Nano Communication Networks 4 (2013) 23–42

Contents lists available at SciVerse ScienceDirect

Nano Communication Networks

journal homepage: www.elsevier.com/locate/nanocomnet

Design and analysis of heterogeneous nanoscale on-chip communication networks

Haera Chung *, Christof Teuscher *

Teuscher Lab, Department of Electrical and Computer Engineering, Portland State University, Portland, OR 97207, United States

ARTICLE INFO

Article history:
Received 26 September 2012
Received in revised form 7 December 2012
Accepted 13 December 2012
Available online 22 January 2013

Keywords:
Networks-on-chip
NoC
Heterogeneous

ABSTRACT

With a continuing downscaling of the physical feature sizes, an increase in complexity and the number of on-chip devices, and an increase in their heterogeneity, the traditional on-chip communication infrastructure needs to be revisited. It has previously been shown that multi-hop communication suffers from high latency and power consumption and that networks with long-range, high-bandwidth, and low power communication links significantly improve the system performance. Yet, it is an open problem what level of heterogeneity and what link type characteristics represent an optimum for on-chip communication networks. In this paper we design and analyze optimal heterogeneous networks by considering different cost and performance trade-offs in a technology-agnostic framework. We show that there is an optimal number of different link types for each set of constraints and that the heterogeneous network performance allows for a higher throughput at a lower cost compared to 2D regular mesh and homogeneous networks. From our results it follows that the link types available with current technology are non-optimal in a heterogeneous setup. We show that the main results are robust against certain model assumptions. In addition, the proposed heterogeneous networks scale up significantly better in terms of both cost and performance. The results are relevant for the design of emerging nanoscale communication fabrics and will help to drive the development of new technology.

1. Introduction

As the number of cores (or IP blocks) integrated on a single chip increases, the communication between them becomes increasingly important. Traditional *Systems-on-Chips* (SoCs) interconnect architectures are based on a shared bus structure, which can carry only one communication transaction at a time. This limits the communication bandwidth and scalability [22]. Such an architecture is therefore not suitable for future nanoscale SoCs, which may have orders of magnitude more components. In recent years, *Network-on-Chip* (NoC) were proposed as a promising solution for designing large and complex on-chip communication problems [2,19]. The NoC paradigm provides better scalability and reusability for future SoCs. Despite these benefits, conventional metal wire interconnects limit the communication performance of NoCs because of their multi-hop nature for long-range on-chip communication. Multi-hop communication causes high latency and power dissipation [15], which can lead to the interconnect consuming 80% of the total chip power [16]. The ITRS roadmap also states that "[i]t is now widely conceded that technology alone cannot solve the on-chip global interconnect problem with current design methodologies".

To solve this problem, we need new interconnect fabrics that can support single-hop communication across an entire chip. In the last few years, several new interconnect technologies, such as photonic interconnects [29], multi-band RF NoC [5], Carbon NanoTubes (CNTs) [9,18], and millimeter wave wireless (mmWave) [11] were proposed and evaluated. However, these new technologies were evaluated and used with specific performance metrics, such as

* Corresponding authors.
E-mail addresses: hrchung@ece.pdx.edu (H. Chung), teuscher@pdx.edu (C. Teuscher).

1878-7789/$ – see front matter © 2013 Elsevier Ltd. All rights reserved.
doi:10.1016/j.nancom.2012.12.002

throughput, latency, or power, while in reality, trade-offs are the key. What may be optimal for one metric may not be optimal for all the others. For example, one can obtain high performance by adding more wires to a NoC, but this will increase the wiring cost and power consumption. How to use these new NoC interconnect technologies in an optimal way is an open problem [25].

In this paper, we propose to design and analyze optimal nanoscale NoCs with a much larger link library to answer the following questions:

- How many heterogeneous link types are optimal?
- Is that optimum different from the number of link types allowed by current technology?
- What are an optimal heterogeneous link type distributions for a given traffic scenario?
- What is an optimal placement of the different link types for a given traffic scenario?
- Do heterogeneous link type networks scale better than mesh networks?

We answered these questions by developing a comprehensive software framework for the design, optimization, and evaluation of complex heterogeneous nanoscale NoC networks. The framework can optimize networks according to any number and combination of the common network performance metrics, such as the wiring cost, the average shortest path length (latency), the throughput, and the energy. Once an optimal heterogeneous NoC architecture has been obtained, we evaluated the network under different realistic traffic models and showed that our proposed heterogeneous architectures outperform homogeneous architectures in performance, energy, and throughput. We compared the results with heterogeneous link type networks with less than ten different link types and showed that they do not represent an optimal solution. In addition, we also demonstrated that our proposed interconnect architectures scale better than regular 2D mesh networks.

The main contributions of this work are as following:

(1) We introduce a library of heterogeneous link types in nanoscale NoCs to solve the current NoC multi-hop wired communication problems and significantly improve the network performance at a low cost. To the best of our knowledge, no one has thoroughly evaluated hybrid networks with three or more different kinds of interconnect technologies in a comprehensive framework that can deal with several design constraints.
(2) We present algorithmic methods to evaluate each different type of link and find the best solutions, such as the optimal number of heterogeneous link types, the optimal wire-length distributions, and an optimal placement of the heterogeneous links. This allows us to obtain optimal networks for a broad range of current and future nanoscale NoC interconnect technologies.
(3) We present evolutionary optimization techniques to obtain optimal NoC topologies. Our networks significantly outperform current three link type networks, homogeneous, and regular mesh networks.

The results presented in this paper are relevant for a broad set of nanoscale NoC and SoC applications, which rely increasingly on different communication channels. Note that we do not make any statement whether our ten link types can actually be implemented in some technology at this point in time. The current study takes a technology-agnostic approach and investigates heterogeneous networks from an abstract perspective in order to obtain fundamental results that are broadly applicable.

2. Related work

Traditional NoC architectures are based on packet-switching networks. In the last few years, several solutions were proposed to improve the network performance. Ogras and Marculescu [26] have proposed inserting a few long-range links to standard mesh NoC topologies to improve the performance of NoCs. The results show that adding long-range links reduces the average distance between source and destination nodes, which increases the network throughput and reduces the average packet latency. The authors did not consider cost and scalability in their paper.

Teuscher [31,32] showed that unstructured nature-inspired NoCs can have benefits for performance and scalability over traditional structured NoCs.

Photonic interconnects were introduced as a promising new technology for NoC communication. Photonic NoC provides lower latency, lower power, and higher bandwidth compared to wired interconnects [28,17]. However, the components are expensive and there are no buffers in the photonic interconnects. Therefore, it is not practical to implement traditional store and process techniques in photonic switches [29].

Chang et al. [7,6] have proposed multi-band RF NoC. The benefit of using RF interconnect is that it transmits electromagnetic (EM) waves, which travel at the effective speed of light, along the wires. Higher bandwidth and lower latency can be obtained by using this technology [5].

Another promising solution for a scalable interconnect architecture are wireless interconnect networks. Lee et al. [20] have proposed a wireless on-chip architecture which uses a hybrid wireless and wired architecture to interconnect cores. They have shown that simple transmitters/receivers operate at the 100–500 GHz sub-Terahertz frequency through miniature antennas, which reduces latency and power consumption compared to a 2-D mesh network. Ganguly et al. [15] introduced a new on-chip antenna based on Carbon NanoTubes (CNTs) for on-chip wireless communication and evaluated the latency, the throughput, and the energy dissipation. They showed that the performance, in terms of throughput, latency, and power consumption improved compared to a general wired network. In particular, more improvement can be seen in non-uniform traffic distributions compared to uniform traffic distributions by using low power and high speed long-range wireless links, which enable a single hop communication between distant nodes. In Deb et al. [11], the authors introduced the design of a hierarchical small-world network with long-range and low power mm-wave wireless links for NoC (mWNoC) and

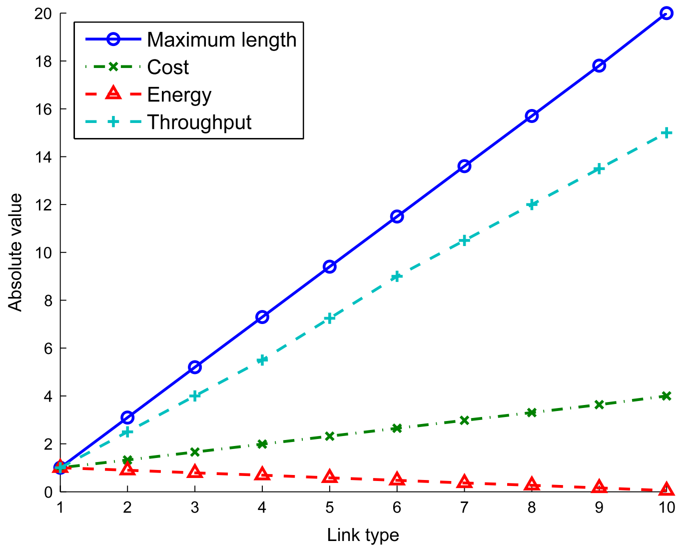


Fig. 1. Definition of the different types of abstract links.

showed a performance improvement in both uniform and non-uniform traffic cases.

While many authors have introduced new solutions to improve the network performance, they did not consider balancing all of the important performance metrics of the network, such as the wiring cost, the average shortest path length, the latency, the throughput, and the power consumption. Also, they compared the results with a regular 2D mesh network only and did not compare the networks with each other. To the best of our knowledge, no one has tried to implement an irregular structure based on NoC architecture with three or more heterogeneous link types.

3. Architectural overview

In this section, we will present the basic architecture of our framework, define measures, and introduce the methodology.

3.1. The network model

We use a graph-theoretic approach to represent our networks [30]. A graph G is denoted by $G = (N, E)$, where N is a set of nodes and E is set of edges. Here an edge represents a bidirectional communication link between two nodes. On such a link, a packet can be sent from a source to a destination node. For example, a 4×4-node 2D mesh network is shown in Fig. 2(c). In this example, node 5 is connected to four different adjacent nodes, $E_5 = \{\{5, 1\}, \{5, 4\}, \{5, 6\}, \{5, 9\}\}$, i.e., a packet can be transmitted between node 5 and the four adjacent nodes, and vice versa. The graph-theoretic approach can easily be extended and it allows us to directly employ all tools and methodologies from the graph and complex network community.

For our purpose, all network nodes are arranged in a 2D grid with a one-unit grid spacing (see Fig. 2). As opposed to previous work, e.g., [26], we do not start with a mesh interconnect, instead, our optimization algorithm is allowed to place links in an unrestricted manner. That allows us to explore the entire search space of network topologies.

3.2. Link type definitions

To explore heterogeneous complex networks with different types of abstract links on the same network, we defined ten types of links by choosing ten evenly spaced points of characteristics between link type 1 and link type 10 as shown in Fig. 1. Link type 1 and link type 10 can be thought of as representatives of metal wires and photonic links [10] respectively. Note that all these link types are abstract and technology-agnostic on purpose in order to obtain results that are broadly applicable. As explained in the introduction, the goal of this paper is to find out what level of heterogeneity is optimal. We have chosen ten different link types for this paper because the differences in each metric were still interpretable. More link types would have made the results significantly harder to analyze and interpret. In Section 5 we also explore heterogeneous link type networks with a non-linear cost mapping of the links to show that the main results remain unchanged.

The y-axis in Fig. 1 shows the value of each metric. Each link is defined by a different value for the maximum wire length, the wiring cost, the energy consumption, and the throughput. The wiring cost is defined as a function of the pre-defined cost of a link type multiplied by the actual length plus a constant,

$$F_{cost} = (a * WireLength^{b}) + c, \quad (1)$$

where a, b, and c are user defined parameters. For example, the cost of link type 10 is pre-defined as 4 and $c = 0$. Note that we have only used the fixed cost factor in Section 5.2. Table 1 shows the absolute values of each metric for link type 1, 6, and 10. The rest of the link type values can be seen

Table 1
Absolute value of each metric for link type 1, 6, and 10.

	Link type 1	...	Link type 6	...	Link type 10
Maximum length	1 unit		11.5 units		20 units
Variable cost	1		2.65 × actual length		4 × actual length
Energy	1 unit/packet		0.48 unit/packet		0.05 unit/packet
Throughput	1 packet/clock		9 packet/clock		15 packet/clock

in Fig. 1. Note that we do not make any statement whether our ten link types can be implemented in some technology at this point in time. We are solely interested in – as stated in the Introduction – to find out what level of heterogeneity is optimal and less about the absolute performance of the networks.

By using these abstract links, the goal then becomes to find an optimal heterogeneous network that has a low cost, a high throughput, and a low energy consumption.

For most of our experiments, we limit the number of each link type. The reason for that is the potential for the network cost to explode if the optimization algorithm is allowed to place unlimited numbers of links when cost is not (or only weakly) considered. Link type limitations are specified in each experiment.

3.3. Network performance metrics

For our purpose, we considered network performance metrics such as the wiring cost, the average shortest path length, the throughput, and the energy dissipation to evaluate the network performance. In this section, we will provide brief definitions of these metrics, which are inspired by [27]. For our purpose, we do not consider packet lengths or flits. Without losing generality, we simply consider a packet as an abstract unit that is communicated on the network. However, we also ran additional performance simulations in Section 6 by using the GEM5 framework, which uses realistic traffic scenarios and flits.

Network wiring cost. We define wiring cost as the sum of the cost of all wires in the network between a source node i and a destination node j that are directly connected.

$$WiringCost_{network} = \sum_{i,j \in G} Cost_{wire(i,j)}. \tag{2}$$

For example, the wiring cost for the 4 × 4-node mesh network shown in Fig. 2(c) is 15. Increasing the network size or adding additional links will increase the total number of links in the network, and hence the wiring cost.

Network average shortest path. The *Average Shortest Path* (ASP) [24] is defined as the average number of hops in the shortest path for all possible i, j pairs of the network nodes with $i \neq j$.

$$ASP_{network} = \sum_{i,j \in G} \frac{distance(i,j)}{N(N-1)} \tag{3}$$

where N is the number of nodes. The *ASP* measures how efficiently packets can be transported on the network. In an uncongested network with shortest path routing, the *ASP* is proportional to the packet latency. The *ASP* for the 4 × 4-node mesh network shown in Fig. 2(c) is 3.87. When we add additional long-range links on the mesh network, the *ASP* will drop quickly while the clustering coefficient stays high. This is commonly called the *small-world* property of a network [33].

Network throughput. When simulating traffic on the networks, packets will be injected into specific nodes with a given injection rate. The injection rate *iR* is defined as the average number of packets injected into a node per clock cycle. The network throughput is defined as the total number of packets arrived at their destination per node per cycle T:

$$TP_{network} = \frac{Total\ packets\ arrived}{(N * Total\ time)} \tag{4}$$

N is the number of nodes, and *Total time* is the time (in clock cycles) that elapses between the occurrence of the first message generation and the last message reception.

Network energy dissipation. The network energy is defined as the sum of the energy required to move packets on the network across links divided by the number of cycles a simulation is run. On each link from source i to destination j, the number of packets sent will require an energy of $E_{packet}(i,j)$. We use our abstract link energy estimates per packet as shown in Fig. 1.

$$E_{network} = \frac{\sum_{i,j \in G} E_{packet}(i,j)}{duration}. \tag{5}$$

3.4. Network traffic

In our framework, packet traffic is simulated in a cycle-accurate manner. However, we do not simulate flits and do not use sophisticated flow control because that is not essential to obtain our results. As a matter of fact, our findings are independent of these details. Traffic is routed by means of a shortest path algorithm. We have chosen to not use flits in order to keep the packet routing deadlock free. There are of course more optimal routing and flow control algorithms than simple shortest path routing, but our approach allows to keep things simple without losing generality. Each node contains FIFO buffers of size N, where N is the number of network nodes, that store packets upon their arrival.

We use three types of synthetic traffic patterns as described below. For each packet, a source/destination pair (i, j) is generated that depends on the traffic pattern used. In addition, we use the SPLASH-2 [34] FFT benchmark for realistic application-based traffic patterns.

Uniform random. In uniform random traffic, the source and destination nodes are randomly chosen among all nodes with equal probabilities.

Hot-spot. In hot-spot traffic, selected hot-spot nodes receive packets with a greater probability (p) than non-hot-spot nodes ($1 - p$). In our experiments, unless otherwise stated, we use two hot-spot nodes, namely node 9 and 54 in 8×8-node networks. For networks of size 10×10-node, 12×12-node, 16×16-node and 20×20-node, node 11 and 88, node 13 and 130, node 17 and 238, and node 21 and 378 are used as hot-spot nodes respectively. The hot-spot probability is $p = 0.25$.

Transpose. In transpose traffic, the source (i, j) and destination (j, i) node pairs are located symmetrically to the diagonal in a matrix. A roulette wheel is used to select the source and destination pair. For our 8×8-node networks, we used the following node pairs: $(19, 26)$, $(13, 41)$, $(57, 15)$, and $(52, 38)$.

3.5. Finding optimal networks

Finding optimal NoCs is all about trade-offs. It is rather straightforward to design a high-performance network (e.g., a fully connected), but it will also have a significant cost. On the other hand, a low-cost network may not offer the best performance. The trade-off between performance and cost in general is a design decision that depends on the application. However, most NoC design problems involve several additional factors besides performance and cost. The problem can therefore straightforwardly be formulated as a multi-objective optimization problem.

Evolutionary algorithms (EAs) [13,12] are a well-known metaheuristic technique to solve multi-objective optimization problems. EAs are stochastic search methods based on the principles of natural biological evolution. The basic operation is based on a population of candidate solutions. First, one randomly generates an initial population of individuals, which will then be evaluated by means of an objective or fitness function. If the termination criteria are not met, one creates a new generation of individuals by applying mutation and recombination operators to the parent individuals. This process is repeated until the best solution is found. A pseudo algorithm of the basic parallel evolutionary optimization procedure is described in Algorithm 1, which is inspired by [14].

ALGORITHM 1: Pseudo-code for a parallel evolutionary algorithm.

```
G ← number of generation, P ← population size,
O ← offspring, g ← 0
Randomly generate initial population P ;
while g < total generation number G do
    g ← g + 1;
    Evaluate initial population P(t);
    Crossover:
    Generate a random number x between (0,1);
    if x < P_c then
        Randomly select two parents ;
        Recombine to create a new offspring O ;
        P'(t) ← P'(t) ∪ O;
    end
    Mutation:
    Generate a random number x between (0,1);
    if x < P_m then
        Mutate to create a new offspring O ;
        P'(t) ← P'(t) ∪ O;
    end
    Evaluate P'(t)
    Selection:
    P(t + 1) ← select(P(t) ∪ P'(t)) ;
end
```

The strength of EAs is that they perform well with problems that have multiple local optima [35]. EAs are typically used to find best solutions for given problems that cannot easily be solved by using other optimization techniques, such as simulated annealing. They were successfully used in a variety of optimization problems, such as scheduling, routing, transportation problems, and engineering design [23,21]. For these reasons, we have chosen evolutionary optimization techniques to design optimal large-scale heterogeneous NoC architectures. We can obtain high quality solutions quickly and evolve large networks straightforwardly. Often, however, the resulting networks look somewhat unstructured to the human eye, and it can thus be hard to understand and analyze them. In a recent paper [8], we have shown that our evolved heterogeneous networks have sub-community structure, where high throughput long-range links are used to communicate between communities and low cost short-range links are used to connect within a community. Our evolutionary algorithm platform is based on the *ParadisEO* framework [4], which is a C++ white-box object-oriented framework dedicated to the reusable design of metaheuristics.

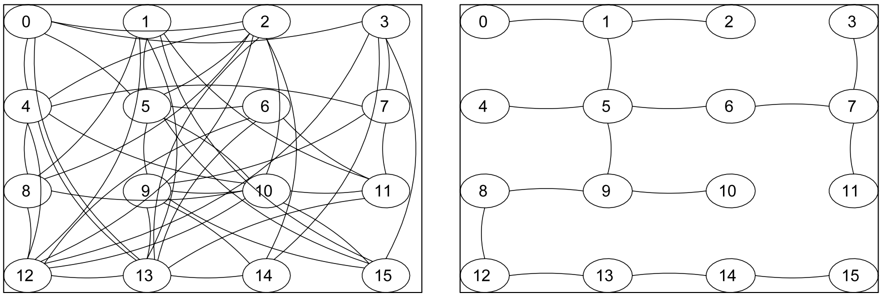
Network representation. In our C++ complex network framework, the individual components that make up a network are nodes, links, and link types. The nodes are implemented as an array of pointers to node objects consisted of the physical (x, y) coordinates of the nodes on the 2D grid. The link object stores the link information as a set of node pairs, which includes all relevant link properties, such as cost, maximum length, throughput, and energy consumption. We do not use a genotypical representation for our evolutionary algorithm, instead, all genetic operators work directly on the network level.

Crossover. We perform crossover in the following way: each individual $p1$ and $p2$ has a set of link data (i.e., node numbers) stored in an array for each link type. For each link type, we randomly pick two crossover points $(pt1, pt2)$ and then perform standard two-point crossover. We do this for each link type. The crossover rate we use in our experiments is 0.6.

Mutation. We use two mutation operators in our framework. The first mutation operator randomly selects a link type and changes the number of links by adding and removing a link from the link vector. The second mutation moves a current link to different location in the network. Unless otherwise specified, the mutation rate is 0.4.

Selection. We use deterministic tournament selection to select new individuals for the next generation.

Fitness function. For most of our problems, we consider multiple network performance factors. To consider two



(a) $w = 0$.

(b) $w = 1$.

Fig. 2. 4×4-node evolved networks. (a) $w = 0$: performance is favored. The result is an almost fully connected network. (b) $w = 1$: wiring cost is favored. The resulting network is a sparsely connected tree. Only one link type was used for this example.

different factors only, we introduce an objective aggregate fitness function. For example, to optimize networks for both cost (*WiringCost*) and performance (*ASP*), we defined an aggregate objective fitness function as the following:

$$f(w) = w \times WiringCost_{network} + (1 - w) \times ASP_{network}. \quad (6)$$

Here, w is the weight factor that allows us to determine the importance of either of the two factors, and *WiringCost* and the *ASP* are normalized. For example, with $w = 0$, only the network performance is considered, and for $w = 1$, only the wiring cost is considered. For $w = 0.5$, both *WiringCost* and *ASP* are equally favored. The aggregate objective function can readily be extended to include additional factors a designer may want to consider, such as energy or area overhead, router cost, and technology-related costs.

Population size and evolutionary runs. Unless otherwise stated, we use a population of 600 individuals evolved over 14,000 generations. For all experiments, we did 10 evolutionary runs with 10 different initial populations and averaged the results. These parameters were determined experimentally and produced very robust results.

Example. As an illustrative example, we optimized networks by changing the weight w in Eq. (6) to see what kind of NoC topologies we would obtain. We used the EA as described above with only one link type. Fig. 2 shows the results for a $4 \times 4 = 16$-node evolved network, which can be represented as a 16×16 adjacency matrix, with different weights $w = 0$ and $w = 1$.

We observe that when performance is favored ($w = 0$), we have more long-range links in the almost fully connected network to reach the destination with less number of hops. However, when we consider wiring cost only ($w = 1$), the network becomes a sparsely connected tree and uses local connections only to keep the network cost as low as possible.

4. Performance evaluation

In this section, we will present the performance evaluation experiments in detail. The goal is to show the number of link types usage for optimal networks and to prove that networks with heterogeneous link types are beneficial in terms of cost, performance, and energy under three different traffic patterns.

4.1. Optimal number of links

The goal of the first experiment was to determine the optimal number of each link type for three types of traffic scenarios, namely uniform random, hot-spot, and transpose traffic. For this experiment, we evolved optimal 8×8-node networks with an injection rate of $iR = 0.6$. We limited the number of each link type to 112 because that is the number of links required to complete a local 2D mesh of size 8×8 nodes. In addition, we prevent multiple connections of a given link type between any two nodes. As a baseline for comparison, we used a mesh network.

We will present the results individually for three different traffic scenarios.

4.1.1. Networks with uniform random traffic

First, we evolved networks under random uniform traffic. For that purpose, we randomly generated packets with source and destination nodes selected with a uniform probability. The injection rate for this experiment was $iR = 0.6$.

Fig. 3 shows the resulting link type distribution for networks optimized for *WiringCost* and throughput *TP* with the aggregate function $f(w) = w \times WireCost + (1 - w) \times TP$. As one can see, the optimal networks use a large number of low cost links of type 1 and 2 to distribute the traffic and provide high throughput instead of using a small number of expensive link types. Fig. 4 shows the corresponding networks for one of the evolutionary runs.

We also evaluated the energy *E* combined with *WiringCost*. As one can see from Fig. 5, the results are quite different from the throughput experiments: more costly long-range links for the majority of the weight value w are used to minimize the energy. Fig. 5 shows that when we gave more weight to *WiringCost*, a higher number of link types were used compared to weights between 0.2 and 0.5. However, in order to reduce the network cost, the total number of links used for each link type is much lower.

4.1.2. Networks with hot-spot traffic

Next, we used hot-spot traffic [26] as a more realistic traffic pattern. The two hot spots are node 9 and 54. The hot spot probability is $p = 0.25$, i.e., 25% of the packets will be sent to the hot-spots.

The result of optimizing the networks for *WiringCost* and *TP* are shown in Fig. 6. The distribution plot shows that

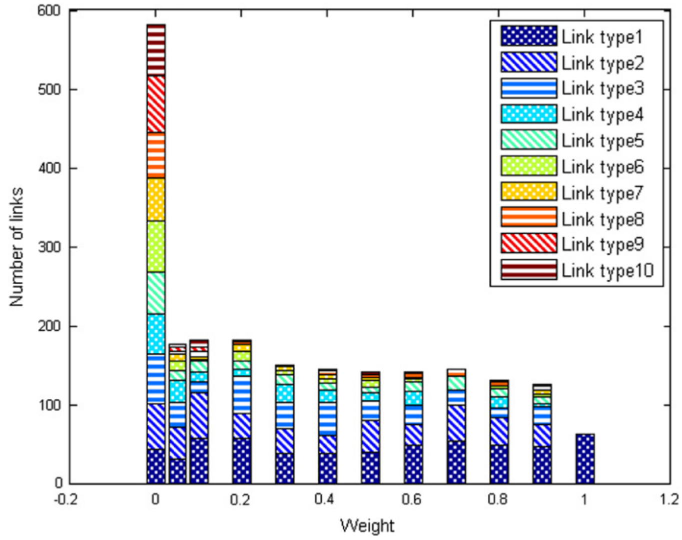
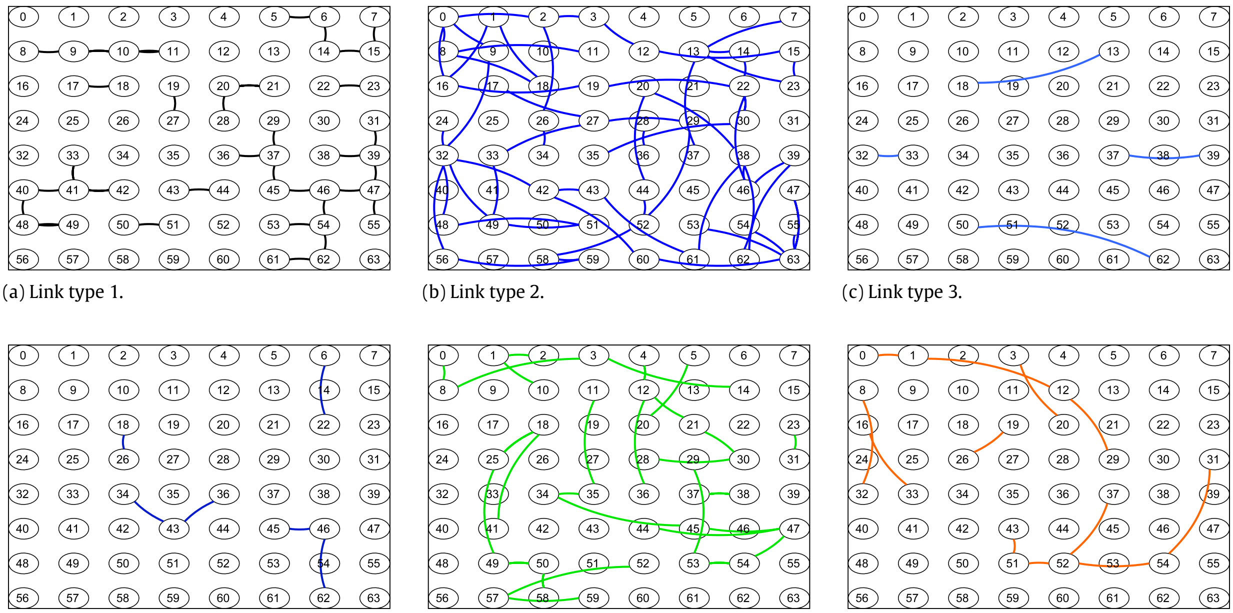


Fig. 3. Heterogeneous link type distribution as a function of the weight w. The networks are optimized for *WiringCost* and *TP* with uniform random traffic. Injection rate $iR = 0.6$.



(a) Link type 1. (b) Link type 2. (c) Link type 3.

(d) Link type 5. (e) Link type 7. (f) Link type 8.

Fig. 4. 8×8-node evolved network with uniform random traffic. (a)–(f) $w = 0.5$: *WiringCost* and *TP* are equally important. Six link types were used in this network. $WiringCost = 430.2$, $TP = 0.59$.

almost all optimal networks use a large number of short-range link to absorb the traffic at a lower cost.

The result for optimizing *WiringCost* and *E* with hot-spot traffic are shown in Fig. 7. Long-range links of type 9 and 10 are used more frequently for the majority of the weight values w to use less energy.

4.1.3. Networks with transpose traffic

In transpose traffic [26], the source (i, j) and destination (j, i) nodes pairs are located symmetrically to the diagonal in a matrix. A roulette wheel is used to select the source and destination pair. Here, we use the following node pairs: (19, 26), (13, 41), (57, 15), and (52, 38) for our 8×8-node networks.

Fig. 8 shows the result for networks optimized for *WiringCost* and *TP*, i.e., the aggregate objective function is $f(w) = w \times WiringCost + (1 - w) \times TP$. As one can see, short-range links of link type 1 and 2 are used more frequently to absorb the traffic for the majority of the weight values. This becomes apparent when we put more weight on *TP*, where the optimal network uses more short-range links compared to the optimal network with hot-spot traffic.

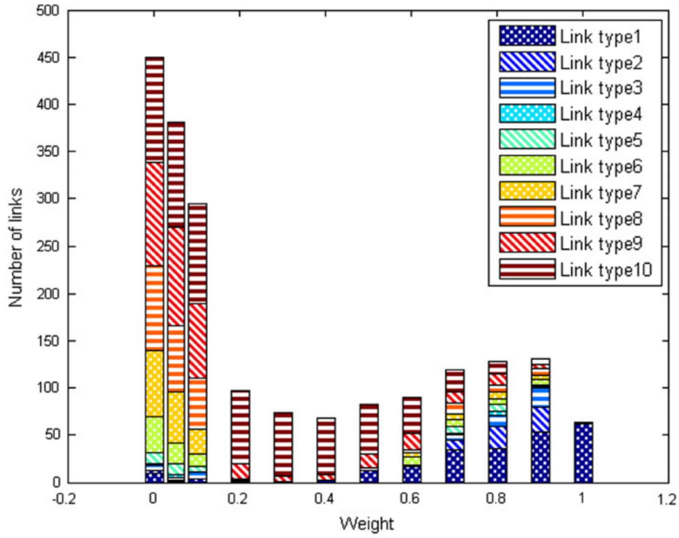


Fig. 5. Heterogeneous link type distribution as a function of the weight w. The networks are optimized for *WiringCost* and E with uniform random traffic. Injection rate $iR = 0.6$.

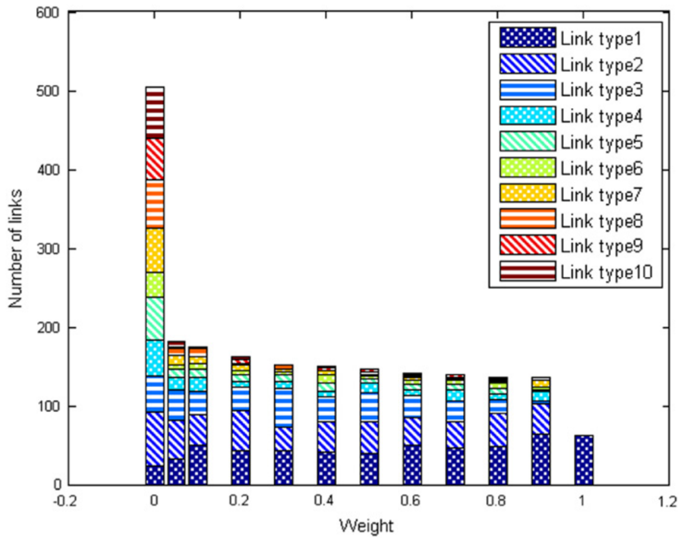


Fig. 6. Heterogeneous link type distribution as a function of the weight w. The networks are optimized for *WiringCost* and *TP* with hot-spot traffic. Injection rate $iR = 0.6$.

Fig. 9 shows the result for networks optimized for *WiringCost* and E. When energy is considered to be more important, the optimal network uses more long-range links because they are more energy-efficient. However, when we give more weight to cost, it uses less long-range and more short-range links to lower the network cost.

4.1.4. Comparison and discussion

We will briefly discuss and compare the results from Sections 4.1.1 to 4.1.3.

Fig. 10 shows the total number of links (i.e., the sum of ten link types) used in the networks from the previous experiments. Note that we had limited the number of each link type to 112, which is the number of links that would be required for a complete local 2D mesh of size 8×8 nodes. As one can see in Fig. 10(a), the evolved networks with transpose traffic use slightly more number of links compared to hot-spot and uniform random traffic to absorb the traffic. Hot-spot and uniform random traffic show very similar results otherwise. When optimized for cost and energy, the results are somewhat different, as Fig. 10(b) shows. The total number of links used in the network under different traffic patterns is very similar. This shows that our results are independent of traffic patterns. Also, considering energy as an optimization factor results in using about half the number of links compared to networks where *TP* and *WiringCost* are considered.

Fig. 11 shows the total number of link type usage in the networks. When the network is optimized for *TP* and

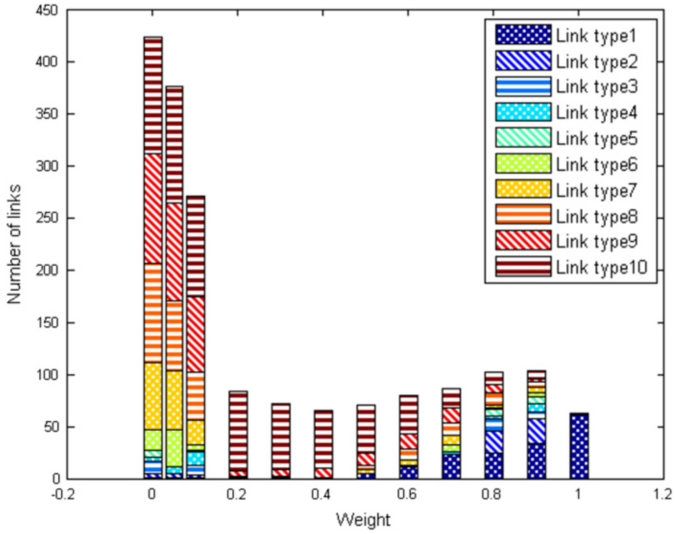


Fig. 7. Heterogeneous link type distribution as a function of the weight w. The networks are optimized for *WiringCost* and *E* with hot-spot traffic. Injection rate $iR = 0.6$.

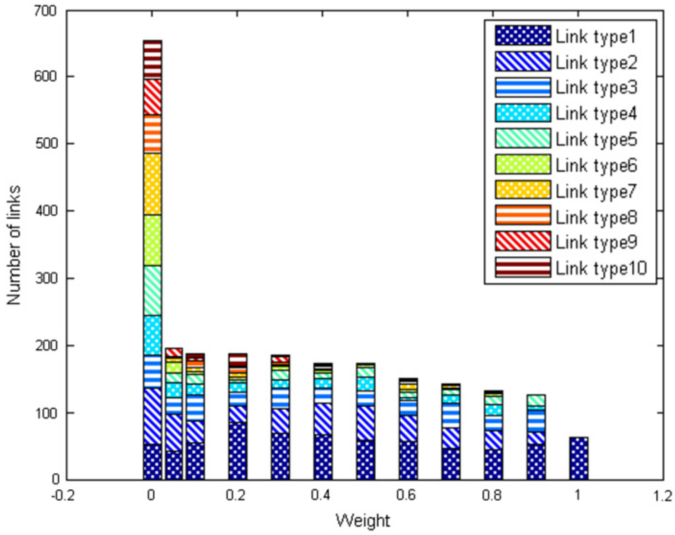


Fig. 8. Heterogeneous link type distribution as a function of the weight w. The networks are optimized for *WiringCost* and *TP* with transpose traffic. Injection rate $iR = 0.6$.

WiringCost, an average of 5.5 link types were used for the majority of the weight value w. We observe that these evolved networks use a large number of low cost link types to absorb the traffic and to provide high throughput instead of using expensive link types. However, when the network is optimized for *EG* and *WiringCost*, the total number of link types is different from the throughput experiment. As one can see from Fig. 11(b), an average of two link types were used for weights w between 0.2 and 0.5, and even more link types were used when we gave more weight to *WiringCost* in order to lower the energy. However, the number of links used for each link type is actually very low, which reduces the cost.

In summary: with the traffic patterns and trade-offs under consideration so far, it is beneficial in all cases (except when cost is the only factor) to make use of heterogeneous links to increase the performance or lower the energy at a low cost. The optimal number of different link types is the large majority of the weight w settings above 3.

4.2. Optimal network throughput

In this section we are investigating the optimal network throughput for different traffic patterns and injection rates. We limited the maximum number of links that can be used in the network for each link type to 112.

For this experiment, we consider 8×8-node networks with three different traffic patterns (see Section 3.4), namely uniform random, hot-spot, and transpose with a target injection rate of $iR = 0.6$. First we find an optimal

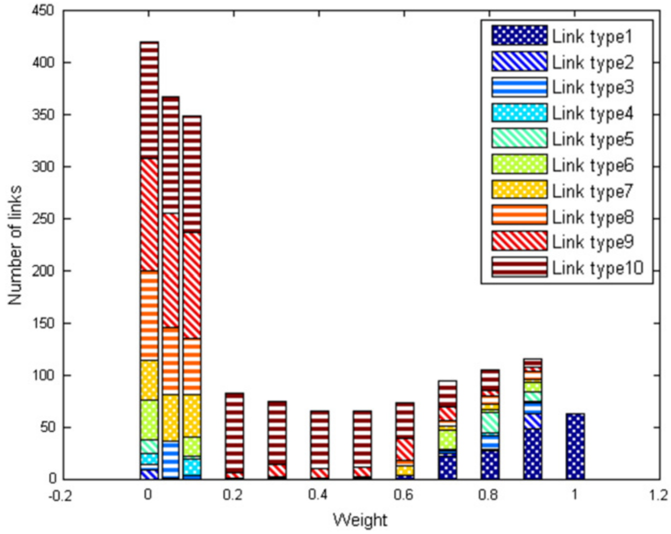


Fig. 9. Heterogeneous link type distribution as a function of the weight w. The networks are optimized for *WiringCost* and *E* with transpose traffic. Injection rate $iR = 0.6$.

network by optimizing cost and throughput for each pattern by equally weighting their importance ($w = 0.5$).

Next, we took these networks that were optimized for $iR = 0.6$ and subjected them to different injection rates to find out the throughput saturation point. As one can see from Fig. 12, the network throughput increases and peaks at the target injection rate of $iR = 0.6$. The throughput peaks at that value because the networks were optimized specifically for that target throughput. Because cost is a factor in the optimization, only the links that are absolutely necessary will be added to the network. Therefore, the network will not be optimal for any traffic that is beyond the target injection rate.

In addition, we also compared the network throughput *TP* of our evolved topologies with a network with three different link types and with a 8×8-node mesh under the same traffic patterns. The choice of using three link types only was motivated by the number of link types available with current technology, i.e., metal wires, wireless, and photonic links. However, note that we do *not* claim to model these link types accurately, which is beyond the scope of this paper. The goal is to work with abstract and technology-agnostic links to obtain results that are more fundamental and more broadly applicable.

Fig. 12 shows that all of our evolved topologies with ten heterogeneous link types provide a higher throughput compared to mesh networks. Moreover, our proposed networks have better performance compared to the networks with three link types. This result shows once again that networks with ten different heterogeneous link types can achieve better performance than all the other networks we considered. It is not surprising that adding long(er)-range connections to a mesh network helps improve the performance. This was successfully shown by Ogras and Marculescu [26], but only for homogeneous networks. In Section 4.4 we show that our heterogeneous networks perform better than any other homogeneous single link type networks.

4.3. Optimal energy dissipation

As the number of cores on a single chip increases, energy dissipation is another challenge in a traditional NoC. Multi-hop communication based on metal wires significantly increases the energy consumption, so any link type that can communicate to distant cores in a single hop with low energy and high bandwidth is interesting to explore.

In this section we use the ten abstract heterogeneous link types defined in Section 3.2 to find optimal networks under *WiringCost* and energy *E* constrains with equal importance, i.e., $w = 0.5, f(w) = w \times WiringCost + (1 - w) \times energy$. We once again limited the maximum number of links that can be used in the network for each link type to 112. The same traffic patterns and injection rate $iR = 0.6$ are used. We then took the optimal networks for each weight w and subjected them to traffic with different injection rates. We compare the results with the networks with three different link types and mesh networks.

As one can see from Fig. 13, the evolved topologies consume much less energy compared to the mesh network for all three traffic patterns. This is because the evolved topologies are interconnected with long-range links that help to reduce the number of hops. In addition, Fig. 13 shows that our evolved topologies with ten different link types consume less energy than the networks with three different link types.

The results show that with respect to energy consumption, optimal networks with heterogeneous links are more efficient than a regular 2D mesh networks. Thus, as seen here and in Section 4.2, the evolutionary algorithm does a splendid job in obtaining networks that meet the target requirements while minimizing both energy and cost.

4.4. Network performance comparison with mesh and homogeneous network

Next, we evaluated the performance metrics, such as throughput *TP* and energy *E* of networks with hetero-

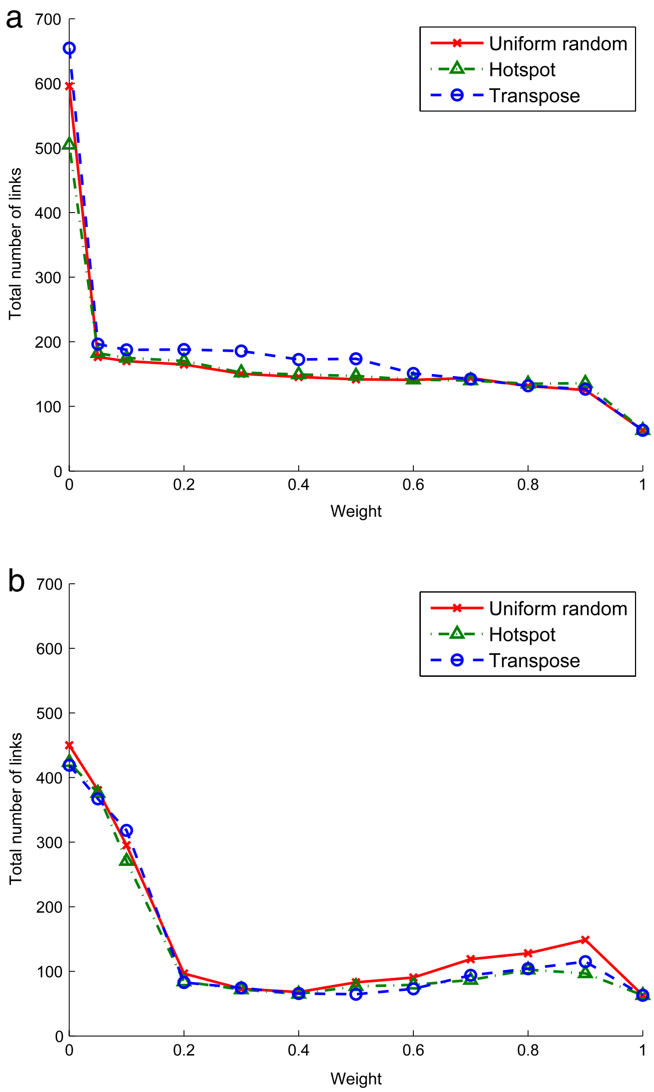


Fig. 10. Total number of links used as a function of the weight w for three different traffic patterns with injection rate $iR = 0.6$. (a) The network is optimized for *WiringCost* and *TP*; (b) the network is optimized for *WiringCost* and *E*.

geneous link type and compared them with regular 2D meshes and networks with homogeneous link types. In addition, we compare the results with networks with three different link types. We define a *homogeneous link type* network as a network that only uses link type 1, but without a length restriction to allow long-range links. This network is based on a regular 2D mesh which contains additional long-range connections to improve the performance. The network therefore is similar to the type of network as proposed by Ogras and Marculescu [26]. The difference is that we consider cost while they did not.

Fig. 14 shows the comparative results of the different heterogeneous link type networks with regular 2D meshes, homogeneous link type, and three different link type networks under uniform random traffic. As one can see, both the network throughput *TP* and energy *E* of the evolved networks with heterogeneous links are significantly better compared to a regular 2D mesh network over the entire weight range, except when the designer gives preference of 80% or more to cost. Also, we compared our evolved networks with link type networks. Fig. 14(a) shows that our proposed networks provide similar or higher throughput compared to the link type networks. Fig. 14(b) shows that when the weight value w is less or equal than 5, our evolved topologies use less energy.

Figs. 15 and 16 shows the same performance comparison for hot-spot and transpose traffic. Again, one can see that our evolved networks with heterogeneous links support high throughput and low energy consumption compared to 2D mesh networks or homogeneous link type networks.

In summary: the results in this section indicate that having multiple types of links improves the network performance and lowers the energy at a low cost.

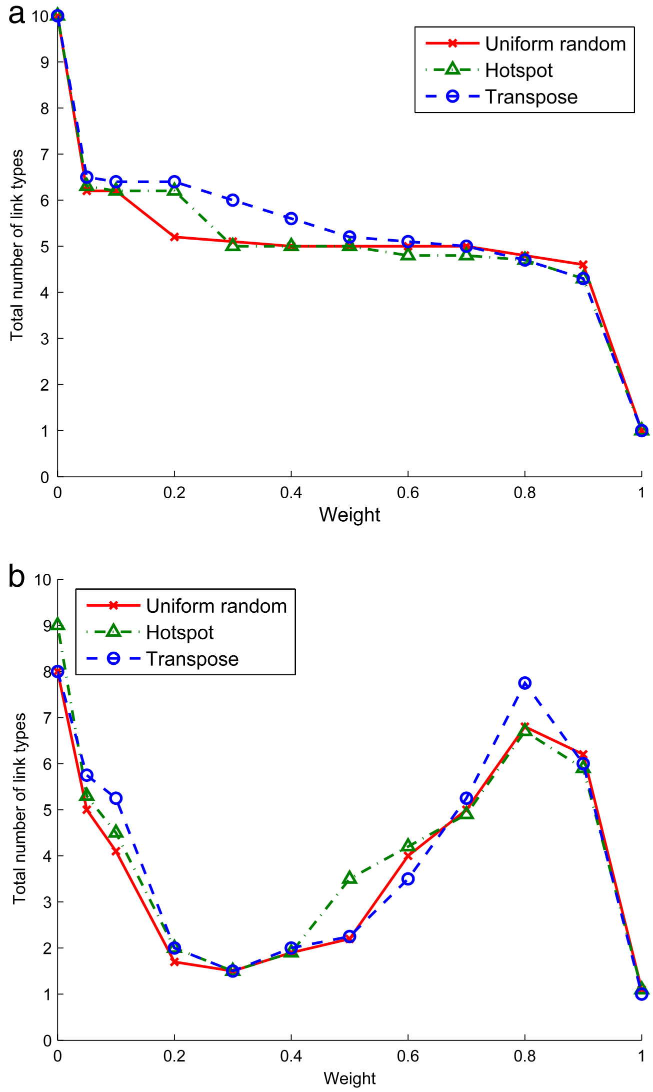


Fig. 11. Total number of link types used as a function of the weight w for three different traffic patterns with injection rate $iR = 0.6$. (a) The network is optimized for *WiringCost* and *TP*; (b) the network is optimized for *WiringCost* and *E*.

4.5. Scalability

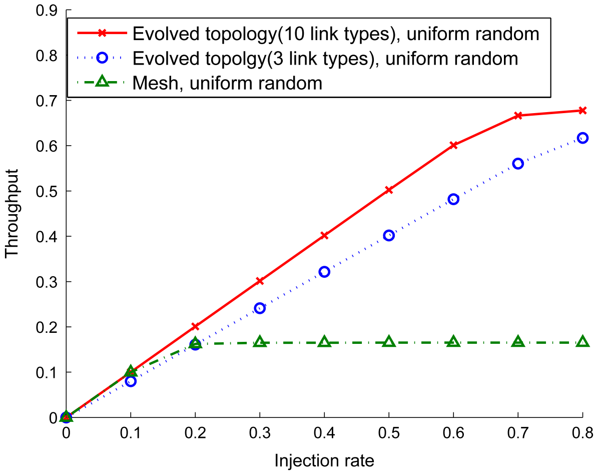
As the number of cores integrated on a single chip increases, scalability is another challenge that needs to be addressed by the NoC community.

We were interested to find how the performance, cost, and energy of networks with heterogeneous links scale up as a function of the system size *N*. For this experiment, we evolved networks of sizes 8×8-node, 10×10-node, and 12×12-node and subjected them to hot-spot traffic with an injection rate of $iR = 0.6$. We then compared the network throughput *TP* and the energy *E* with regular 2D mesh networks. To account for the bigger network sizes, we increased the maximum number of link types to 180 for a 8×8-node, 264 for a 10×10-node, and 12×12-node network respectively.

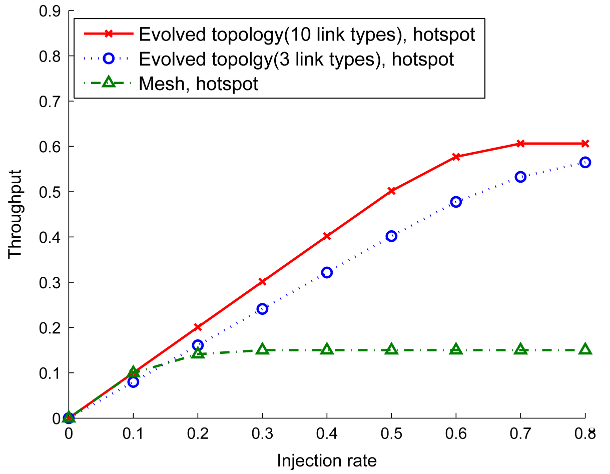
The results for the network throughput *TP* and the energy *E* of different sizes are shown in Fig. 17. As one can see, networks with heterogeneous link types provide higher throughput and lower energy consumption compared to regular 2D mesh networks for all system sizes under consideration. We also observe that the network throughput decreases as the system size increases. This is simply due to network congestion.

Fig. 18(a) and (b) shows that using a mix of long-range, high throughput, and low energy links improve the network performance but also increase the network cost. However, when we analyze the network performance in terms of throughput per cost ($TP/WireCost$) as shown in Fig. 18(c), our evolved topologies perform better than 2D regular mesh networks when the network size is bigger than 240 nodes.

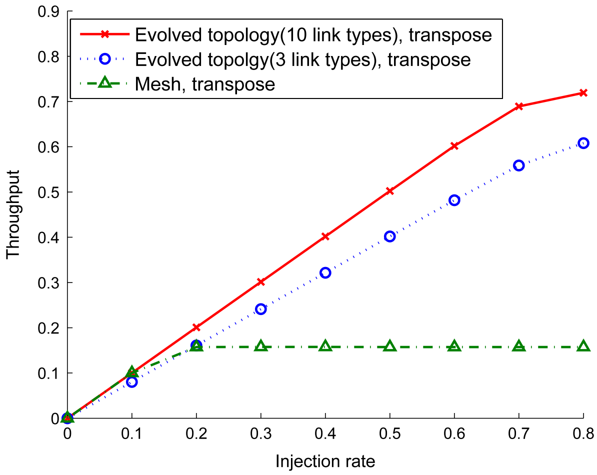
The evolved topologies for optimizing energy *E* and *WiringCost* are expensive compared to 2D regular mesh networks. However, these provide lower energy usage compared to mesh networks and scale well, as shown in Fig. 19.



(a) Uniform random traffic.



(b) Hot-spot.



(c) Transpose.

Fig. 12. Network throughput of evolved topologies and mesh networks under different traffic scenarios with a target injection rate of $iR = 0.6$. All evolved topologies with heterogeneous links perform significantly better than a 2D mesh topology.

5. Model variation

5.1. Performance comparison between linear and non-linear cost mapping of the links

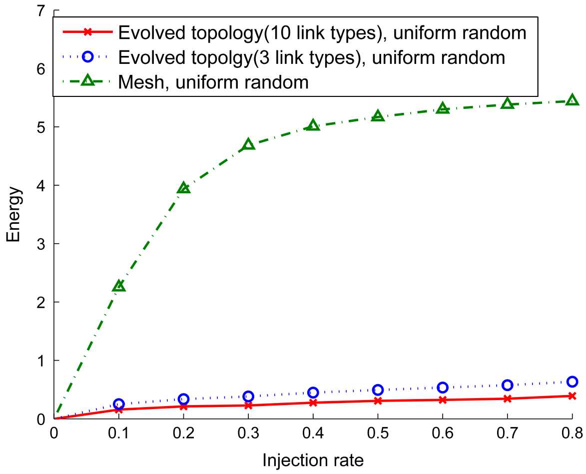
In order to show that our model is robust against certain assumptions, we have also performed additional simulations by using a non-linear (i.e., exponential) cost mapping of the links. Fig. 20 shows the non-linear cost mapping of each link type (see Fig. 1 for a comparison with the linear mapping). In order to reduced the effect of the other parameters, we decided to only change the cost function for this experiment. However, the outcome would be similar if the other relationships were changed to a non-linear mapping as well.

We then compared the results with the linear model shown in Fig. 1 in Section 3.2. For this experiment, we evolved optimal 8×8-node networks under uniform traffic with an injection rate of $iR = 0.6$. We limited the maximum number of links that can be used in the network for each link type to 112. Fig. 21 shows that the main results remain unchanged, i.e., the throughput is literally the same and the link type distribution differs by one link type only. These results confirm once again that heterogeneous link type networks are beneficial compared to homogeneous and regular mesh networks and that our model is robust against certain assumptions.

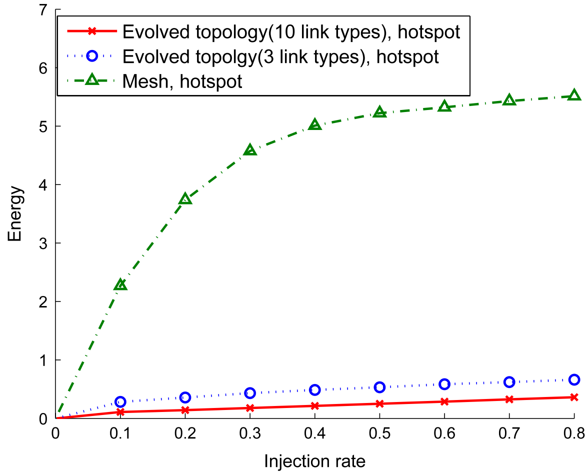
Based on these results, we believe that our abstraction level is appropriate for the type of study we performed. As we move toward more mature technologies, our approach can straightforwardly be used with the most recent technology parameters.

5.2. Performance comparison between different cost formula

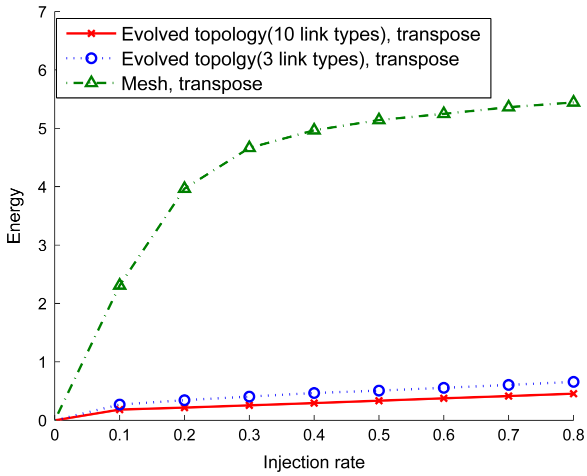
To account for different cost assumptions, we added a fixed cost component for each new technology that is being used when a new link type is instantiated. The goal is to show that our main outcomes remain unchanged under that new and more realistic model. The cost formula



(a) Uniform random traffic.



(b) Hot-spot.



(c) Transpose.

Fig. 13. Network energy of evolved topologies and mesh networks with different traffic scenarios. Injection rate $iR = 0.6$.

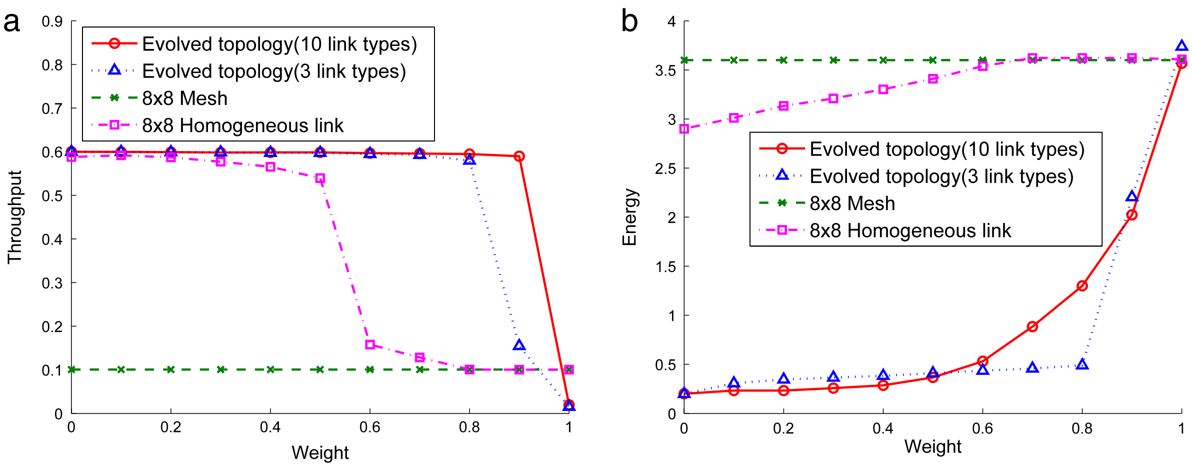


Fig. 14. Performance comparison of heterogeneous link type network with regular 2D mesh and homogeneous link type network under uniform random traffic. Injection rate $iR = 0.6$. (a) Network throughput TP as a function of the weight w. (b) Network energy E as a function of the weight w.

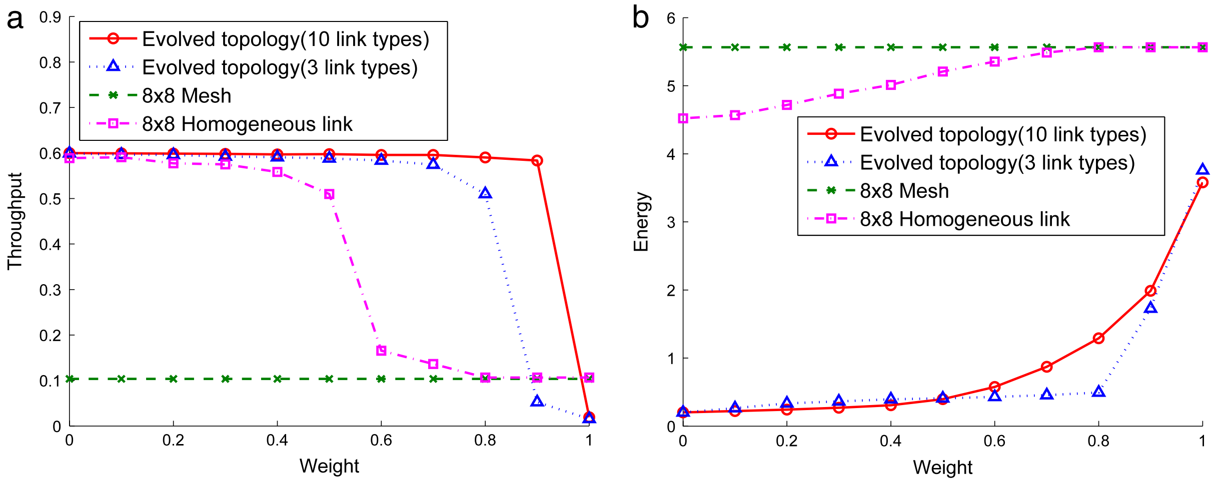


Fig. 15. Performance comparison of heterogeneous link type network with regular 2D mesh and homogeneous link type network under hot-spot traffic. Injection rate $iR = 0.6$. (a) Network throughput TP as a function of the weight w. (b) Network energy E as a function of the weight w.

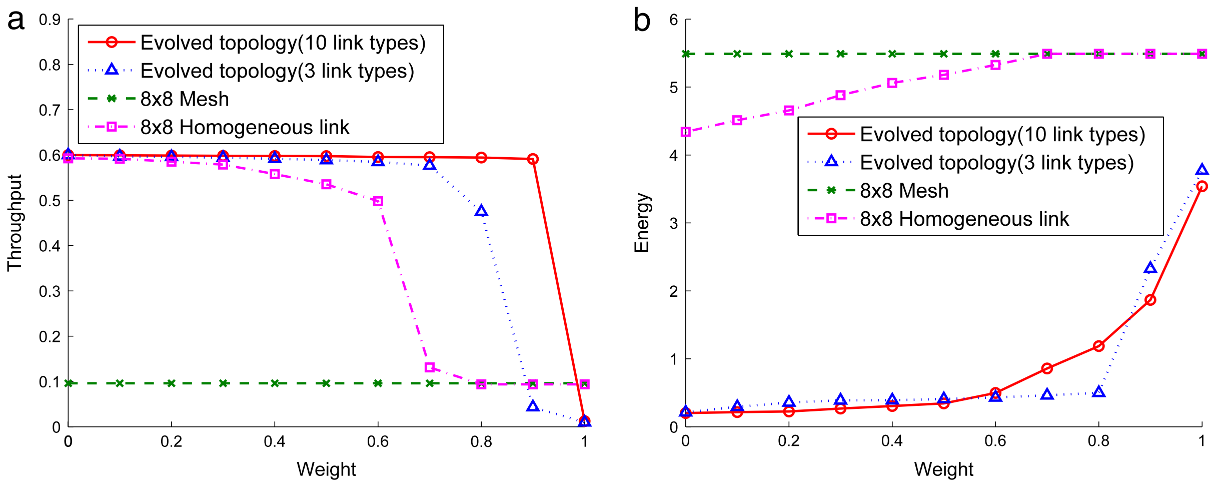


Fig. 16. Performance comparison of heterogeneous link type network with regular mesh and homogeneous link type network under transpose traffic. Injection rate $iR = 0.6$. (a) Network throughput TP as a function of the weight w. (b) Network energy E as a function of the weight w.

is shown in Eq. (1), Section 3.2. Table 2 shows the fixed cost values c we used for each link type technology. We assumed that the technology becomes more expensive with more powerful links.

Fig. 22 shows the results by optimizing *WireCost* and *TP* under uniform traffic. As one can see, the main results of our article remain unchanged. In particular, the throughput is almost identical and the link type distribution differs by one link type.

6. Performance evaluation by using the GEM5

In order to validate the results of our abstract framework, we used the GEM5 platform [3] to run a realistic benchmark with one of our evolved topologies. We used the SPLASH-2 [34], which is the most commonly used multi-threaded benchmark suite for parallel machines with shared memory in both academia and industry. As a real application benchmark, we use the Fast Fourier Transforms (FFT) [1] kernel, which is a complex one-dimensional algorithm that computes the discrete Fourier transform, to measure the network performance, evaluate our obtained evolved network with ten different heterogeneous link types, and compare the network performance with a regular 2D mesh.

Fig. 23 shows the network performance comparison between our evolved topology and a regular mesh network. As one can see, the latency with the SPLASH-2 FFT traffic load is much lower for our evolved topology. This validates our abstract framework and shows that the obtained network architecture with ten different link types provides high network performance compared to a regular 2D mesh.

Table 2
Fixed technology cost for link type 1, 6, and 10.

	Link type 1	···	Link type 6	···	Link type 10
Technology cost c	0.5		3		5

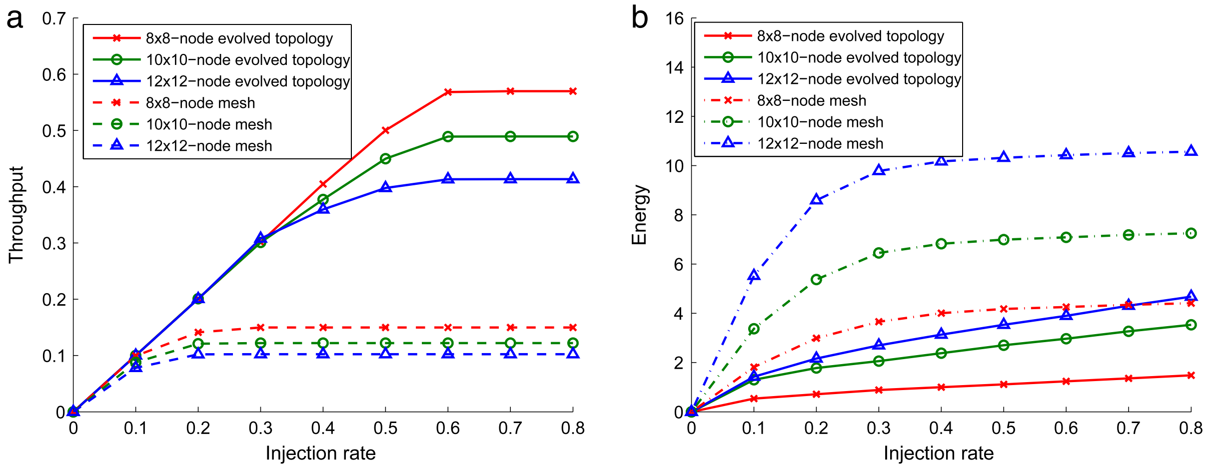


Fig. 17. Network throughput *TP* and energy *E* for different system sizes and injection rates. $w = 0.5$, $iR = 0.6$, hot-spot traffic. (a) Network throughput *TP* as a function of the injection rate. (b) Network energy *E* as a function of the injection rate.

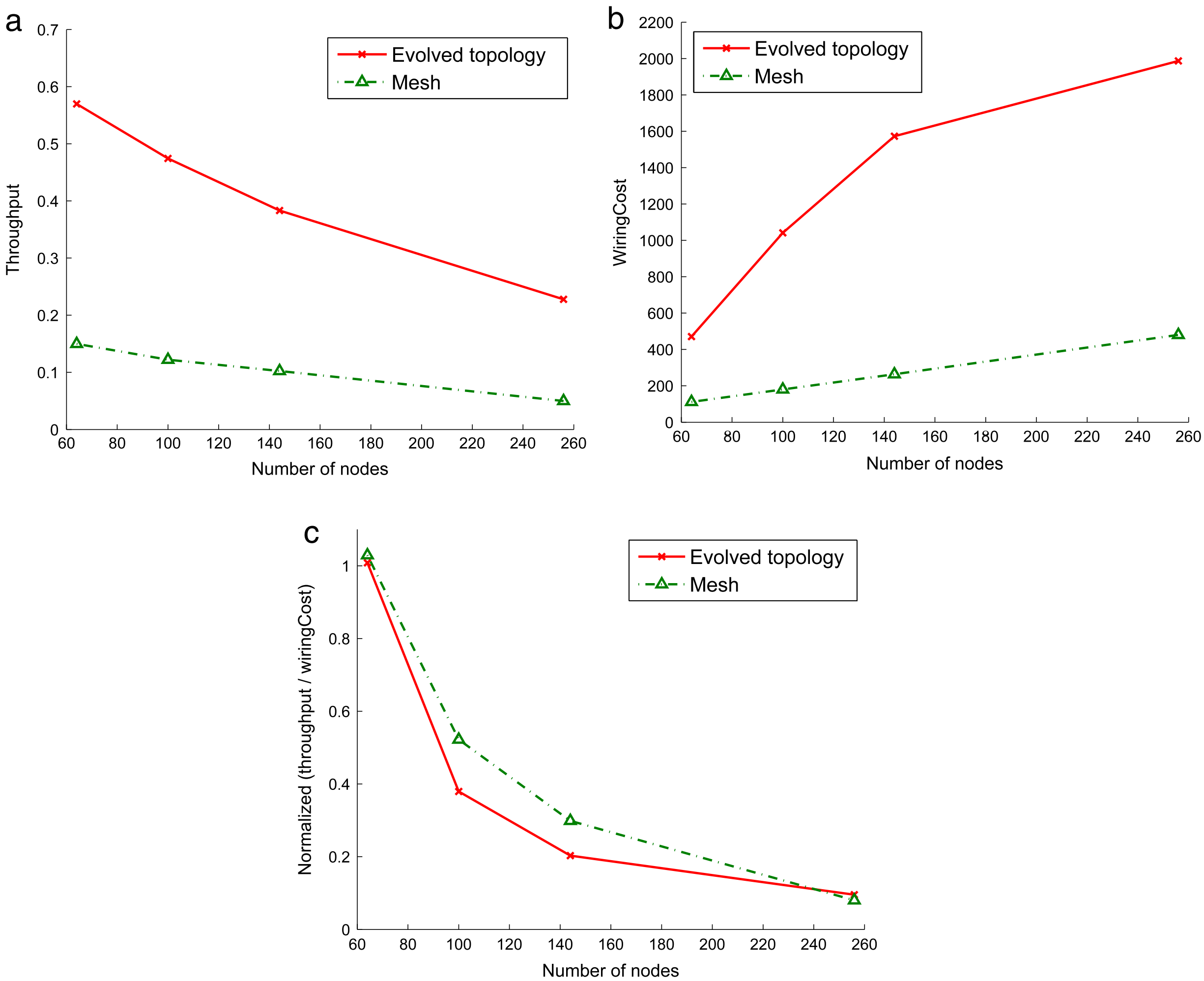


Fig. 18. Scalability for network sizes under hot-spot traffic. *WireCost* and *TP* is equally weighted ($w = 0.5$) using the aggregate objective function: $f = w \times WiringCost + (1 - w) \times TP$.

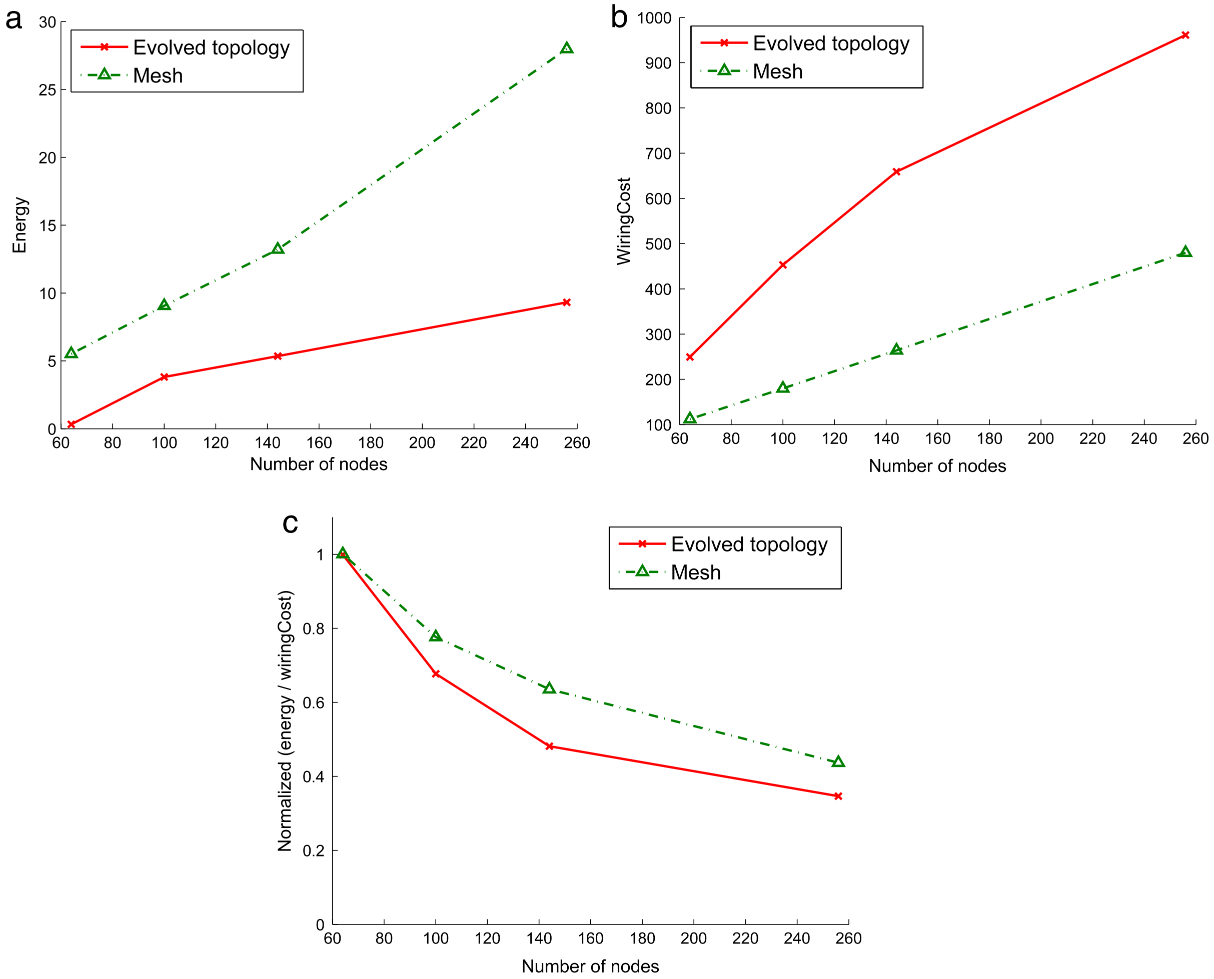


Fig. 19. Scalability for network sizes under hot-spot traffic. *WireCost* and E is equally weighted ($w = 0.5$) using the aggregate objective function: $f = w \times WiringCost + (1 - w) \times E$.

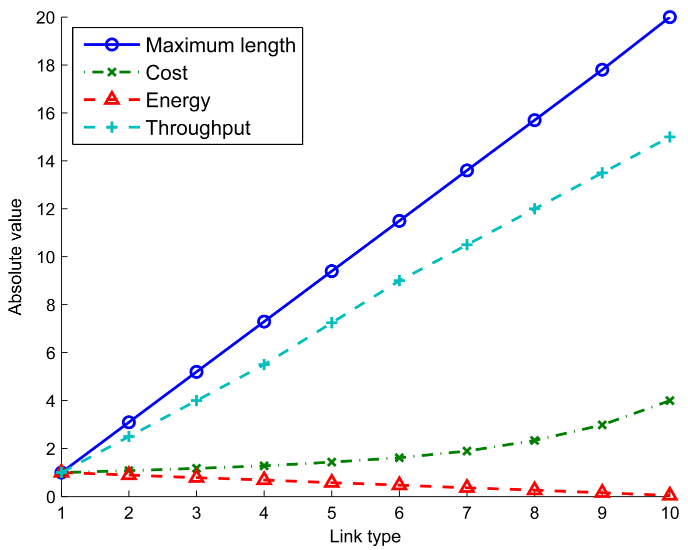


Fig. 20. Definition of the ten different types of abstract links with a non-linear (i.e., exponential) cost mapping.

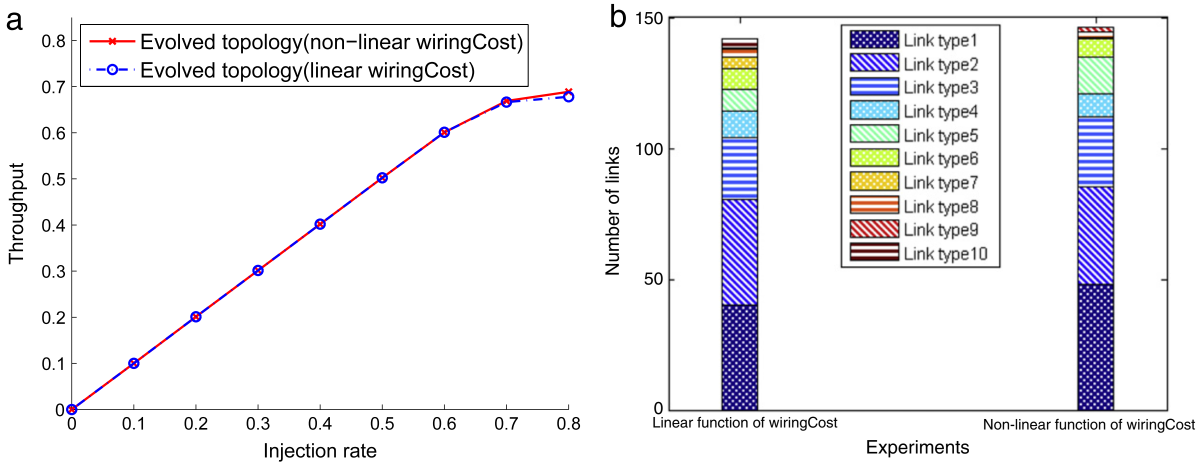


Fig. 21. Performance comparison between linear and non-linear cost mapping of links. Networks are optimized for *WireCost* and *TP* under uniform random traffic by equally weighting their importance ($w = 0.5$). Injection rate $iR = 0.6$. (a) Network throughput comparison of evolved topologies. (b) Comparison of heterogeneous link type distribution.

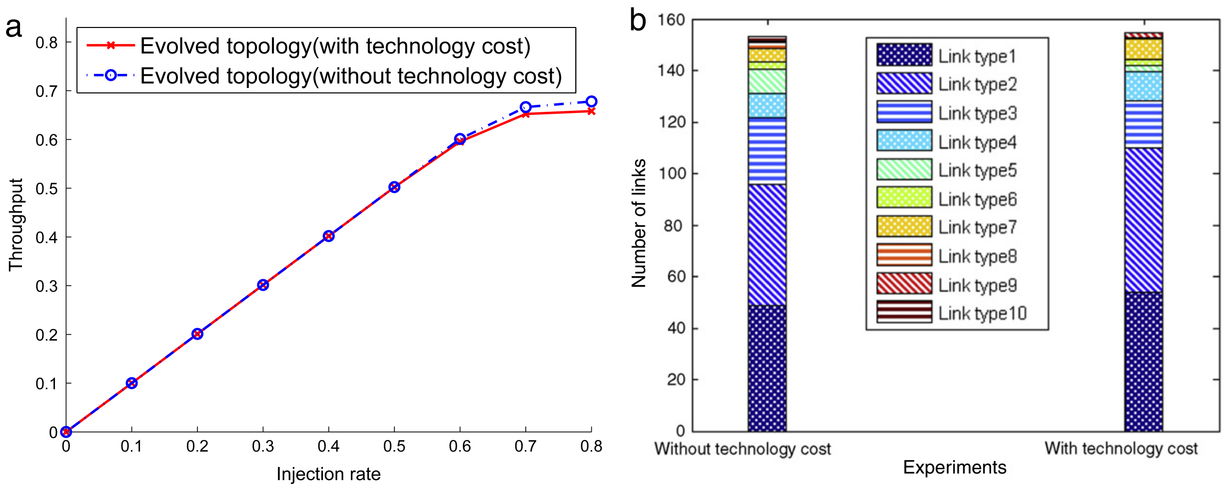


Fig. 22. Performance comparison of networks with and without technology cost. Networks are optimized for *WireCost* and *TP* under uniform random traffic by equally weighting their importance ($w = 0.5$). Injection rate $iR = 0.6$. (a) Network throughput comparison of evolved topologies. (b) Comparison of heterogeneous link type distribution.

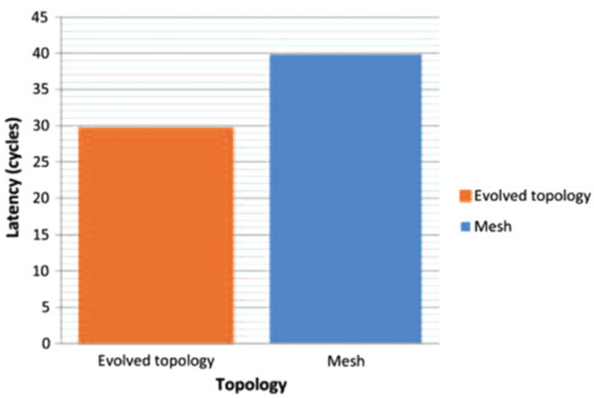


Fig. 23. Performance comparison obtained with gem5 between evolved and mesh topologies.

7. Conclusion

In this paper, we presented the benefits of using ten heterogeneous link types in a generic nanoscale NoC architecture to solve the traditional multi-hop communication problem and to improve the overall network performance. We used an evolutionary framework to evolve optimal networks under various design constraints, traffic patterns, and injection rates. Compared to other work that has been done, we do not construct a network on top of a regular mesh topology. Instead, the evolutionary algorithm can place links without restriction, which allows exploring the entire search space. We used our ten different abstract link types to design an efficient NoC architecture and evaluate and compare the network performance with homogeneous and 2D regular mesh networks. Moreover, we compared the results with the network with three link types, which in current technology could represent metal wires, wireless, and photonic links.

We have shown that our optimal networks based on heterogeneous link types provides higher throughput and lower energy consumption compared to homogeneous link type networks and regular 2D mesh networks under uniform random, hot-spot, and transpose traffic patterns. We also showed that our evolved topologies perform better than networks with three different link types in terms of throughput *TP* and energy *E*. These results suggest that using ten or more link types could lead to even better solutions for nanoscale on-chip communication system than using one or three different link types, as commonly used today.

To test our abstract model against variations in the assumptions we made, we performed simulations with non-linear cost mappings and considered fixed technology cost. Even with these new assumptions, all the outcomes of the article remained valid. This confirms that our modeling abstraction level is appropriate for the type of study we performed and that our results and fundamental and broadly applicable. Simulations done with the GEM5 framework also support all our results.

When long-range links are added to the network, performance and cost both need to be considered. With an increase in system size, the number of long-range links on the network increases to absorb the additional network traffic. This supports an efficient network scalability without significantly reducing system performance.

We have shown that a high performance heterogeneous network fabric can be designed at a low cost by using high-throughput long-range links to communicate between the clustered subnets and low cost short-range links within the cluster subnet. Using a large number of short-range links and placing them at the high traffic areas in the network leads to high network throughput at a lower cost compared to using several longer-range links of the same type.

We conclude by giving brief answers of the questions posed in the introduction. First, our results clearly confirm that heterogeneous link type networks are beneficial compared to homogeneous and regular mesh networks. We also showed that our evolved ten link type topologies provide high throughput and low energy compared to three link type networks. We have provided optimal link type distributions and optimal link placements in the paper. Last but not least, we were able to confirm that the networks scale better compared to regular mesh networks.

The results are relevant for the design of emerging nanoscale on-chip communication fabrics and will help to drive the development of new technology. In particular, it would be desirable to have a broader range of on-chip communication links available, provided one could find appropriate technologies. This, naturally, would pose an increasing challenge on the actual manufacturing and integration, which would need to be addressed.

Acknowledgments

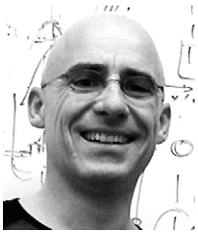
The authors are grateful to Partha Pande and Stephan Müller for the fruitful discussions and feedback.

References

[1] D. Bailey, FFTs in external or hierarchical memory, The Journal of Supercomputing 4 (1) (1990) 23–35.
[2] L. Benini, G. De Micheli, Networks on chips: a new SoC paradigm, Computer 35 (1) (2002) 70–78.
[3] N. Binkert, B. Beckmann, G. Black, S.K. Reinhardt, A. Saidi, A. Basu, et al., The GEM5 simulator, SIGARCH Computer Architecture News 39 (2) (2011) 1–7.
[4] S. Cahon, E. Talbi, N. Melab, ParadisEO: a framework for parallel and distributed biologically inspired heuristics, in: Parallel and Distributed Processing Symposium, 2003. Proceedings. International, 2003, p. 9.
[5] M. Chang, J. Cong, A. Kaplan, M. Naik, G. Reinman, E. Socher, et al., CMP network-on-chip overlaid with multi-band RF-interconnect, in: High Performance Computer Architecture, 2008. HPCA 2008. IEEE 14th International Symposium on, 2008, pp. 191–202.
[6] M. Chang, V. Roychowdhury, L. Zhang, H. Shin, Y. Qian, RF/wireless interconnect for inter-and intra-chip communications, Proceedings of the IEEE 89 (4) (2001) 456–466.
[7] M. Chang, E. Socher, S. Tam, J. Cong, G. Reinman, RF interconnects for communications on-chip, in: Proceedings of the 2008 International Symposium on Physical Design, ISPD'08, ACM, 2008, pp. 78–83.
[8] H. Chung, A. Asnodkar, C. Teuscher, A structural analysis of evolved complex networks-on-chip, in: Proceedings of the Fifth International Workshop on Network on Chip Architectures, NoCArc'12, 2012, pp. 17–22.
[9] H. Dai, Carbon nanotubes: synthesis, integration, and properties, Accounts of Chemical Research 35 (12) (2002) 1035–1044.
[10] S. Deb, K. Chang, A. Ganguly, P. Pande, Comparative performance evaluation of wireless and optical NoC architectures, in: SOC Conference, SOCC, 2010 IEEE International, 2010, pp. 487–492.
[11] S. Deb, A. Ganguly, K. Chang, P. Pande, B. Beizer, D. Heo, Enhancing performance of network-on-chip architectures with millimeter-wave wireless interconnects, in: Application-specific Systems Architectures and Processors, ASAP, 2010 21st IEEE International Conference on, 2010, pp. 73–80.
[12] K. DeJong, Evolutionary Computation, first ed., The MIT Press, 2002.
[13] E. Elbeltagi, T. Hegazy, D. Grierson, Comparison among five evolutionary-based optimization algorithms, Advanced Engineering Informatics 19 (43–53) (2005) ACM ID: 1641013.
[14] M. Fulgham, L. Snyder, Performance of chaos and oblivious routers under non-uniform traffic, Tech. Rep. Univ. of Washington, 1993.
[15] A. Ganguly, K. Chang, S. Deb, P. Pande, B. Belzer, C. Teuscher, Scalable hybrid wireless network-on-chip architectures for multicore systems, IEEE Transactions on Computers 60 (10) (2011) 1485–1502.
[16] International Technology Roadmap for Semiconductors, 2007 edition, Tech. Rep. 2007, http://www.itrs.net/Links/2007ITRS/.
[17] A. Joshi, C. Batten, Y. Kwon, S. Beamer, I. Shamim, K. Asanovic, et al., Silicon–photonic clos networks for global on-chip communication, in: Networks-on-Chip, 2009. NOCS 2009. 3rd ACM/IEEE International Symposium on, San Diego, CA, USA, 2009, pp. 124–133.
[18] K. Kempa, J. Rybczynski, Z. Huang, K. Gregorczyk, A. Vidan, B. Kimball, et al., Carbon nanotubes as optical antennae, Advanced Materials 19 (2007) 421–426.

[19] S. Kumar, A. Jantsch, J. Soininen, M. Forsell, M. Millberg, J. Oberg, et al., A network on chip architecture and design methodology, in: VLSI, 2002. Proceedings. IEEE Computer Society Annual Symposium on, 2002, pp. 105–112.
[20] S. Lee, S. Tam, I. Pefkianakis, S. Lu, M. Chang, C. Guo, et al., A scalable micro wireless interconnect structure for CMPs, in: Proceedings of the 15th Annual International Conference on Mobile Computing and Networking, MobiCom'09, 2009, pp. 217–228.
[21] J. Lienig, K. Thulasiraman, A genetic algorithm for channel routing in VLSI circuits, Evolutionary Computation 1 (4) (1993) 293–311. ACM ID: 1326646.
[22] R. Lu, A. Cao, C. Koh, SAMBA-bus: a high performance bus architecture for system-on-chips, IEEE Transactions on Very Large Scale Integration (VLSI) Systems 15 (1) (2007) 69–79.
[23] Z. Michalewicz, Genetic Algorithms + Data Structures = Evolution Programs, Springer, 1998.
[24] M. Newman, The structure and function of complex networks, SIAM Review 45 (2003) 167–256.
[25] U. Ogras, J. Hu, R. Marculescu, Key research problems in NoC design: a holistic perspective, in: Proceedings of the 3rd IEEE/ACM/IFIP International Conference on Hardware/Software Codesign and System Synthesis, 2005, pp. 69–74.
[26] U. Ogras, R. Marculescu, "It's a small world after all": NoC performance optimization via long-range link insertion, IEEE Transactions on Very Large Scale Integration (VLSI) Systems 14 (7) (2006) 693–706.
[27] P. Pande, C. Grecu, M. Jones, et al., Performance evaluation and design trade-offs for network-on-chip interconnect architectures, IEEE Transactions on Computers 54 (8) (2005) 1025–1040.
[28] A. Shacham, K. Bergman, L. Carloni, On the design of a photonic network-on-chip, in: Proceedings of the First International Symposium on Networks-on-Chip, 2007, pp. 53–64.
[29] A. Shacham, K. Bergman, L. Carloni, Photonic networks-on-chip for future generations of chip multiprocessors, IEEE Transactions on Computing 57 (9) (2008) 1246–1260.
[30] M. Steen, Graph Theory and Complex Networks: An Introduction, Maarten van Steen, 2010.
[31] C. Teuscher, Nature-inspired interconnects for self-assembled large-scale network-on-chip designs, Chaos: An Interdisciplinary Journal of Nonlinear Science 17 (2) (2007) 1–12.
[32] C. Teuscher, A. Hansson, Non-traditional irregular interconnects for massive scale SoC, in: Proceedings of the IEEE International Symposium on Circuits and Systems, ISCAS, 2008, pp. 2785–2788.
[33] D. Watts, S. Strogatz, Collective dynamics of 'small-world' networks, Nature 393 (6684) (1998) 440–442.
[34] S. Woo, M. Ohara, E. Torrie, J. Singh, A. Gupta, The SPLASH-2 programs: characterization and methodological considerations, SIGARCH Computer Architecture News 23 (2) (1995) 24–36.
[35] A. Zhou, B. Qu, H. Li, S. Zhao, P. Suganthan, Q. Zhang, Multiobjective evolutionary algorithms: a survey of the state of the art, Swarm and Evolutionary Computation 1 (1) (2011) 32–49.

Haera Chung is currently a Ph.D. candidate in the Department of Electrical and Computer Engineering, Portland State University, USA. She received her B.S. and M.S. degree in Computer Science Engineering from Korea University and Ewha Women's University, South Korea respectively. Her research interests include heterogeneous network-on-chip architectures and on-chip communications.



Christof Teuscher is an Assistant Professor in the Department of Electrical and Computer Engineering (ECE) with joint appointments in the Department of Computer Science and the Systems Science Graduate Program. He also holds an Adjunct Assistant Professor appointment in Computer Science at the University of New Mexico (UNM). Dr. Teuscher obtained his M.Sc. and Ph.D. degree in Computer Science from the Swiss Federal Institute of Technology in Lausanne (EPFL) in 2000 and 2004 respectively. His main research focuses on emerging computing architectures and paradigms. For more information visit: http://www.teuscher-lab.com/christof.