE LINK INSERTION ALGORITHM

We start by formulating the application-specific long-range link inserting problem. After that, we present the solution details following a top-down approach.

A. System Model and Basic Assumptions

The system of interest consists of a set $T$ of $m \times n$ tiles interconnected by a 2-D mesh network\(^1\), as shown in Fig. 1. The tiles of the network (referred to as PEs) are populated with processing and/or storage elements that communicate with each other via the network. We do not make any assumption about the distribution of the packet injection rates, but only consider the relative rate (or frequency) at which different PEs communicate with each other.

Due to limited on-chip buffer resources and low-latency requirements, we assume wormhole switching. However, the results presented here are also applicable to packet- and virtual cut-through switching. The routing algorithm for the mesh network has to be minimal and deadlock-free. The deadlock-free property is desirable for on-chip networks since deadlock detection and recovery mechanisms are too expensive in terms of silicon resources and may lead to unpredictable delays. Hence, $XY$ routing is assumed for the mesh network.

To minimally distort the regularity of the original mesh, we limit to one the number of long-range links that can be added to any router. As such, we obtain significant performance gain with minimal modification on the initial topology. The regular routers continue to use the default $XY$ routing algorithm, while a new deadlock-free routing algorithm is proposed for the routers with extra links.

B. Problem Formulation

The communication volume between the PE located at tile $i \in T$ and the PE located at tile $j \in T$ is denoted by $V_{ij}$. We compute the frequency of communication between the PEs $i$ and $j$, $f_{ij}$, by normalizing the intertile communication volume

$$\forall i,j,p,q \in T, \quad f_{ij} = \frac{V_{ij}}{\sum_{p \neq q} V_{pq}}. \quad (1)$$

Inserting long-range links introduces an overhead due to the additional wires, extra ports in the routers, and repeaters used in the implementation of long-range links. Hence, we need to have a precise measure of this overhead.

We measure the length of the long-range links $s(l)$ in multiples of basic link units, which are identical to the regular links used in the mesh network, as shown in Fig. 9. This measure also reflects the repeater costs, since the number of repeaters required by a long-range link is given by $s(l) - 1$. For example, a resource constraint of $S$ means that the total length of the long-range links inserted to the initial network consists of at most $S$ units of standard links. Finally, the critical traffic load at which the network enters the congested phase is denoted as $\lambda_c$. Equipped with this notation, we can state now the application-specific long-range link insertion problem.

**Given**

- $f_{ij} (\forall i,j \in T)$
- Maximum number of links that can be added $S$
- The default routing strategy for the mesh network $R$

**Determine**

- The set of long-range links to be added on top of the mesh network, $L_S$
- A deadlock-free routing strategy which governs the use of the newly added long-range links such that

$$\text{max}(\lambda_c) \text{ subject to } \sum_{l \in L_S} s(l) < S \quad (2)$$

and at most one long-range link is added per router.

To give some intuition, the newly added long-range links are meant to maximize the critical traffic value $\lambda_c$, subject to the total amount of available on-chip resources. Differently stated, inserting long-range links provides increased throughput and reduced latency compared to the original critical load, as shown in Fig. 2.

In general, the objective of inserting long-range links is by no means limited to maximizing $\lambda_c$. On the contrary, other objective functions, such as increased fault-tolerance and guaranteed service, can replace (or augment) the objective of maximizing $\lambda_c$.

C. Iterative Long-Range Link Insertion Algorithm

The algorithm (see Fig. 3) starts with a standard mesh network and takes the communication frequencies $[f_{ij}]$ in (1) between the network tiles, the default routing algorithm ($R$), and the amount of resources allowed to use ($S$) as inputs. Then, the algorithm selects all possible pairs of tiles (i.e., $C(|T|, 2)$ pairs, where $|T|$ is the number of nodes in the network) and inserts
long-range links between them. After inserting each long-range link, the resulting network is evaluated to find out the gain obtained over the previous configuration. Since we seek to maximize $\lambda_c$, the gain is measured as the increase in the critical traffic load, as detailed in Section III-D.

After the most beneficial long-range link is found, the information about this link is stored and the amount of utilized resources updated. This procedure repeats until all available resources are used. Once this happens, the architecture file and routing data are generated for the new configuration.

**D. Evaluation of the Critical Traffic Value**

While the impact of the routing strategy, switching technique, and network topology on the critical point have been studied through simulation [14], no work has been aimed at maximizing the critical traffic value subject to resource constraints. The major obstacle in optimizing the critical load comes from the difficulty in modelling the variation of the critical value, as a function of different design decisions.

Several theoreticians [15], [34] propose to estimate the criticality point of a network using mean field theory models. The key idea is to reduce the estimation of the network criticality to just one parameter which can be computed analytically, much faster than simulation. This is important since using accurate estimates from simulation is simply too costly to be used in any optimization loop.

In the following, we relate the critical load $\lambda_c$ to the *free packet delay* $\tau_0$, which is the packet travel time when no other packet is present in the network. Let $\lambda(t)$ be the total packet injection rate at time $t$, i.e.,

$$\lambda(t) = \sum_{i \in T} \lambda_i(t), \lambda_i \text{ is the injection rate of tile } i \in T,$$

In the free state, i.e., when the average of $\lambda(t) < \lambda_c$, the network is in a steady state. Hence, the average packet injection rate ($\lambda$) is equal to the average packet delivery rate, that is

$$\lambda \approx \frac{N_{\text{ave}}}{\tau_{\text{ave}}},$$

where $N_{\text{ave}}$ represents the average number of packets in the network and $\tau_{\text{ave}}$ is the average time each packet spends in the network.

The exact value of $\tau_{\text{ave}}$ is a function of the traffic injection rate, as well as the network topology and routing strategy. While there is no available analytical model for calculating $\tau_{\text{ave}}$, we observe that $\tau_{\text{ave}}$ shows a weak dependence on the traffic injection rate when the network is in the free state. Hence, $\tau_0$ can be used to approximate $\tau_{\text{ave}}$. If we denote the average number of packets in the network at the onset of the criticality by $N_{\text{ave}}^{c, \text{initial}}$, we can write the following relation:

$$\lambda_c \approx \frac{N_{\text{ave}}^{c, \text{initial}}}{\tau_0}.$$  \hspace{1cm} (5)

This approximation happens to be an upper bound for the critical load $\lambda_c$, since $\tau_0 \leq \tau_{\text{ave}}(\lambda_c)$. We note that $\lambda = \frac{N_{\text{ave}}}{\tau_{\text{ave}}}$ also obeys Little’s law. Indeed, other theoretical studies approximate $\lambda_c = \frac{N_{\text{ave}}^{c, \text{initial}}}{\tau_0}$ using mean field [15] or distance models [43], again under uniform traffic assumption.

Since the number of messages in the network, at the onset of the criticality, is bounded by the network capacity $N_{\text{ave}}^{c, \text{initial}}$, the critical traffic load and the average packet latency are inversely proportional to each other. Indeed, if the average packet latency decreases, the phase transition is delayed, as demonstrated in Fig. 4, where the latency reduction is due to the presence of the long-range links. Our optimization technique uses the relationship between $\tau_c$ and $\tau_{\text{ave}}$ to maximize the critical load.

1) Experimental Verification of (4) and (5): For completeness, we verified (4) and (5) experimentally, as shown in Fig. 4. For reference, the dotted line shows the actual packet injection rate ($\lambda_c$). The solid line with the square marker on it is obtained for an $8 \times 8$ network under hotspot traffic, as the ratio between the average number of packets in the network and the average packet delay at that particular injection rate.

These plots clearly show that, before entering criticality, there is a good agreement between the actual value obtained through simulation and the one predicted by (4). As mentioned before, the exact value of the average packet delay for a given load, $\tau(\lambda)$, is found by simulation. The dashed line with triangular markers in Fig. 4 illustrates the upper bound given by (5). We observe that this expression provides a good approximation at lower data rates and holds the upper bound property.

2) Computation of $\tau_0$: For arbitrary traffic patterns characterized by the communication frequencies, $f_{ij}$ for $i, j \in T, \tau_0$ can be written as

$$\tau_0 = \sum_{i \neq j} f_{ij} \left[ d(i, j)(t_r + t_s + t_w) + \max(t_s + t_w) \left[ \frac{L}{W} \right] \right]$$

where $d(i, j)$ is the distance between router $i$ and router $j$, and $t_r, t_s$, and $t_w$ are the architectural parameters representing the time needed to make the routing decision, traverse the switch and the link, respectively [14]. Finally, $L$ is the packet length and $W$ is the width of the network channel.

For a standard mesh network, the Manhattan distance ($d_M$) is used to compute $d(i, j)$, i.e.,

$$d_M(i, j) = |i_x - j_x| + |i_y - j_y|$$

![Flow diagram showing the process of finding the most beneficial link to add.](image-url)
where subscripts \(x\) and \(y\) denote the \(x\)- and \(y\)-coordinates, respectively. For the routers with long-range links, an extended distance definition is needed in order to take the long-range connections into account. Hence, we use the following generalized definition:

\[
d(i, j) = \begin{cases} 
   d_M(i, j), & \text{no long-range link} \\
   \min (d_M(i, j), 1 + d_M(k, j)), & \text{if } I(i, k) \text{ exist.} 
\end{cases}
\]

In this equation, \(I(i, k)\) means that node \(i\) is connected to node \(k\) by a long-range link. The applicability of the distance definition is illustrated in Fig. 5. Note that the distance computation does not require any global information about the network. Hence, the routing algorithm is decentralized and its implementation is simple.

**E. Small-World Properties of Networks Customized Via Long-Range Links**

As discussed in Section III-B, the application-specific long-range links are inserted to optimize the performance of standard grid-like networks by minimally altering their structure.

The customization procedure is inspired by the small-world effect. Simply stated, the small-world networks combine the advantages of short internode distance (which is a characteristic of random graphs) and high clustering (which is primarily observed in regular graphs).

We note that we are not trying to demonstrate that the resulting network is necessarily a small-world network, in a strict sense. In fact, the small size of the networks we are dealing with and the limited number of additional links allowed during the optimization process makes such a behavior hard to observe. Moreover, by limiting the number of additional long-range links per router to just one link prevents the emergence of true hubs which are omnipresent in many small-world networks. Instead, our algorithm for inserting long-range links induces small-world effects. More precisely, our algorithm decreases the average internode distance significantly, while improving the clustering coefficient. The remaining of this section demonstrates the impact of long-range links on these properties.

1) **Impact of Long-Range Links on the Average Inter-Node Distance**: In a network with application-specific traffic characterized by \(f_{ij}\), we compute the average internode distance (\(\mu\)) as

\[
\mu = \sum_i \sum_{j \neq i} f_{ij} d(i, j)
\]

where \(f_{ij}\) and \(d(i, j)\) are given in (1) and (8), respectively.

Several theoretical studies [31], [42] assume uniform traffic, which turns out to be a special case of (9) if \(f_{ij} = 1/(n(n - 1))\forall i, j \in T\). The reduction in the average internode distance due to the long-range links is analyzed for a \(4 \times 4\) mesh network under uniform, hotspot, and multimedia (MMS) traffic. For hotspot traffic, three arbitrarily selected nodes receive extra traffic compared with the remaining nodes, while for the MMS benchmark the traffic pattern is extracted from an A/V system [19]. More information about these traffic patterns is given in Section VIII. In all cases, the long-range links were inserted with a constraint of \(S = 12\); this translates into four long-range links for the networks under study.

After inserting the long-range links, under uniform traffic, the average internode distance drops from 2.67 to 2.32, as shown in Table I. Considering that a random network with 16 nodes and mean degree 3 would have \(\mu \sim \ln(16)/\ln(3) = 2.52\), inserting long-range links indeed induces a small-world effect. For hotspot and MMS benchmarks, the improvement is larger;
this is simply because in these examples there is more room for optimization due to the skewness of the traffic patterns.

2) Impact of Long-Range Links on the Clustering Coefficient: Achieving a higher clustering compared with the random networks of the exact same size is another manifestation of small-world effect. The degree of clustering (i.e., how tightly the nodes are interconnected in a network) is measured by the clustering coefficient [42]. If the node \( i \) has \( n_i \) neighbors and there are \( L_i \) links between these neighbors, then the clustering coefficient of node \( i \), \( C_i \), can be expressed as

\[
C_i = \frac{2L_i}{n_i(n_i - 1)}. \tag{10}
\]

The clustering coefficient of the entire network (\( C_N \)) is found by averaging the clustering coefficients over all nodes. Hence, a large \( C_N \) implies that the nodes situated closer to each other are highly connected.

The clustering coefficient in a mesh topology is zero because none of the immediate neighbors of a given node are directly connected to each other. On the other hand, inserting long-range links increases the clustering coefficient of mesh networks. The impact of inserting long-range links on the clustering coefficient of a \( 4 \times 4 \) mesh network under uniform, hotspot, and MMS traffic is summarized in Table II. The increase in clustering coefficient is obtained as a by-product of the proposed link insertion algorithm, since the algorithm does not directly aim at improving the clustering coefficient. As explained in Section III-D, we insert the long-range link which improves the free packet delay the most. Another alternative would be adding the link which gives the highest performance/cost ratio. Such an algorithm obviously favors the addition of shorter links and, hence, produces a higher clustering in the network, as shown in Table II.

IV. ROUTING WITH LONG-RANGE LINKS

The routing strategy proposed in this section produces minimal paths towards the destination by utilizing the long-range links effectively. The algorithm first checks whether there exists a long-range connection to the current router, as shown in Fig. 6. If there is no such link, the default XY routing algorithm is used. Otherwise, the distance to the destination with and without the long-range link is computed using (8). Since only local information is used when computing the distance, the proposed approach is scalable and provides global improvements in the network dynamics. If the long-range link produces a shorter distance to the destination, the algorithm checks whether or not using this link may cause deadlock before accepting it as a viable route. To guarantee freedom from deadlock, some limitations on using long-range links are introduced by utilizing the turn model [16].

In the original mesh network, the links extend either along the East–West (E-W) or North–South (N-S) directions. Consequently, the basic turn model prohibits one out of four possible turns to avoid cyclic dependencies. On the other hand, the long-range links can extend in two directions, such as NE-SW or NW-SE. For example, the long-range link depicted in Fig. 5 connects two nodes with different \( x \)- and \( y \)-coordinates and extends along the NE-SW direction. As a result, using a long-range link may result in a turn from one of the middle directions NE, NW, SE, or SW to the one of the main directions N, S, E, and W. Therefore, we need to prohibit additional turns in order to avoid cycles caused by the long-range links.

In our model, we prohibit all turns from south. We also note that S-to-E and S-to-W turns for the regular links are already prohibited by the default XY routing algorithm. Finally, long-range links may introduce 180° turns. For example, the shortest path between two nodes may involve a turn from a long-range link entering the node from the east to the regular link extending towards east (i.e., a W-to-E turn). To break such cycles, 180° turns from south and west (180° turns from negative directions) are also prohibited. As a result, we provide deadlock-free routing by limiting the routing choices for the long-range links and refer to the resulting strategy as South-East routing.\(^2\) Note

\(^2\)The other possible choice that could be used with XY routing is North-East routing.
that we do not need to consider turns from a middle direction to another one, since at most one long-range link is connected to a node.

In what follows, we prove formally that the proposed routing algorithm is indeed deadlock-free. The proof can be skipped without losing the continuity of the main ideas.

**Theorem 1:** The combination of XY routing for the routers without any long-range link, and South-East routing algorithm for the routers with (at most) one long-range link is deadlock-free.

**Proof:** A routing algorithm is deadlock-free if the network channels can be enumerated such that the algorithm always routes the packets along channels with strictly increasing (or decreasing) numbers [11]. Using the notation in [16], we assign each channel in a \( m \times n \) grid a two-digit number \((a,b)\), where \( r \geq \max(3m,3n) \) and \((a,b)_r = a \times r + b\). Fig. 7(a) shows the enumeration of the channels entering an arbitrary node with coordinates \((x,y)\), where \( 0 \leq x \leq m - 1 \) and \( 0 \leq y \leq n - 1 \). The numbering scheme for a 4 \( \times \) 4 mesh network with three pairs of long-range links is illustrated in Fig. 7(b).

It can be observed that the proposed routing algorithm forwards the packets only to the channels with strictly increasing ordering. To prove that this is indeed the case for all packets, we analyze each possible input to an arbitrary node in Fig. 8(a)–(h) using the numbering scheme introduced in Fig. 7(a). For instance, Fig. 8(a) shows a long-range link input from the NW direction. The outgoing long-range link in the opposing direction can connect to node \((x-k_x,y+k_y)\), where \( 1 \leq k_x \leq x \) and \( 1 \leq k_y \leq n - 1 - y \). Investigating the channel numbers reveals that the channels that do not result in strictly increasing ordering are prohibited by the proposed algorithm.

In general, a long-range link originating from node \((x,y)\) can be connected to a node \((x \pm k_x,y \pm k_y)\), where \( 1 \leq k_x \leq x \) when the long-range link extends to the negative \( x \) direction and \( 1 \leq k_y \leq m - 1 - x \) when it extends to the positive \( x \) dimension. Likewise, \( 1 \leq k_y \leq y \) when the long-range link extends to the negative \( y \) direction and \( 1 \leq k_x \leq n - 1 - y \) when it extends to the positive \( y \) dimension.

Fig. 8(a)–(h) shows that the proposed algorithm routes the packets to channels with increasing numbers for all possible inputs. Of particular interest are Fig. 8(a), (b), and (h). These figures show that the turns that do not result in a strictly increasing channel ordering are the turns from the south, which are prohibited by the proposed algorithm.

We have shown that the proposed algorithm always routes the packets along the channels with strictly increasing numbers. As a result, it is deadlock-free.

The final thing to consider is the possibility of a long-range link acting as a traffic attractor and then becoming a bottleneck in the network. For this reason, we assess the amount of traffic already assigned to a long-range link and route further traffic over the link only if it is not likely to become a bottleneck. This method is static and does not need feedback from the network. An alternative approach would be to monitor the congestion level on the long-range link and the downstream router and then route the packets adaptively using the permissible turns depicted in Fig. 8. This is left for future work.

**V. Implementation of Long-Range Links**

In order to preserve the advantages of structured wiring, the long-range links are segmented into regular, fixed-length, network links connected by repeaters. The repeaters can be thought of as simplified routers consisting of only two ports that accept an incoming flit, store it into a first-in first-out (FIFO) buffer, and finally forward it to the output port, as illustrated in Fig. 9. The repeaters essentially pipeline the long-range links and produce routes which bypass the routers, while looking identical to the original paths provided by the routers. The use of repeaters with at least two-flit buffering capabilities guarantees latency-insensitive operation, as discussed in [7] and [8].

Another consideration is the increase in the size of the routers with extra links due to the additional port. This overhead has to be taken into account while computing the maximum number of long-range links which can be added to the regular mesh network. Although there is no theoretical limitation on the number of additional links a router can have, a maximum of one long-range link per router is used in our approach. This way, the regularity of the mesh network is minimally altered, while still enjoying significant improvements over the standard mesh network.

**VI. Energy-Related Considerations**

Here, we investigate the proposed approach from an energy consumption point of view. One can measure the energy consumption using the \( E_{\text{sh}} \) metric [44], defined as the energy re-
is given by

$$E_{\text{link}} = E_{L\text{link}} + E_{B\text{link}} + E_{S\text{link}}$$  \hspace{1cm} (11)$$

where $E_{L\text{link}}$, $E_{B\text{link}}$ and $E_{S\text{link}}$ represent the energy consumed by the link, buffer, and switch in the router, respectively.

The total energy consumption before inserting the long-range links can be expressed as

$$E_M = \sum_{i,j} V_{ij} [d_M(i,j)E_{L\text{link}} + (d_M(i,j)+1)E_{B\text{link}}]$$

$$+ \sum_{i,j} V_{ij} \sum_r E_{S\text{link}}(r)$$  \hspace{1cm} (12)$$

where $V_{ij}$ and $d_M(i,j)$ are the communication volume and Manhattan distance between nodes $i$ and $j$, respectively. The switching energy is summed up over all of the routers the message goes through but written separately to emphasize the difference in router sizes. For example, due to the increased router arity, the switch at a router with five I/O ports has a larger energy consumption compared with the switch involving only three I/O ports.

Using a similar approach, the total energy consumption after the insertion of long-range links can be expressed as

$$E_{ML} = \sum_{i,j} V_{ij} [d_M(i,j)E_{L\text{link}} + (d_M(i,j)+1)E_{B\text{link}}]$$

$$+ \sum_{i,j} V_{ij} \sum_r E_{S\text{link}}(r)$$

where $E_{S\text{link}}(r)$ represents the switching energy after the insertion of the long-range links. We note that the energy consumed in the links remains the same whether or not the message uses the long-range links, since the total number of links traversed remains the same. The same argument is also valid for the buffer energy consumption, assuming that similar buffers are used in repeaters and routers.

On the other hand, the switch energy consumption is affected, since some messages are eventually routed through the routers with extra links, while some others will be routed through repeaters and thus bypass several routers. In fact, the latter scenario provides a reduction in the communication energy consumption due to the elimination of the crossbar switch, while the former induces a penalty due to increasing size of the router. Overall, we expect a small impact on the energy consumption.

We evaluated the energy consumption before and after inserting long-range links using a simulator and an FPGA prototype. We observe that the link and buffer energy consumption increases by about 2% after inserting long-range links, while the switch energy consumption drops by about 7%, on average. Furthermore, the Orion model [40] integrated to our simulator.
VII. PRACTICAL USE OF LONG-RANGE LINKS

Inserting application-specific long-range links enables a higher network throughput and a lower average packet latency compared to a pure mesh architecture. As a result, the application-specific long-range links can be employed in several scenarios.

- First, when the application mapping is given or there are tight constraints on mapping the application to the network, inserting long-range links can greatly enhance the performance of the regular mesh architecture. The proposed technique works also well in conjunction with an already existing mapping algorithm (e.g., [19]) due to the moderate run-time requirements. For example, the long-range link addition algorithm can be invoked after a permissible mapping is obtained to see how much additional improvement can be obtained.

- The long-range links can also be exploited to achieve fault-tolerance or quality-of-service (QoS) operation. For example, the use of a long-range link can be limited to a few connections which provide guaranteed latency (or throughput) for a selected set of nodes. Furthermore, the long-range links can also support multiple use-case scenarios [29]. For example, a long-range link can provide guaranteed service for some use cases, while serving the best-effort traffic for improved performance during other use-cases.

- The improvement in network performance can be exploited to optimize the system power consumption. For example, the operating voltage can be scaled down while still achieving the same throughput of a pure mesh network of exact same size. As a result, the overall power consumption can be minimized.

- Finally, the long-range links can be considered in a reconfigurable context to obtain a common architecture which can be optimized for a larger class of applications.

VIII. EXPERIMENTAL RESULTS

Next, we present an extensive experimental study involving synthetic and real traffic patterns. For each benchmark, the standard mesh network and the mesh network with a small number of long-range links inserted to it are compared. Sections VIII-A and VIII-B present simulation results obtained using an in-house cycle-accurate (C++-based) NoC simulator developed specifically for this project. The simulator models the long-range links as explained in Section V. In Section VIII-C, we present our results obtained using an FPGA prototype.

The worst-case complexity of the technique (i.e., the link insertion and the routing table generation) is $O(SN^2)$, with $2 < \alpha < 3$. The run-time of the algorithm for the examples we analyzed ranges from 0.14 s for a 4 × 4 network, to less than half hour for an 8 × 8 network, on a Pentium III machine with 768-MB memory under Linux OS.

### TABLE III

<table>
<thead>
<tr>
<th>Workloads</th>
<th>Critical load (packet/cycle)</th>
<th>Latency at the critical load (cycle)</th>
</tr>
</thead>
<tbody>
<tr>
<td>hotspot4</td>
<td>0.41</td>
<td>196.9</td>
</tr>
<tr>
<td>hotspot6</td>
<td>0.62</td>
<td>224.5</td>
</tr>
<tr>
<td>transpose4</td>
<td>0.35</td>
<td>89.3</td>
</tr>
<tr>
<td>transpose6</td>
<td>0.54</td>
<td>165.9</td>
</tr>
</tbody>
</table>

A. Experiments Involving Synthetic Traffic Workloads

We first demonstrate the effectiveness of adding long-range links to standard mesh networks by using the hotspot and transpose traffic inputs. For hotspot traffic, three nodes are selected randomly to act as hotspot nodes. Each node in the network sends packets to these hotspot nodes with a higher probability compared with the remaining nodes. For transpose traffic, on the other hand, each node communicates only with the symmetric node with respect to the diagonal of the network. The critical traffic load values for some 4 × 4 and 6 × 6 mesh networks, under hotspot and transpose traffic patterns, are given in Table III. We observe that inserting four long-range links to a 4 × 4 network (the resulting network is shown in Fig. 1) under hotspot traffic makes the phase transition region shift from 0.41 packet/cycle to 0.50 packet/cycle. Similarly, due to the addition of long-range links, the average packet latency at 0.41-packet/cycle injection rate drops from 196.9 to 34.4 cycles!

The variation of the network throughput and average packet latency as a function of traffic injection rate are plotted for hotspot and transpose traffic in Figs. 10 and 11, respectively. For transpose traffic, the phase transition region shifts from a throughput of 0.35 packet/cycle to 0.52 packet/cycle. Given the changes in the overall network dynamics, this is a huge improvement in network capabilities. Likewise, with the addition of long-range links, the average packet latency at 0.35-packet/cycle injection rate drops from 89.3 to 25.2 cycles.

It is interesting to observe that the improvement obtained for a 6 × 6 network in Table III, under transpose traffic, is smaller compared with other cases. The primary reason for this behavior is that some of the newly added long-range links act as fact as traffic attractors. Since the proposed routing algorithm is not adaptive, i.e., it does not consider the congestion in the channel links at run time, these links eventually become the bottleneck in the network. This observation suggests that developing an adaptive algorithm, at the expense of increased resources, has the potential to improve the impact of long-range link insertion algorithm even beyond the results presented here.

1) Scalability Analysis: To evaluate the scalability of the proposed technique, we performed several experiments with network sizes ranging from 4 × 4 to 10 × 10. For the 4 × 4 and 6 × 6 networks, four bidirectional links are inserted to the standard mesh configuration. Similarly, for the 8 × 8 and 10 × 10 networks, the IDs of the hotspot nodes are: 5, 11, and 12. For example, for the 4 × 4 network, this translates to (1,1), (3,2), (0,3), as shown in Fig. 5.
networks, five and six bidirectional links are inserted, respectively. Fig. 12 shows that the proposed technique results in consistent improvements when the network size scales up. For example, the critical load of a $10 \times 10$ network, under hotspot traffic, shifts from 1.18 packet/cycle to 1.40 packet/cycle after inserting only six bidirectional long-range links consisting of 32 regular bidirectional links total. This result is similar to the improvements obtained for smaller networks. Fig. 12(a) also reveals that the critical traffic load grows with the network size due to the increase in the total bandwidth. Likewise, we note substantial reductions in the average packet latency across different network sizes after inserting long-range links, as shown in Fig. 12(b).

2) **Comparison With Topologies of Higher Dimensionality:** Finally, we note that customizing 2-D mesh networks with long-range links is a more general approach than choosing tori or other network topologies of higher dimensionality or simply inserting express links based on a fixed rule [10], [25]. Indeed, the on-chip implementation of these networks looks similar to the implementation of a 2-D 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 wiring rule. In fact, the application-specific customization with long-range links will generate standard higher dimensional networks (or reduce to inserting by-pass links), if we replace the link insertion algorithm with a static link insertion rule. As a result, our proposed technique can achieve a better performance compared to a higher dimensional network, although it utilizes about the same or less resources.

To be more concrete, we implemented a $4 \times 4$ 2-D torus network with folded links [12] and a mesh network with 8 unidirectional links generated by our proposed technique for a resource constraint threshold of $S = 12$. Our simulations show that the critical traffic load of the network customized using our proposed technique is 4% larger than that of the torus network. Moreover, the average packet latency in our design, at 0.48-packet/cycle injection rate (which is close to the critical load of the torus network), is only 34.4 cycles compared with 77.0 cycles for the torus network. This significant gain is obtained by utilizing only half of the extra links needed by the torus; indeed, inserting the most beneficial links for a given traffic pattern makes more sense than blindly adding channels following a fixed rules, which is precisely the case for the folded torus.

**B. Experiments Involving Real Traffic Loads**

Here, we evaluate the performance of the link insertion algorithm using three realistic applications. The first two applications, *auto industry* and *telecom* benchmarks, are retrieved from the E3S benchmark suite [13] and mapped onto $4 \times 4$ and $5 \times 5$ networks, respectively, using an automated mapping tool. The third application is a multimedia application (MMS) which includes an H263 video encoder/decoder, an MP3 audio encoder/decoder pairs. This application is first partitioned into 40 concurrent tasks and then assigned and scheduled onto 16 IPs.
Fig. 12 The improvement in the (a) critical traffic load and (b) average packet latency for increasing network sizes.

![Graph showing improvement in critical traffic load and average packet latency](image)

Connected in a $4 \times 4$ mesh network [19]. The long-range links are inserted with a constraint of $S = 12$ and $S = 20$ for the $4 \times 4$ and $5 \times 5$ network, respectively.

The variation of average packet latency and network throughput as a function of traffic injection rates for the auto industry benchmark is given in Fig. 13. These plots show that the insertion of long-range links shifts the critical traffic load from 0.29 to 0.33 packet/cycle (about 13.6% improvement). Similarly, the average packet latency for the network with long-range links is consistently smaller compared to that of a pure mesh network. For instance, at 0.29-packet/cycle injection rate, the latency drops from 98.0 to 30.3 cycles, giving about 69.0% reduction.

![Graphs showing traffic injection rate versus average packet latency and network throughput for auto industry benchmark](image)

Similar improvements have been observed for the telecom benchmark, as shown in Fig. 14. Specifically, the critical traffic load is improved from 0.44 to 0.60 packet/cycle, showing a 36.3% increase due to the insertion of long-range links. Likewise, the latency at 0.44-packet/cycle traffic injection rate drops from 73.1 to 28.2 cycles. Finally, a pure $4 \times 4$ mesh network running the MMS application has a critical traffic load of 0.26 packets/cycle, while the network customized using application-specific long-range links has a traffic load of 0.29 packets/cycle. Moreover, the average packet latencies at 0.26 packets/cycle are 96.0 and 31.0 cycles for the original mesh and customized networks, respectively.

Implementing long-range links requires inserting buffers in the repeaters. In order to demonstrate that the savings are
primarily coming from using the long-range links, we also added extra buffers to the channels of the pure mesh network with inserted links, equal to the amount of buffers utilized for the long-range links. Table IV summarizes the results for standard mesh network (M), standard mesh network with extra buffers (MB), and the network with long-range links (L). We observe that the buffer insertion improves the critical load by 3.5% for the auto industry benchmark. On the other hand, the corresponding improvement due to long-range links is 13.6% over initial mesh network and 10% over the mesh network with additional buffers. Likewise, we note that, due to inserting long-range links, the average packet latency reduces by 69% compared with the original latency value and 57.0% compared to the mesh network with extra buffers.

Consistent results have been obtained for the synthetic traffic workloads mentioned in the previous section and for the telecom benchmark. Due to the limited space, we report only the results for the telecom benchmark since this reflects a real application. The results in Table IV show that, with the addition of extra buffers, the critical traffic point shifts only from 0.44 to 0.46 packet/cycle. Inserting long-range links, on the other hand, shifts the critical point to 0.60 packet/cycle, which represents a huge improvement in the network throughput capability. Similarly, the average packet latency obtained by the proposed technique is almost one third of the latency provided by standard mesh and about one-half of the latency provided by mesh with extra buffers.

### C. FPGA Prototyping

To further demonstrate the effectiveness of our proposed methodology, we present an FPGA prototype using a Xilinx Virtex-II XC2V4000 device. First, a 4 × 4 2-D mesh network which works under wormhole routing is implemented using Verilog HDL. The packets in the network are divided into 16-bit flits, since the width of the channels is 16 bits. It takes four cycles for the router to route the header flit, and then the remaining flits follow the header in a pipelined fashion. The packet size in terms of number of flits is also parameterized. For flexibility reasons, the routing strategy is implemented as a lookup-table and fed to the network before configuring the FPGA. Finally, the routers in the network have 16 × 16 output buffers at each port, i.e., the depth of the FIFO buffers is 16.

The area of routers with different number of ports is summarized in Table V. 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. The increase in the area is mainly due to the FIFO buffer of the additional port and the increased size of the crossbar switch. In addition to the individual routers, we implemented a 4 × 4 mesh network and a mesh network customized using long-range links of total length 40 for auto industry, hotspot, and transpose traffic patterns discussed in Sections VIII-A and VIII-B. The customized network with four long-range links (consisting of 12 regular link segments) utilizes 7152 slices, which is 469 slices larger than the original mesh network, as shown in Table V.

We observe that the improvements in the average message latency and network throughput measured using the FPGA prototype are consistent with the simulation results. The latency comparison under transpose traffic is plotted in Fig. 15, while further experimental results can be found in [33]. This basically validates our simulation results and offers a solid basis for the newly proposed approach.

We also performed accurate energy measurements on the FPGA prototype using the cycle-accurate power measurement tool developed by Lee et al. [23]. The experiments have shown minimal impact on the energy consumption and validated the theoretical expectations and simulation results in Section VI [33].
IX. CONCLUSION AND FUTURE WORK

In this paper, we have presented a novel design methodology for inserting application-specific long-range links to standard mesh NoC architecture. It has been analytically and experimentally demonstrated that inserting long-range links has an important impact on the dynamic, as well as static properties of the network.

As the main theoretical contribution, we have shown that additional long-range links increase significantly the critical traffic workload. We have also demonstrated that this increase brings a significant reduction in the average packet latency of the network, as well as substantial improvements in the achievable throughput.

We plan to extend the present work in several directions. Given the impressive improvements in the packet latency, there is enough slack to be exploited, so we can use DVS strategies in conjunction with this work [37]. Also, while our current work employs oblivious routing to utilize the long-range links, we can extend this formulation to employ adaptive routing instead. One can also consider removing some of the links with low utilization, as well as inserting long-range links, to minimize the implementation cost and penalty on the regularity. Other possible extensions include inserting long-range links for different objective functions such as fault-tolerance and QoS operation.

ACKNOWLEDGMENT

The authors would like to thank H. G. Lee of Seoul National University for his contribution to the prototyping work. They would also like to acknowledge the feedback and suggestions they received from the editors and reviewers.

REFERENCES


Umit Y. Ogras (S’00) received the B.S. degree from Middle East Technical University, Ankara, Turkey, in 2000, and the M.S. degree from The Ohio State University, Columbus, in 2002, both in electrical engineering. He is currently working toward the Ph.D. degree in the Department of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, PA. His research focuses on communication-centric design methodologies for nanoscale systems-on-chip, with a special interest on network-on-chip communication architectures.

Radu Marculescu (S’94–M’98) received the Ph.D. degree in electrical engineering from the University of Southern California, Los Angeles, in 1998. He is currently an Associate Professor with the Department of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, PA. His current research focuses on developing design methodologies and software tools for system-on-chip design, on-chip communication, and ambient intelligence.

Dr. Marculescu is a member of the Association for Computing Machinery. He was a recipient of the National Science Foundation’s CAREER Award (2001) in the area of design automation of electronic systems. He received the 2005 Transactions on Very Large Scale Integration Systems (T-VLSI) Best Paper Award from the IEEE Circuits and Systems Society, two Best Paper Awards from the Design Automation and Test in Europe (DATE) Conference in 2001 and 2003, and a Best Paper Award from the Asia and South Pacific Design Automation Conference in 2003. He was also awarded the Carnegie Institute of Technology’s Ladd Research Award in 2002.