Optimal Network Topologies and Resource Mappings for Heterogeneous Networks-on-Chip

Haera Chung
Portland State University

Let us know how access to this document benefits you.
Follow this and additional works at: https://pdxscholar.library.pdx.edu/open_access_etds
Part of the Computer and Systems Architecture Commons, and the Systems and Communications Commons

Recommended Citation

10.15760/etd.997

This Dissertation is brought to you for free and open access. It has been accepted for inclusion in Dissertations and Theses by an authorized administrator of PDXScholar. For more information, please contact pdxscholar@pdx.edu.
Optimal Network Topologies and Resource Mappings for
Heterogeneous Networks-on-Chip

by
Haera Chung

A dissertation submitted in partial fulfillment of the
requirements for the degree of

Doctor of Philosophy
in
Electrical and Computer Engineering

Dissertation Committee:
Christof Teuscher, Chair
Douglas V. Hall
Dan Hammerstrom
Xiaoyu Song
Jingke Li

Portland State University
2013
Abstract

Communication has become a bottleneck for modern microprocessors and multicore chips because metal wires don’t scale. The problem becomes worse as the number of components increases and chips become bigger. Traditional Systems-on-Chips (SoCs) interconnect architectures are based on shared-bus communication, which can carry only one communication transaction at a time. This limits the communication bandwidth and scalability. Networks-on-Chip (NoC) were proposed as a promising solution for designing large and complex SoCs. The NoC paradigm provides better scalability and reusability for future SoCs, however, long-distance multi-hop communication through traditional metal wires suffers from both high latency and power consumption. A radical solution to address this challenge is to add long-range, low power, and high-bandwidth single-hop links between distant cores. The use of optical or on-chip RF wireless links has been explored in this context. However, all previous work has focused on regular mesh-based metal wire fabrics that were expanded with one or two additional link types only for long-distance communication.

In this thesis we address the following main research questions to address the above-mentioned challenges: (1) What library of different link types would represent an optimum in the design space? (2) How would these links be used to design an application-specific NoC architecture? (3) How would applications use the resulting NoC architecture efficiently? We hypothesize that networks with a higher degree of heterogeneity, i.e., three or more link types, will improve the network throughput and consume less energy compared to traditional NoC architectures.

In order to verify our hypothesis and to address the research challenges, we design
and analyze optimal heterogeneous networks under different realistic traffic models by considering different cost and performance trade-offs in a comprehensive technology-agnostic simulation framework that uses metaheuristic optimization techniques. As opposed to related work, our heterogeneous links can be placed anywhere in the network, which allows to explore the entire search space. The resulting application-specific networks are then analyzed by using complex network techniques, such as community detection and small-worldness, to understand how heterogeneous link types are used to improve the NoCs performance and cost.

Next, we use the application-specific networks as a target architecture for other applications. The goal is to evaluate the performance of our new NoCs for applications they have not been designed for by finding optimal resource allocations.

Our results show that there is an optimal number of heterogeneous link types for each set of constraints and that networks with three or more heterogeneous link types provide significantly higher throughput along with lower energy consumption compared to both homogeneous link type and regular 2D mesh networks under three different traffic scenarios. Our evolved networks with three different technology-driven link types, namely metal wires, wireless, and optical links, provide 15% more throughput and fourteen times less energy consumption compared to homogeneous link type network. When ten different abstract link types are used in the design, 12% more throughput and 52% less energy consumption are obtained compared to networks with three different technology-driven link types. This shows that heterogeneous NoC designs based on traditional metal wires, wireless, and optical links, occupy a non-optimal spot in the entire design space. Our results further show that heterogeneous NoCs scale up significantly better in terms of performance and cost compared to mesh networks. We uncovered that network
communities evolve robustly and that heterogeneous link types are efficiently establishing inter- and intra-subnet connections depending on their link type properties. We also show that mapping an application on our application-specific NoC architecture provides on average 45% more throughput at 70% less energy consumption compared to regular 2D mesh networks. The NoCs are therefore not only good for the application they were designed for, but for a broad range of other applications as well.

The results of this thesis provide fundamental and novel insights into the design of optimal NoC by using several communication link types. A key outcome is that the level of communication link heterogeneity provided by current technology is non-optimal. A larger choice of link types would allow designers to reach optimal points in the design space that cannot be reached with current technology. The results are relevant for the design of large-scale emerging nanoscale communication fabrics and will help to drive the development of new on-chip communication technology.
Dedication

This dissertation is dedicated to my husband, Hoon Park, whose love, support, and faithful prayer provided me the strength and faith to accomplish my goals.
Acknowledgments

This dissertation would not have been accomplished without the help and support of many people in so many ways. It is my great pleasure to take this opportunity to express my sincere thanks to them.

First and foremost, I would like to express my sincere gratitude to my advisor, Dr. Christof Teuscher for his continuous encouragement, support, guidance and excellent scientific advice throughout the course of this research and the dissertation. He taught me how to be a good researcher and educator, which I will never forget.

I am grateful to my committee members, who are Drs. Douglas V. Hall, Dan Hammerstrom, Xiaoyu Song, and Jingke Li for taking the time to serve as my committee members, and for their valuable suggestions and guidance to improve my dissertation.

I would like to give my special gratitude to Dr. Partha Pande, a professor at Washington State University for the fruitful discussion and feedback on my research.

I am thankful to ECE department faculty and staff for the continuing support to complete my doctoral program. I gratefully acknowledge and appreciate the ECE Graduate Design Automation Scholarship, funded by Mr. Gregory K. Hinckley, President of Mentor Graphics Corporation, which have supported me to focus more on my research.

Thanks to my project team members Allen, John, Gokul, Stephan, Anusha, and Sarah for their feedbacks on my work and also thank past and present Teuscher-lab members for their support.
Most importantly, this dissertation is dedicated in the memory of my grandmother, whom I love and miss most. I would never forget her love and interminable prayer for me. I would also like to dedicate this dissertation to my family for their continuous care and encouragement throughout my study. My most heartfelt gratitude is for my parents, Dr. Ha-Woo Chung and Jung-Ae Lee, who taught me the importance of education and made it possible with their unlimited love, patience, support, and faithful prayer. My deepest gratitude and love goes to my husband Hoon Park for his endless love, prayer, and understanding.

Above all, I would like to thank the God, the Almighty, for His blessings through this journey. I could never have done this without the faith I have in Him.
Table of Contents

Abstract i

Dedication iv

Acknowledgments v

List of Tables xii

List of Figures xiii

1 Introduction 1
  1.1 Motivation .................................................. 1
  1.2 Thesis Problem Statement ................................. 8
  1.3 My Contributions .......................................... 11
  1.4 Organization of the Dissertation ......................... 16

2 Related Work 18
  2.1 Emerging On-Chip Communication Media ................. 18
  2.2 Resource Allocation in Network-on-Chip Architectures .... 22

3 A Framework for Heterogeneous NoC Architectures 26
  3.1 The Network Model ........................................ 26
  3.2 Network Performance Metrics ............................. 27
    3.2.1 Network Wiring Cost ................................ 28
    3.2.2 Network Average Shortest Path ...................... 29
3.2.3 Network Throughput ........................................... 30
3.2.4 Network Energy Dissipation ................................. 30
3.3 Network Traffic Models ......................................... 31
   3.3.1 Uniform Random Traffic .................................. 32
   3.3.2 Hot-Spot Traffic ............................................ 32
   3.3.3 Transpose Traffic ........................................... 33
   3.3.4 SPLASH-2 Benchmarks ................................... 33
3.4 Network Simulation and Representation ....................... 34
   3.4.1 Network Abstraction ....................................... 35
   3.4.2 Node Abstraction ........................................... 35
   3.4.3 Link Abstraction ........................................... 36
   3.4.4 Link Type Abstraction ................................... 36
   3.4.5 Packet Abstraction ....................................... 37
3.5 Finding Optimal Networks ...................................... 37
   3.5.1 Initialization .............................................. 39
   3.5.2 Fitness Function ........................................... 40
   3.5.3 Crossover .................................................. 41
   3.5.4 Mutation .................................................. 42
   3.5.5 Selection .................................................. 43
   3.5.6 Population Size and Evolutionary Runs .................. 43
   3.5.7 Example .................................................. 44

4 Performance Evaluation of NoC Architecture with Three Different Technology-driven Heterogeneous Link Types 47

   4.1 Link Type Definitions ....................................... 47
   4.2 Optimal Number of Links .................................... 49
4.2.1 Networks Without Traffic .......................... 49
4.2.2 Networks with Uniform Random Traffic ................. 52
4.2.3 Networks with Hot-spot Traffic .......................... 57
4.2.4 Networks with Transpose Traffic .......................... 62
4.2.5 Comparison and Discussion ............................ 67
4.3 Optimal Network for Throughput ............................ 71
4.4 Optimal Network for Energy Dissipation ........................ 75
4.5 Network Performance Comparison with Mesh and Homogeneous Networks ........................ 78
4.6 Model Variation ........................................ 83
  4.6.1 Performance Comparison of Heterogeneous Link Type Networks with Homogeneous Link Types ........................ 83
  4.6.2 Performance Evaluation with Different Traffic Scenarios ........................ 85
  4.6.3 Performance Comparison with a Technology-driven Cost Formula ........................ 89
4.7 Performance Evaluation by Using GEM5 ........................ 91
4.8 Discussion ........................................ 94

5 Network Analysis .............................. 96
  5.1 Network Community .................................... 97
  5.2 Network Modularity ................................... 107
  5.3 Small-Worldness ..................................... 108
  5.4 Discussion ........................................ 111

6 Performance Evaluation of NoC Architectures with Ten Different Heterogeneous Link Types ........................ 112
6.1 Link Type Definitions ........................................... 112
6.2 Optimal Number of Links ........................................ 114
   6.2.1 Networks with Uniform Random Traffic ............... 115
   6.2.2 Networks with Hot-spot Traffic ....................... 119
   6.2.3 Networks with Transpose Traffic ..................... 120
   6.2.4 Comparison and Discussion ............................. 121
6.3 Optimal Network for Throughput ............................. 126
6.4 Optimal Network for Energy Dissipation ................. 129
6.5 Network Performance Comparison with Mesh and Homogeneous
   Networks .......................................................... 131
6.6 Scalability ....................................................... 137
6.7 Model Variation .................................................. 142
   6.7.1 Performance Comparison between Linear and Non-linear Cost
         Mapping of the Links ..................................... 142
   6.7.2 Performance Comparison between Different Cost Formula 145
6.8 Discussion ....................................................... 147

7 Core Assignment for Heterogeneous NoC Architectures 148

7.1 Finding Optimal Placement in Heterogeneous NoC Architectures 149
   7.1.1 Initialization .............................................. 151
   7.1.2 Fitness function ......................................... 152
   7.1.3 Genetic Operators ........................................ 153
   7.1.4 Selection .................................................. 155
   7.1.5 Population size and evolutionary runs ............... 156
7.2 Network Performance Comparison with Mesh-based NoC Architecture 157
7.3 Network Performance Comparison of Heterogeneous Evolved Networks with Multi-traffics ........................................ 163
7.4 Discussion .............................................................................. 167

8 Conclusions and Future Work 168
8.1 Conclusions ........................................................................... 168
8.2 Future Work .......................................................................... 171

References 173
List of Tables

4.1 Definition of the three different types of abstract links . . . . . . . . 48

5.1 Small-worldness for evolved topologies. If $S$ is greater than 1, the
network is said to be a small-world network. . . . . . . . . . . . . . 110

6.1 Absolute value of each metric for link type 1, 6, and 10. . . . . . . . 114
6.2 Fixed technology cost for link type 1, 6, and 10. . . . . . . . . . . . 145
List of Figures

1.1 Delay for gates, local wire, and global wire versus feature size [79].
As the transistor size becomes smaller, gate and local wire delay decrease, however, global wire delay significantly increases. This shows the limitation of traditional metal wires for future large-scale SoC designs. ................................................................. 2

1.2 Different on-chip communication architectures. (a) Traditional bus-based communication. (b) Point-to-point link communication. (c) NoC-based communication. (d) 5×5-node mesh-based NoC. CMP = component and R = router. ...................................................... 3

1.3 Examples of two different types of NoC topologies. (a) A standard structure of a 5×5-node mesh. (b) A fully customized structure of 5×5-node. An irregular topology, can be derived by altering the connectivity of a regular structure. Here each node is composed of a core (CMP) and router (R). ...................................................... 6

1.4 Heterogeneous NoCs. (LEFT) Different features of components are integrated in the network. (MIDDLE) Different types of links are used to interconnect between components. (RIGHT) Different types of interconnect fabrics are used for the vertical connections between layers. ................................................................. 9
1.5 Optimal network and mapping design flow for heterogeneous NoCs. The CHCN framework consists of two sub-frameworks, HCNEA and HCNMap framework. The framework finds optimal application-specific heterogeneous NoC architectures and optimally allocate an application to our obtained networks.

2.1 Hybrid NoC architectures. (a) Photonic ring-based hybrid photonic NoC. Source: [10]. Photonic ring interconnect placed on top of a regular mesh network. (b) RF-interconnect NoC. Source: [31]. Z-shaped RF-interconnect is built on top of a regular mesh network. (c) Hybrid wired/wireless NoC. Source: [65]. Wired links are used to communicate intra-components in a sub-network and wireless links are used to communicate between hubs.

2.2 Heterogeneous NoC mapping. Each core in a given application needs to be mapped on a target mesh NoC architecture. This diagram illustrates one possible mapping scenario.

3.1 $4 \times 4$-node network with 1-unit grid spacing. A node consists of a core and router.

3.2 $8 \times 8$-node mesh with a one-unit spacing between nodes. Hot spot nodes are marked with a $\times$.

3.3 Class structure of the Heterogeneous Complex Network (HCN) framework.
3.4 Heterogeneous Complex Network Evolutionary Algorithm (HCNEA) framework design flow. The framework starts from generating a set of initial networks by randomly placing links in a 2D grid structure. Networks are evaluated by using the defined aggregate objective function. Two genetic operators, crossover and mutation, are applied to generate new networks. This process is repeated until the best solution is found.

3.5 Two-point crossover operator.

3.6 Comparison of different population sizing and convergence. We observe that the population size of 600 converged to the minimum fitness around 6,000 generations while a population size of 400 converged around 9,000 generations.

3.7 4×4-node evolved networks. (a) $w = 0$: performance is favored. The result is an almost fully connected network. $WiringCost = 100$, $ASP = 1.6$. (b) $w = 0.5$: wiring cost and performance are equally favored. A hub in node 9 evolves. $WiringCost = 25.4$, $ASP = 1.9$. (c) $w = 1$: wiring cost is favored. The resulting network is a sparsely connected tree. $WiringCost = 15$, $ASP = 3.9$. Only one link type was used for this example.

4.1 Heterogeneous link type distribution as a function of the weight $w$. Depending on the importance of performance versus cost, a different number of each link type is used. A tree evolves when cost is favored ($w = 1$), while a highly connected network with, wiring one link type only, significant hub (node 37) evolves if performance is favored ($w = 0$).
4.2 8 × 8-node evolved networks without traffic. (a) $w = 0.05$: ASP only is important. $WiringCost = 1002.8$, $ASP = 1.9$. (b) $w = 0.5$: wiring cost and ASP are equally important. $WiringCost = 248.6$, $ASP = 2.8$. (c) $w = 1.0$: wiring cost only is important. $WiringCost = 63$, $ASP = 3.6$. Black links: type 1; green links: type 2; red links: type 3.

4.3 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 $iR = 0.6$. High throughput links of link type 2 and 3 are used when we give more weight to $TP$ to absorb the network traffic. When we increase the weight to $WiringCost$, the networks use a smaller number of link type 3 to reduce cost. When $WiringCost$ only is considered, link type 1 only is used in the network.

4.4 8 × 8-node evolved networks with uniform random traffic. (a) $w = 0.05$: $TP$ only is important. High throughput links of link type 2 and 3 are used to improve network performance. $WiringCost = 342.8$, $TP = 0.59$. (b) $w = 0.5$: wiring cost and $TP$ are equally important. Link type 2 is used to keep the throughput high while reducing the network cost. $WiringCost = 249.2$, $TP = 0.58$. (c) $w = 1.0$: wiring cost only is important, therefore, link type 1 only is used in the network. $WiringCost = 63$, $TP = 0.02$. Black links: type 1; green links: type 2; red links: type 3.
4.5  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 $iR = 0.6$. Energy-efficient link of link type 2 and 3 are used to minimize network energy consumption when we give more weight to $E$. When we increase weight to $WiringCost$, the networks use less expensive link of link type 1 and 2 to balance $WiringCost$ and $E$.

4.6  $8 \times 8$-node evolved networks with uniform random traffic. (a) $w = 0.05$: $E$ only is important. $WiringCost = 385.3$, $E = 0.4$. Long-range link of link type 3 are used to reduce the number of hops to lower the network energy dissipation. (b) $w = 0.5$: wiring cost and $E$ are equally important. Link type 2 is used to keep the throughput high while reducing the network cost. $WiringCost = 154.2$, $E = 0.9$. (c) $w = 1.0$: wiring cost only is important, therefore, link type 1 is only used in the network. $WiringCost = 63$, $E = 18.1$. Black links: type 1; green links: type 2; red links: type 3.

4.7  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 $iR = 0.6$. Link type 1, 2, and 3 are all used compared to the evolved networks under uniform random traffic to distribute the network traffic.
4.8 8 × 8-node evolved networks with hot-spot traffic. (a) \( w = 0.05 \): TP only is important. \( WiringCost = 863.1, TP = 0.59 \). (b) \( w = 0.5 \): wiring cost and TP are equally important. \( WiringCost = 598.3, TP = 0.58 \). (c) \( w = 1.0 \): wiring cost only is important. \( WiringCost = 63, TP = 0.02 \). Black links: type 1; green links: type 2; red links: type 3. Hot spot nodes are marked with a ×. High throughput long-range link type 3 is used to communicate between distant nodes located nearby the two hot spots to distribute the network traffic.

4.9 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 \( iR = 0.6 \). Energy-efficient link type 3 is used more frequently for the majority of the weight value \( w \) compared to the evolved networks under uniform random traffic to lower energy consumption.

4.10 8 × 8-node evolved networks with hot-spot traffic. (a) \( w = 0.05 \): \( E \) only is important. \( WiringCost = 902.7, E = 0.2 \). (b) \( w = 0.5 \): wiring cost and \( E \) are equally important. \( WiringCost = 189.3, E = 0.4 \). (c) \( w = 1.0 \): wiring cost only is important. \( WiringCost = 63, E = 3.7 \). Black links: type 1; green links: type 2; red links: type 3. Hot spot nodes are marked with a ×. Long-range energy-efficient link type 3 is used more frequently for the majority of the weight value \( w \) to lower ASP to reduce energy dissipation.
4.11 Heterogeneous link type distribution as a function of the weight $w$.
The networks are optimized for $WiringCost$ and $TP$ with transpose traffic. Injection rate $iR = 0.6$. High throughput long-range link type 3 is more frequently used in the network compared to the evolved networks under hot-spot traffic to absorb the network traffic. 63

4.12 8×8-node evolved networks with transpose traffic. (a) $w = 0.05$: $TP$ only is important. $WiringCost = 955.4$, $TP = 0.59$. (b) $w = 0.5$: wiring cost and throughput are equally important. $WiringCost = 613.4$, $TP = 0.58$. (c) $w = 1.0$: wiring cost only is important. $WiringCost = 63$, $TP = 0.01$. Black links: type 1; green links: type 2; red links: type 3. To support high throughput, a large number of high throughput link type 3 is used in the network. . . . 64

4.13 Heterogeneous link type distribution as a function of the weight $w$.
The networks are optimized for $WiringCost$ and $E$ with transpose traffic. Injection rate $iR = 0.6$. Energy-efficient evolved networks are generated by using a large number of link type 2 compared to the evolved networks under hot-spot traffic. . . . . . . . . . . . . . . . . 65

4.14 8×8-node evolved networks with transpose traffic. (a) $w = 0.05$: $E$ only is important. $WiringCost = 576.3$, $E = 0.3$. (b) $w = 0.5$: wiring cost and $E$ are equally important. $WiringCost = 185$, $E = 0.4$. (c) $w = 1.0$: wiring cost only is important. $WiringCost = 63$, $E = 3.8$. Black links: type 1; green links: type 2; red links: type 3. A large number of link types 2 is used to reduce the network energy consumption compared to the evolved network under hot-spot traffic. 66
4.15 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 $E$. When we give 90% weight to network performance, (1) the evolved networks optimized for $WiringCost$ and $TP$ under hot-spot and transpose traffic use 100% and 83% more links respectively compared to the networks under uniform random traffic to absorb the network traffic, (2) the evolved networks optimized for $WiringCost$ and $E$ under hot-spot and transpose traffic use 14% and 29% more links respectively compared to the networks under uniform random traffic.

4.16 The use of each link type in the network under 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 $E$. Hot-spot traffic use a large number of link type 1 and 3 compared to the evolved networks under uniform random and transpose traffic to increase throughput at a low cost. When network performance is more considered, high throughput energy-efficient link types are used to support the network to improve the performance.

4.17 Network throughput of evolved topology and mesh network 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.
4.18 4 × 4-node mesh-based network with long-range links. Long-range links are inserted where it is most beneficial to improve the network performance. Redrawn from: [121].

4.19 8 × 8-node optimal evolved networks under different traffic scenarios. WiringCost and TP is equally weighted (w = 0.5) using an aggregate objective function: \( f = w \times \text{WiringCost} + (1 - w) \times TP \). Black links: type 1; green links: type 2; red links: type 3. Hot spot nodes are marked with a ×.

4.20 Network energy of evolved and mesh networks with different traffic scenarios. Injection rate \( iR = 0.6 \). All evolved topologies with heterogeneous links use significantly less energy compared to a regular 2D mesh topology.

4.21 8 × 8-node optimal evolved networks with different traffic scenarios. WireCost and E is equally weighted (w = 0.5) using the aggregate objective function: \( f = w \times \text{WiringCost} + (1 - w) \times E \). Black links: type 1; green links: type 2; red links: type 3. Hot spot nodes are marked with a ×.
4.22 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 $w$. Our evolved topologies with heterogeneous link types under uniform random traffic perform significantly better for the majority of the weight value $w$ compared to regular 2D mesh networks. Moreover, when network performance and wiring cost are equally weighted ($w = 0.5$), our evolved network provides 14% more throughput and 17 times less energy consumption compared to a homogeneous link type network.

4.23 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 $w$. Our evolved topologies with heterogeneous link types under hot-spot traffic perform significantly better for the majority of the weight value $w$ compared to regular 2D mesh networks. Moreover, when network performance and wiring cost are equally weighted ($w = 0.5$), our evolved network provides 21% more throughput and 13 times less energy consumption compared to a homogeneous link type network.
4.24 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 \( w \). Our evolved topologies with heterogeneous link types under transpose traffic perform significantly better for the majority of the weight value \( w \) compared to regular 2D mesh networks. Moreover, when network performance and wiring cost are equally weighted \( (w = 0.5) \), our evolved network provides 10% more throughput and 13 times less energy consumption compared to a homogeneous link type network.

4.25 Performance comparison between the networks with three different link types vs. a single link type. (a) The networks are optimized for \( WireCost \) and \( TP \); (b) the networks are optimized for \( WireCost \) and \( E \) under uniform random traffic by equally weighting their cost-performance weight \( (w = 0.5) \). Injection rate \( iR = 0.6 \). Evolved networks with three different technology-driven link types provide higher through and lower energy consumption compared to two different homogeneous networks.

4.26 Performance comparison of networks evolved under random uniform traffic and tested on hot-spot and transpose traffic. (a) network throughput; (b) energy; \( w = 0.5 \). We conclude that our evolved topology is optimized for uniform random traffic, and at the same time outperforms on both hot-spot and transpose traffic patterns.
4.27 Performance comparison of networks evolved under hot-spot traffic and tested on uniform random and transpose traffic. (a) network throughput; (b) energy; $w = 0.5$. We observe that our evolved topology is optimized for hot-spot traffic, and at the same time outperform on both uniform random and transpose traffic patterns.

4.28 Performance comparison of networks evolved under transpose traffic and tested on uniform random and hot-spot traffic. (a) network throughput; (b) energy; $w = 0.5$. We observe that our evolved topology is optimized for transpose traffic, and at the same time outperform on both uniform random and hot-spot traffic patterns.

4.29 Performance comparison of networks with and without technology cost. Networks are optimized for $WireCost$ and $TP$ under uniform random traffic by equally weighting their cost-performance weight ($w = 0.5$). Injection rate $iR = 0.6$. (a) Throughput comparison of the evolved topologies; (b) heterogeneous link type distribution comparison.

4.30 Performance comparison obtained with gem5 between evolved and mesh topologies. Our evolved topology has a 33%, 41%, and 37% lower latency compared to a regular 2D mesh network under FFT, Radix, and LU traffic respectively. The results validate our abstract framework.
4.3.1 Performance comparison obtained with gem5 between evolved and homogeneous topologies. Under FFT, Radix, and LU traffic, our evolved topology with heterogeneous link types has on average a 31% and 11% lower latency compared to a homogeneous network with link type 1 and link type 2 respectively. The results validate our abstract framework.

5.1 Network community structure. Three communities are detected in this network. There are many links between nodes within community while fewer links between communities. This shows that the nodes within community are more densely connected. Redrawn from [117].

5.2 Community structure of the evolved network under hot-spot traffic by using the aggregate objective function: $f = w \times WiringCost + (1 - w) \times TP$. Throughput TP are equally important ($w = 0.5$). Black links: type 1; green links: type 2; red links: type 3. Hot-spot nodes are marked with a $\times$. The thick dash-dotted lines show the communities.

5.3 Link type usage of evolved network optimized for WiringCost and throughput TP with equal importance ($w = 0.5$) hot-spot traffic. (a) Connectivity between and within communities. (b) Wire-length distribution for each link type used between communities and within a community.
5.4 Link type distribution of the evolved network under hot-spot traffic by using the aggregate objective function: \( f = w \times WiringCost + (1 - w) \times TP \). A large number of short-range links of link type 1 and 2 are used within community to absorb the network traffic. A small number of long-range link of link type 3 is used to communicate between communities. ........................................ 102

5.5 Community structure of the evolved network under transpose traffic by using the aggregate objective function: \( f = w \times WiringCost + (1 - w) \times TP \). Throughput \( TP \) are equally important \( (w = 0.5) \). Black links: type 1; green links: type 2; red links: type 3. The thick dash-dotted lines show the communities. ............... 103

5.6 Link type usage of evolved network optimized for \( WiringCost \) and throughput \( TP \) with equal importance \( (w = 0.5) \) transpose traffic. (a) Connectivity between and within communities. (b) Wire-length distribution for each link type used between communities and within a community. ........................................ 105

5.7 Link type distribution of the evolved network under transpose traffic by using the aggregate objective function: \( f = w \times WiringCost + (1 - w) \times TP \). A large number of short-range link of link type 1 and 2 are used within community to absorb the network traffic. A small number of long-range link of link type 2 and 3 are used to communicate between communities. ........................................ 106

5.8 Network modularity of networks optimized for hot-spot and uniform random traffic. The results confirm Clune et al.’s [39] results on biological networks. ........................................ 108

xxvi
5.9 Watts-Strogatz small-world topology can be produced by increasing probability $P$. (a) Regular lattice; (b) small-world network; (c) random graph. Redrawn from: [159].

6.1 Definition of the different types of abstract links.

6.2 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 $iR = 0.6$. All ten different link types are used when we give more weight to $TP$. Short-range links of type 1, 2, and 3 are frequently used over the majority of the weight value $w$ to absorb the network traffic.

6.3 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 $iR = 0.6$. A large number of high performance long-range links of type 8, 9, and 10 are used to reduce energy consumption when we put more weight in $E$. These long-range links reduce the average shortest path length.

6.4 $8 \times 8$-node evolved network with uniform random traffic. (a)-(f) $w = 0.5$: $WiringCost$ and $TP$ are equally important. $WiringCost = 430.2$, $TP = 0.59$. Six link types were used in this network. Among the others, a large number of short-range links of type 1 and 2 and less expensive high throughput links of type 7 are frequently used to distribute the network traffic.
6.5 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 \( iR = 0.6 \). A higher number of high performance links of type 7, 8, 9, and 10 are used to improve the network performance compared to the evolved networks under uniform random traffic. 

6.6 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 \( iR = 0.6 \). A large number of high-performance long-range links of type 9, and 10 are used to reduce energy consumption.

6.7 Heterogeneous link type distribution as a function of the weight \( w \). The networks are optimized for \( WiringCost \) and \( TP \) with transpose traffic. Injection rate \( iR = 0.6 \). A higher number of high performance link types are used in the networks to provide high throughput compared to the evolved networks under uniform random and hot-spot traffic.

6.8 Heterogeneous link type distribution as a function of the weight \( w \). The networks are optimized for \( WiringCost \) and \( E \) with transpose traffic. Injection rate \( iR = 0.6 \). A large number of high performance long-range links of type 9, and 10 are used to reduce energy consumption.
6.9 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 $E$. The evolved networks optimized for $WiringCost$ and $TP$ under transpose traffic use slightly more links compared to the evolved networks under uniform random and hot-spot traffic to absorb the network traffic. The total number of links used in the evolved networks optimized for $WiringCost$ and $E$ under three different traffics are similar. This suggests that our results are independent of the traffic scenarios.

6.10 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 $E$. For weights $w$ between 0.2 and 0.5, the evolved networks optimized for $WiringCost$ and $TP$ use an average of 5.5 link types while the networks optimized for $WiringCost$ and $E$ use an average of 2 link types. Energy-optimized networks use high performance links of type 9 and 10 to reduce the energy consumption. When we give more weight to $WiringCost$, a lower number of various link types are used to lower the energy.
6.11 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. The evolved networks with ten different link types improve the throughput by 12%, 7%, and 17% under uniform random, hot-spot, and transpose traffic respectively compared to the networks with three different technology-driven link types.

6.12 Network energy of evolved topologies and mesh networks with different traffic scenarios. Injection rate $iR = 0.6$. The evolved networks with ten different link types show 66%, 25%, and 66% less energy consumption under uniform random, hot-spot, and transpose traffic respectively compared to the networks with three different technology-driven link types.
6.13 Performance comparison of heterogeneous link types 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 $w$. Our evolved topologies with heterogeneous link types under uniform random traffic perform significantly better for the majority of the weight value $w$ compared to regular 2D mesh networks. Moreover, when network performance and wiring cost are equally weighted ($w = 0.5$), our evolved network provides 14% more throughput and 17 times less energy consumption compared to a homogeneous link type network and the same or better performance compared to a three link type network. . . . . . . . . . . . . . . . . 133

6.14 Performance comparison of heterogeneous link types 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 $w$. Our evolved topologies with heterogeneous link types under hot-spot traffic perform significantly better for the majority of the weight value $w$ compared to regular 2D mesh networks. Moreover, when network performance and wiring cost are equally weighted ($w = 0.5$), our evolved network provides 21% more throughput and 13 times less energy consumption compared to a homogeneous link type network and the same or better performance compared to a three link type network. . . . . . . . . . . . . . . . . 134
6.15 Performance comparison of heterogeneous link types 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 $w$. Our evolved topologies with heterogeneous link types under transpose traffic perform significantly better for the majority of the weight value $w$ compared to regular 2D mesh networks. Moreover, when network performance and wiring cost are equally weighted ($w = 0.5$), our evolved network provides 10% more throughput and 13 times less energy consumption compared to a homogeneous link type network and the same or better performance compared to a three link type network.

6.16 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. Our evolved networks outperform regular 2D mesh networks even on larger systems. However, with larger systems, the performance is not up to the level of smaller systems because the networks evolved with 50% weight on cost limits the use of long-range links.
6.17 Scalability for network sizes under hot-spot traffic. WireCost and TP is equally weighted \((w = 0.5)\) using the aggregate objective function: \(f = w \times \text{WiringCost} + (1 - w) \times TP\). For the network throughput per cost \((TP/WireCost)\), our evolved topologies perform even better than 2D regular mesh networks for large-scale networks.

6.18 Scalability for network sizes under hot-spot traffic. WireCost and \(E\) is equally weighted \((w = 0.5)\) using the aggregate objective function: \(f = w \times \text{WiringCost} + (1 - w) \times E\). For the network energy per cost \((E/WireCost)\), our evolved topologies have lower energy requirements compared to 2D regular mesh networks and scale better.

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

6.20 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. Even though the cost function has changed, the number of link types used is the same, but 1% throughput improvement is shown.
6.21 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. One less link type is used compared to the evolved network without technology cost. The network throughput decreased by 1%.

7.1 Application core assignment on a target mesh NoC topology. This diagram illustrates one possible mapping scenario. Five cores in the application are mapped to \(3 \times 3\)-node mesh network.

7.2 Design flow of the Heterogeneous Complex Network Mapping (HC-NMap) framework. Our evolutionary algorithm platform is based on the ParadisEO framework [30] and we combine this framework with our own C++ HCN framework to evaluate the network metrics.

7.3 Possible initial solutions of the NoC mappings for the Figure 7.2. The nodeID represents the ID of a target NoC node.

7.4 General one-point crossover operator. A random crossover point is selected within a chromosome, then the chromosomes beyond this point are interchanged between the two parents to generate new children.

7.5 New crossover operator inspired by [133].

7.6 General mutation operator. Node ID 1 and 6 are selected and genes 2 and 8 are interchanged to generate a new child.
7.7 Fitness as a function of generation. Using a population of 60, convergence occurs in an average of 7,500 generations. 156

7.8 Communication graph for the 38 cores TVOPD. Node and weight of the edge represents core and the communication bandwidth respectively. Redrawn from: [114]. 158

7.9 $8 \times 8$-node optimal evolved network for TVOPD application. $WiringCost$ and $TP$ is equally weighted ($w = 0.5$) using an aggregate objective function: $f = w \times WiringCost + (1 - w) \times TP$. Black links: type 1; green links: type 2; red links: type 3. 159

7.10 Performance comparison of heterogeneous NoC and mesh topologies as target NoCs. The TVOPD application is optimally mapped on two different target NoC architectures, i.e., evolved heterogeneous and mesh networks. $w = 0$ means that throughput only is considered. $w = 1$ means that energy only is considered. All application-specific NoC architectures with heterogeneous link types have a near 48% throughput improvement as well as 100% less energy consumption compared to regular mesh NoC topologies. $WiringCost$ for the evolved network is 207.66. 161
7.11 Performance comparison of heterogeneous NoC and mesh topologies as target NoCs. The SPLASH-2 benchmark FFT application is mapped on two different target topologies, i.e., evolved heterogeneous and mesh networks. When throughput only is considered \((w = 1)\), the evolved network provides 90% more throughput compared to a mesh network. Energy-aware evolved network consume 57% less energy than a mesh network. The \(WiringCost\) for the evolved network is 188.70.

7.12 Performance comparison of two different applications mapped on the application-specific networks. Two applications, TVOPD and FFT, are mapped on three different NoC architectures. The results show that the application-specific NoC architectures support the best performance for the given application.

7.13 Performance improvement of the obtained heterogeneous network evolved under two different application traffics for given applications. Multi-traffic based evolved NoC architecture provides higher throughput and lower energy dissipation, however, the network becomes more expensive. The \(WiringCost\) for the evolved network is 256.93.
1

Introduction

1.1 Motivation

Current advanced integrated circuit (IC) technology enables to integrate a large number of cores (or IP blocks) on a single chip [14, 122]. With an increasing number and diversity of IP blocks, the communication between them becomes increasingly important and challenging [72, 73, 128]. One of the important features in multi-cores Systems-on-Chips (SoCs) design is the interconnect fabric [115]. According to ITRS roadmap [79], gate and local wire delay decrease as the transistor size becomes smaller, however, global wire delay significantly increases in future SoC designs as shown in Figure 1.1. This is because traditional metal wires do not scale in wire length [73], i.e., the smaller you make them, the slows the communication. Inserting repeaters on an interconnect can reduce the global wire delay, however, these will decrease the design performance in terms of energy (or power) and area. The on-chip interconnect architectures, thus directly affects the overall SoC’s performance.

To support the SoC’s design performance requirements, novel on-chip communication architectures were investigated over the last few years. Two types of traditional SoC interconnect structures are introduced, namely shared-bus-based and point-to-point communication structures [94, 95]. Bus-based communication carries only one communication transaction at a time (see Figure 1.2 (a)). This architecture is simple to design and has a low area cost, however, it limits the
Figure 1.1: Delay for gates, local wire, and global wire versus feature size [79]. As the transistor size becomes smaller, gate and local wire delay decrease, however, global wire delay significantly increases. This shows the limitation of traditional metal wires for future large-scale SoC designs.

communication bandwidth and scalability and also consumes large amounts of energy [96, 101]. Point-to-point (P2P) communication, which directly interconnects between components, is the most effective network architectures (see Figure 1.2 (b)). This architecture provides high bandwidth, low latency, and low energy consumption. However, as the number of components increases, the number of connections significantly increases as well [40,96]. For example, three components require three connections and five components require ten connections (see Figure 1.2 (b)). The number of connections required for $N$ components are $N(N - 1)/2$. Therefore, such architectures are not suitable for future large-scale SoCs.

In recent years, Network-on-Chip (NoC) structures (see Figure 1.2 (c) and (d)) were proposed as a promising solution for designing large and complex SoCs [18,44, 45,82,93]. A general NoC architecture consists of cores (IPs), switches (routers),
Figure 1.2: Different on-chip communication architectures. (a) Traditional bus-based communication. (b) Point-to-point link communication. (c) NoC-based communication. (d) $5 \times 5$-node mesh-based NoC. $CMP = \text{component}$ and $R = \text{router}$. 
and interconnects (links). Each core sends data packets to another core through switches and interconnects. Figure 1.2 (c) shows an example of NoC-based network that has five components. As one can see, six links are needed to communicate between the components through the two routers while ten links are needed for the P2P architecture (see Figure 1.2 (b)). Therefore, this NoC paradigm provides better scalability and reusability and provides thus an alternative to traditional bus-based and point-to-point communication architectures for future SoCs [82,96].

Many different NoC topologies, such as Mesh [54], Torus [45], SPIN [2], Octagon [87], and Tree [126] have been introduced. Figure 1.2 (d) shows an example of 5×5-node 2D mesh NoC architecture. A mesh-based topology is considered by many designers because the network structure is simple and easy to implement [125].

Despite these NoC benefits, there are still several limitations in traditional NoC architectures.

First, conventional metal wire interconnects limit the communication performance of NoCs because of their multi-hop nature for long-range on-chip communication [34]. Multi-hop communication causes high latency and power dissipation [65,118], which can lead to the interconnect consuming 80% of the total chip power [80]. The ITRS roadmap also states that “[t]he 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. Several new interconnect technologies, such as photonic interconnects [86,143,145], multi-band RF NoC [31,33,41], Carbon NanoTubes (CNTs) [42,89], and millimeter wave wireless (mmWave) [49,50,172] were proposed and evaluated in the past. However, these evaluations were
done using a specific performance metric, such as latency, throughput, or power. In reality, however, trade-offs are key and what may be optimal for one metric may not be optimal for all the others. For example, we can obtain higher performance by adding more wires to a NoC, but that will increase the wiring cost and the power consumption. It is an open problem how to use new NoC interconnect technologies in an optimal way [20, 104, 120, 123, 137].

**Second**, using a single type of link for on-chip communication may not satisfy the performance requirements. As many different types of components (heterogeneous IP cores), such as diverse processors (CPUs, GPUs, and DPSs), memories, reconfigurable units, I/O devices, etc., can be integrated on a single chip, designing NoCs-based systems that support the network performance requirements becomes even more complex. Different bandwidth and speed may require to communicate between different pairs of components to meet the network performance. To manage heterogeneity and support different network requirements, such as performance, energy, and memory size of components in the design, heterogeneous link types are needed for future on-chip interconnect communication (e.g., embedded and mobile platforms).

**Third**, current standard interconnect topologies (see e.g., Figure 1.3 (a)) namely mesh, torus, and octagon, etc., used to implement various multi-processor systems are not suitable for future large-scale SoC designs. This is because such standard topologies may suffer from high latency and power consumption due to multi-hop communication as the number of cores per single chip increases. However, the network performance can be improved by adding long-range links in the network [121]. An example of application-specific customized network is shown in Figure 1.3 (b). Here each node consists of a core and router. As one can see, long-range links are
added to improve the network performance and irregular structure is obtained to reduce the network cost. When a network is optimized for one or more metrics, the obtained customized network guarantees the best performance.

Figure 1.3: Examples of two different types of NoC topologies. (a) A standard structure of a 5 × 5-node mesh. (b) A fully customized structure of 5 × 5-node. An irregular topology, can be derived by altering the connectivity of a regular structure. Here each node is composed of a core (CMP) and router (R).

Each design has different requirements and constraints. For example, high quality data visualization applications requires a high bandwidth and wireless communication systems and portable devices require low power consumption. Real-time voice streaming requires a low latency, but not necessarily a high bandwidth. As long as the traffic patterns can be statically analyzed, on-chip network can be customized by the behavior of specific target application. Application-specific
NoC approach has been presented in [3, 16, 74, 110, 150] and shown to have performance benefits. Also, many researchers have presented specific NoC architectures, such as SPIN [2, 70], ÆHEREAL [68, 136], QoS [59], QNOC [24, 138], ANOC [15], MANGO [25, 26], SOCBUS [162], PROTEO [147], NOSTRUM [107], HERMES [109], XPIPES [19, 43], and CHAIN [12]. All of these NoCs have different features and design requirements. QNOC, ANOC, PROTEO, MANGO and CHAIN are implemented using clockless circuits while ÆHEREAL, XPIPES, and NOSTRUM are implemented using synchronous circuits. HERMES, NOSTRUM, and SOCBUS are based on 2D mesh topologies but SPIN is based on a fat tree topology. NOSTRUM, SOCBUS, and SPIN support scalable architectures for homogeneous system and XPIPES and CHAIN are a platform-based design that support heterogeneous component systems. Therefore, it is important to build customized application-specific NoC architectures with heterogeneous link types to satisfy the communication constraints of the design.

Fourth, tools for NoC implementation are essential for efficient NoC-based multi-processor system design and evaluation. Several research groups have studied and developed automatic tools, such as the XpipesCompiler [81, 113] and the Communication Synthesis Infrastructure (COSI) [130–132] for the generation of optimal networks based on libraries of components, routers, and interfaces. However, these are essentially network synthesis tools that generate a floorplan and the synthesizable NoC components and do not support our proposed NoC architecture with heterogeneous link types features.

To solve the current NoC architecture limitations, we need to evaluate the value of the different types of new interconnect links by implementing an application-specific heterogeneous NoC architecture and compare the network performance
with mesh and homogeneous NoC architectures. Therefore, we need a new design and evaluation framework to implement an optimal heterogeneous NoC architecture that has multiple different link types. This will allow us to further explore the benefits of using heterogeneous link types in the networks.

Heterogeneous NoC architectures can be implemented as one of three main categories: (1) a network with different components, (2) interconnect fabrics, and (3) vertical interconnect access structures. This is shown in Figure 1.4. In the last few years, researchers have studied and presented NoC architectures with different types of components [88, 129, 163], interconnect fabrics [50, 65, 121], and structure of vertical interconnect access [58, 169, 170]. In this dissertation, we focus on finding optimal heterogeneous NoC architectures with multiple different interconnect fabrics for different performance aspects. The difference from previous work, e.g., [50, 121], is that our optimal heterogeneous NoC architectures do not start with a standard mesh topology to reduce network cost. To the best of our knowledge, no one has explored an irregular NoC architecture with three or more different link types.

1.2 Thesis Problem Statement

The goal of the dissertation is to introduce a practical evaluation methodology that can help NoC designers to find optimal solutions in these high-dimensional design spaces. First, we find optimal heterogeneous NoC architectures with three or more different link types by using an Evolutionary Algorithm (EA) [4, 9, 55], a technique that is most suitable for multi-objective optimization problems. Second, we find optimal placement and routing strategies for a given optimal network and evaluate the network performance.
In this dissertation, we propose to design and analyze optimal nanoscale NoCs with heterogeneous links to answer the following research questions:

- Are heterogeneous link type networks beneficial compared to homogeneous networks and regular 2D meshes?

- How many heterogeneous link types are optimal?

- What are 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?
• Do irregular structure interconnection networks with heterogeneous link type scale better than regular mesh networks?

• Do sub-networks and small-world networks evolve with heterogeneous link type?

• How can real applications be mapped optimally on our proposed heterogeneous interconnection network fabrics?

• Is our proposed application-specific fabric beneficial for real applications compared to regular mesh networks?

We answer these questions by developing a comprehensive software framework, called CHCN (Combined Heterogeneous Complex Network), for the design, optimization, and evaluation of complex heterogeneous NoC networks as shown in Figure 1.5. In CHCN framework, two sub-frameworks, HCNEA (Heterogeneous Complex Network Evolutionary Algorithm) framework (see Chapter 3) and HCNMap (Heterogeneous Complex Network Mapping) framework (see Chapter 7) are provided. The HCNEA framework is used to find an optimal NoC architecture with heterogeneous link type and to evaluate the performance of optimized 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 is obtained, we evaluate the network under uniform [23,157] and non-uniform (e.g., hot-spot [64,71,140] and transpose [46,168]) synthetic traffic patterns as well as real application, such as the SPLASH-2 benchmarks [75,148,165] to show the benefits of using heterogeneous link types in the networks. We then use the HCNMap framework to tackle the problem of optimal
core assignment for heterogeneous NoC based on given system. This is of particular importance for emerging reconfigurable circuits that may offer heterogeneous NoCs, but will have to accommodate a wide range of different applications.

Figure 1.5: Optimal network and mapping design flow for heterogeneous NoCs. The CHCN framework consist of two sub-frameworks, HCNEA and HCNMap framework. The framework finds optimal application-specific heterogeneous NoC architectures and optimally allocate an application to our obtained networks.

1.3 My Contributions

In this dissertation, we presented methodologies to solve open NoC design problems and find optimal network fabrics for application specific heterogeneous multiprocessor systems. We proposed irregular structure interconnect optimal NoC
architectures with heterogeneous link types. Each link type was characterized by a set of specifications, which include performance and cost measures. We compared the performance of heterogeneous to homogeneous networks under different realistic traffic models and showed that our proposed heterogeneous architectures outperform homogeneous and mesh architectures in performance, energy, and throughput. In addition, we also demonstrated that our proposed interconnect architectures scale better than regular 2D mesh networks.

Our framework is based on technology-agnostic and abstract link types, which can however be mapped directly with actually technology. The benefits of a technology-agnostic approach is the broad applicability of our fundamental results. Moreover, we proposed to design and analyze optimal NoCs with a much larger link library to show that the link types available with current technology are non-optimal in a heterogeneous setup. The research is relevant for the design of emerging nanoscale communication fabrics and will help drive the development of new technology.

In addition, we developed evolutionary optimization method to place a target application on our evolved heterogeneous fabrics and compared the performance of network to regular mesh network to show that the application specific NoC architecture platform is needed to meet the application requirements.

The contributions of this work are as following:

1. We introduce heterogeneous link types in NoCs to solve the current NoC multi-hop wired communication problems and significantly improve the network performance in terms of throughput and energy at a low cost. To the best of our knowledge, no one has thoroughly evaluated heterogeneous networks with three or more different kinds of interconnect technologies in a
comprehensive framework that can deal with several design constraints.

2. We introduce an irregular interconnect structure to design a low-cost and high-performance heterogeneous NoC architectures. We show the performance improvement at a low cost compared to a regular mesh networks. To the best of our knowledge, no one has proposed irregular heterogeneous NoC fabrics.

3. We present algorithmic methods to evaluate each different technology-driven 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. Our technology-agnostic approach allows us to obtain optimal networks for a broad range of current and future NoC interconnect technologies. This work is submitted for publication in full in [37].

4. We present evolutionary optimization techniques to obtain optimal NoC topologies. Our networks significantly outperform homogeneous and regular networks, and we can therefore say that networks with heterogeneous link types built from current technology are very promising because they can cover points in the design space that homogeneous networks cannot reach. This work is submitted for publication in full in [37].

5. We analyze the network topologies from a complex network perspective to determine the community structure and the small-worldness. Our results show that a hierarchical community structure evolves and that most networks have the small-world property. Both the community structure and the small-world property mean that such networks are more scalable than traditional
topologies, such as a 2D mesh. This work is submitted for publication in brief in [37] and has been published in full in [35].

6. We present evolutionary optimization techniques to obtain optimal NoC topologies with a much larger heterogeneous link library. The networks significantly outperform current technology-driven link types, homogeneous, and regular mesh networks. These results confirms that the link types with current technology are non-optimal for heterogeneous nanoscale on-chip communication architectures. This work is submitted for publication in full in [36].

7. We show that the obtained optimal NoC topologies use a small number of long-range links to communicate between the cluster subnets and a large number of short-range links within the bluster subnet to further improve the network performance while lowering the cost. This work is submitted for publication in full in [36].

8. We present algorithmic methods to solve NoC mapping problems. The novel mapping methods allow analyzing the application behaviors and the target network fabric configurations and finding the optimal mapping solution. This allows us to find the best solution for placing multi-cores applications in an irregular heterogeneous NoC structures while optimizing the network performance metrics.

9. We present a novel heterogeneous complex network optimization framework to evaluate the networks and find the best optimal solution for heterogeneous network. Until recently, there is no single simulation tool that can (1) explore the complex heterogeneous network space and find the best optimal network fabric and (2) explore the different tradeoffs between network performance
and cost metrics. This helps NoC designers to build efficient heterogeneous interconnection network fabrics for his or her applications.

10. I have published the following papers in the area:

- Journals:
  
  

- Conferences:
  
  
  (b) C. Teuscher, H. Chung, A. Grimm, A. Amarnath, and N. Parashar. The power of power-laws: Or how to save power in SoC. In International Green Computing Conference and Workshops (IGCC ’11), pages 1-6, July 2011.
  
1.4 Organization of the Dissertation

The rest of the dissertation is organized in seven chapters.

Chapter 2 reviews the current emerging interconnect fabrics proposed by many research groups for efficient communication of future large-scale NoC architecture and existing methodologies used for mapping topology on NoC platforms.

Chapter 3 describes the basic heterogeneous architecture of our framework, defines network performance metrics, and introduces the network traffic scenarios that is used in this dissertation.

Chapter 4 presents the benefits of using heterogeneous link types for NoC designs. We obtain optimized heterogeneous NoC architectures by using three different technology-driven link types and evaluate and compare the network performance in terms of throughput and energy with homogeneous and 2D regular mesh networks.

Chapter 5 analyzes the structure of the obtained heterogeneous NoC architecture by using tools from complex network science, such as community detection and small-worldness, to understand how heterogeneous link types are placed to improve the network performance.

Chapter 6 presents NoC architectures by using ten different abstract link types. We show that optimized networks with ten different link types outperform not only homogeneous and regular mesh networks but also networks with three different...
technology-driven link types.

Chapter 7 describes details of the basic architecture of our core assignment framework and shows that application mapping onto application-specific optimized NoC architecture with heterogeneous link types supports better network performance compared to mesh NoC architectures.

Chapter 8 concludes the dissertation and presents future work.


Related Work

2.1 Emerging On-Chip Communication Media

Network-on-Chip (NoC) have been introduced as a promising solution for scalable communication of future large-scale multi-cores systems and to support the performance and power requirements of future applications. Traditional NoC architectures are based on packet-switching networks. Each packet goes through switches/routers and interconnect links, thus generating large amounts of multi-hop long-range communication. This can significantly affect the network performance and power consumption [105].

In the last few years, several solutions were proposed to improve the network performance. Ogras and Marculescu [121] have proposed inserting a few long-range links to standard mesh NoC topologies. 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. However, the authors did not consider cost and scalability in their paper.

Using existing metal interconnects will be highly inefficient and become harder to satisfy the design requirements, such as power, delay, and reliability for future large-scale electronic systems [34, 151, 152]. This is because as interconnects are scaled due to the increase in the number of components per single chip, their resistance increases. Therefore, using metal as a long-range link will be challenging for interconnect delay and power dissipation [80]. Researchers have recently proposed different emerging interconnect technologies to be used as NoC communication
Photonic interconnects were introduced as a promising new technology for NoC communication [29, 72, 86, 145, 171]. A few recent works have shown that photonic NoCs provide lower latency, lower power, and higher bandwidth compared to wired interconnects [86, 99, 119, 143, 144]. Photonic networks can support transmitting large amount of data across long distances at low latencies [10, 90]. For example, Bahirat and Pasricha [10] introduced photonic ring-based hybrid photonic NoC architecture. The photonic ring waveguide, used as a global on-chip communication channel, is built on top of a regular 2D mesh network to improve the latency and reduce power dissipation (see Figure 2.1 (a)). However, the components are expensive and there are still open problems that need further investigation, e.g., photonic NoCs require to integrate true speed on-chip light sources [21, 72], high-radix photonic switches [145], and thermal management [160].

Another promising interconnect technology is multi-band RF interconnects (RF-I) introduced by Chang et al. [31–33]. They considered various implementation challenges of hybrid NoC architecture using RF-I technology (RF transmission line) laid in a zig-zag pattern on top of a wired mesh (see Figure 2.1 (b)) and demonstrated the performance achievements in [31]. The benefit of using RF interconnect is that it transmits electromagnetic (EM) waves, which travel at the effective speed of light along the wires. The RF interconnects provide fast data transport over long distances. The work shows that higher bandwidth and lower latency can be obtained by using this technology [31].

Floyd et al. [60] first introduced on-chip wireless interconnects as a new interconnect technology for clock distribution. Zhao and Wang [173] proposed CMOS
Figure 2.1: Hybrid NoC architectures. (a) Photonic ring-based hybrid photonic NoC. Source: [10]. Photonic ring interconnect placed on top of a regular mesh network. (b) RF-interconnect NoC. Source: [31]. Z-shaped RF-interconnect is built on top of a regular mesh network. (c) Hybrid wired/wireless NoC. Source: [65]. Wired links are used to communicate intra-components in a sub-network and wireless links are used to communicate between hubs.
ultra-wideband (UWB) wireless technology for high-data rate low-power short-range communication. Lee et al. [97] 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. In Deb et al. [50], the authors introduced a design of a hierarchical small-world network with long-range and low power mm-wave wireless links for NoC (mWNoC) and showed a performance improvement for both uniform and non-uniform traffics. Ganguly et al. [65] introduced a new on-chip antenna based on Carbon NanoTubes (CNTs) for on-chip wireless communication (see Figure 2.1 (c)) and evaluated latency, throughput, and energy dissipation. The network is divided into small sub-networks and uses wireless links to communicate between sub-networks through hubs located in each sub-network while wired links are used to communicate between the components within the sub-network. 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. Although the RF/wireless interconnects have many advantages, this new technology is still in an early stage of evolution. It needs more research to solve open problems, such as packaging and interference issues, ultra-high frequency requirements, efficient on-chip antennas, and power overhead.

Each of the above mentioned emerging interconnect technologies has several issues and problems that need to be investigated. However, in the past few years,
a few researchers have studied and shown some promising initial results that they may be suitable candidates for global on-chip interconnects in the future large-scale multi-core systems. 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 only compared the results with a regular 2D mesh network and did not in general compare the networks with each other. In addition, their networks are based on regular mesh. Teuscher [154,155] showed that unstructured nature-inspired NoCs can have benefits for performance and scalability over traditional structured NoCs.

2.2 Resource Allocation in Network-on-Chip Architectures

As current semiconductor technology allows integrating heterogeneous multi-cores on a single chip, finding an optimal resource allocation for specific applications to a given NoC architecture is an important step, but also one of the difficult problems in NoC design [7,17,20,80,104,111,120]. This is because each core of the application needs to be placed to a node in a target NoC architecture as shown in Figure 2.2, and the network performance will vary depending on where cores are allocated.

The NoC mapping and routing algorithms and techniques have been presented by various researchers. Hu and Marculescu [77] present a branch-and-bound algorithm, which builds a search tree and repeats branch and bound steps to find the least energy consuming leaf node for mapping cores onto a regular mesh-based NoC architecture. Zhou et al. [175] use a genetic algorithm for random core mapping on a 2D mesh NoC architecture to optimize the network delay and Lei and
Kumar [98] propose a two-step genetic algorithm, namely partitioning and embedding steps to map applications onto a regular mesh NoC structures to minimize the run time. In the first step, for simplification, each core is placed to a node in the network by using average link delay as a constant value. In the next step, the real delay is used to find an optimal mapping solution. They showed that their proposed method performs well with large-size networks. In Lu et al. [102], a clustering technique is applied with simulated annealing to obtain better initial mapping with larger networks in less run time. Their algorithm first partition network nodes into groups according to the actual distance between nodes as well as communication cores into groups based on the number of connections and bandwidth of each connection. Then, they use simulated annealing to find optimal
core-to-node mapping solution. Koziris et al. [91] presented a physical mapping algorithm (PMAP) that maps the most highly connected node between application and target network node to reduce the communication overhead for graph tasks and Murali et al. [111,112] proposed a fast mapping algorithm (NMAP) that splits traffic under bandwidth constraints to minimize communication delay. In Shen et al. [146], a new efficient binomial mapping and optimization algorithm (BMAP) is presented to optimize the hardware cost by reducing the total network traffic. The complexity of synthesis is reduced $O(N^2)$ time, where $N$ is the total number of IPs, compared to NMAP. However, all these papers are optimized for single performance metric, which may not be optimal for other metrics. In a real world application, if network performance is the only metric considered, it may consume more power than necessary by performing better than needed. Similarly, when power is minimized, performance requirements for a given application may not be met [13,92].

To solve this problem, multi-objective optimization techniques are generally used. Ascia et al. [5] presented a multi-objective genetic algorithm for NoC mapping that finds the Pareto-optimal solution between power consumption and performance. Ascia et al. [6] also presented Pareto-based Branch-and-Bound (PBBB) approach to solve multi-objective optimization problems for a mesh-based NoC mapping. Their algorithm minimized the network latency and the power dissipation. In Jena and Sharma [84], a multi-objective genetic algorithm (MGAP) is proposed to explore mesh-based NoC mapping to maximize the throughput and minimize the power. Their algorithm has two steps, 1) the task assignment problem (TA-GA) and 2) the core-tile mapping (CT-GA) to find an optimal solution. Power consumption is minimized in TA-GA step, and distance between cores is reduced.
to increase the network throughput in CT-GA step. Sepulveda et al. [141, 142] presented an evolutionary approach Multi-objective Adaptive Immune Algorithm (MAIA) to solve regular 2D mesh NoC mapping problem. The results show that this algorithm reduces latency and power dissipation compared to multi-objective genetic-based (MGAP) and branch-and-bound (PBBB) algorithm. In Srinivasan and Chatha [149], a genetic-algorithm-based technique is presented to obtain optimal mapping that optimized power and area. The power and energy consumption can be reduced by minimizing the total number of routers used. Radu and Vintan [135] introduced an optimized simulated annealing (OSA) algorithm to solve the NoC mapping problem. They found that the runtime speed of OSA is faster than a general simulated annealing algorithm without losing the quality of the mapping solution.

While new optimization techniques and algorithms for NoC mapping are presented by many authors, they are proposed specifically for a regular 2D mesh structure. To the best of our knowledge, no one has tried to map an application onto customized irregular topology with heterogeneous link types. To do so, new multi-objective optimization techniques for irregular topology NoC mapping are needed. Since the Pareto-optimal mapping of IP cores onto a NoC architecture by considering performance and power consumption is an NP-hard problem [66], we use well-known evolutionary algorithms, which have been successfully used to solve NoC mapping problems in the past.
A Framework for Heterogeneous NoC Architectures

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

3.1 The Network Model

Graph theory has been commonly used to design and analyze network topology [52, 153]. This is because many problems can be represented by graphs, and behavior of complex networks can be efficiently analyzed. In this dissertation, we use a graph-theoretic approach to represent our networks.

A graph $G$ is denoted by $G = (N, E)$, where $N$ is a set of nodes and $E$ is set of edges. A node represents a core and router while 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 $4 \times 4$-node 2D mesh network is shown in Figure 3.1 (a). 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 in a single hop.

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. All network nodes are arranged in a 2D grid with a one-unit grid spacing (see Figure 3.1).
3.2 Network Performance Metrics

In order to analyze and evaluate the performance of evolved network with heterogeneous link types (i.e., different types of links), we need to measure the performance metrics that are of interest. We considered network performance metrics such as the wiring cost, the average shortest path length, the throughput, and the energy dissipation. In this section, we will provide brief definitions of these metrics, which are inspired by [127]. For our purpose, we do not consider packet lengths or flits. We simply consider a packet as an abstract unit that is communicated on the network. However, to validate our results, we also ran additional performance simulations in Section 4.7 by using the GEM5 framework, which uses realistic traffic scenarios and flits.
3.2.1 Network Wiring Cost

Ogras and Marculescu [121] showed that inserting long-range links on top of standard mesh network improves the performance of NoC. These long-range connections can dramatically reduce the average number of hops between any two nodes. Moreover, using high bandwidth communication links can lead to high throughput and low network latency. However, these interconnect fabrics are expensive because of the additional links. There are trade-offs between the network performance and network cost, therefore, total cost of the network is an important metric that needs to be considered to evaluate the network and find optimal ways to use additional long-range links.

We define the network 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.

$$WiringCost_{network} = \sum_{i,j \in G} Cost_{wire(i,j)}$$  \hspace{1cm} (3.1)

The cost of each wire is defined as a function of the pre-defined cost of a link type multiplied by the actual length plus a constant $c$,

$$Cost_{wire} = (a \times WireLength^b) + c,$$  \hspace{1cm} (3.2)

where $a$, $b$, and $c$ are user-defined parameters. For example, the network wiring cost for the $4 \times 4$-node network shown in Figure 3.1(b) is 15, where the pre-defined cost of link type is 1 ($a = 1$) and the technology cost is 0 ($c = 0$). Increasing the network size or adding additional links will increase the total number of links in
the network, and hence the wiring cost. Note that we have only used an additional fixed cost factor in Section 4.6.3.

3.2.2 Network Average Shortest Path

As the number of cores increases, the number of hops of a packet traveling from a source to a destination in the network also increases. Finding the shortest path between source and destination node is an important metric to measure to improve the network performance. This eventually reduces the maximum number of hops that a packet needs to travel on the network and increases the network throughput and reduces the network energy.

The Average Shortest Path (ASP) [116] is defined as the average number of hops in the shortest path for all possible \( i, j \) pairs of the network nodes with \( i \neq j \).

\[
ASP_{network} = \sum_{i,j \in G} \frac{distance(i,j)}{N(N-1)}, \tag{3.3}
\]

where \( N \) is the number of nodes. The distance between every pair of nodes in the network is calculated by using Dijkstra’s algorithm [53]. 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 \( 4 \times 4 \)-node network shown in Figure 3.1(b) is 3.87. When we add additional long-range links on the mesh network, the ASP will drop quickly while the clustering coefficient (i.e., how well a node is connected with its neighbor nodes) stays high. This is commonly called the small-world property of a network [159]. We will measure this property in Section 5.3.
3.2.3 Network Throughput

Throughput is one of the most important performance metrics to evaluate the performance of the network. This metric measures the total amount of data that can be sent across an interconnect fabric within a given time. 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 [127].

The network throughput is defined as the total number of packets arrived at their destination per node per cycle \( T \) [127]:

\[
TP_{\text{network}} = \frac{\text{Total packets arrived}}{(N \times \text{Total time})}
\]  

(3.4)

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

3.2.4 Network Energy Dissipation

Energy is also an important metric that needs to be measured to evaluate the overall performance of the network. When a data packet travels on the network, the energy is consumed by the routers and interconnect fabrics [1]. As the number of cores increases, the total number of hops needed for a packet to travel to its destination increases. As a result, it will increase the total energy dissipation in the network. Therefore, minimizing the energy consumption in the network is one of the NoC design goals, especially for embedded architectures.
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 $E_{\text{packet}}(i,j)$ [127]. We use the realistic energy estimates per packet as shown in Table 4.1.

$$E_{\text{network}} = \frac{\sum_{i,j \in G} E_{\text{packet}}(i,j)}{\text{duration}}$$ \hspace{1cm} (3.5)

### 3.3 Network Traffic Models

In a network, data (information) moves around from one node to another node until it reaches its destination. This data is split and formed into packets and the packets are then further split into flits (flow control digits). In our framework, packet traffic is simulated in a cycle-accurate manner. We consider a packet as an abstract unit that is communicated on the network. We do not simulate flits and do not use sophisticated flow control because they are not essential to obtain our results. However, we validate our results in Section 4.7 by using the GEM5 framework, which uses flits. As a matter of fact, our findings are independent of these details.

When a network is generated, we first check for network connectivity to see whether the links and nodes in a network correspond to a connected graph. If a network is not connected, there is no way packets can be routed well, on some parts of the network. Therefore, a network must be connected to run a traffic simulation.

In this dissertation, 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.

A routing table of size $N$, where $N$ is the number of network nodes, is generated for every node. It stores the next hop information for the different destinations. All packets will be sent to the next node by referring the routing table of the current node. FIFO buffers that store packets upon their arrival are built into the nodes with a control signal that tells whether the buffer is full or has empty spaces. If the next node buffer is full, a packet will stay in the current node.

For the network traffic simulations, we use three 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 [167] FFT, RADIX, and LU benchmarks for realistic application-based traffic patterns.

### 3.3.1 Uniform Random Traffic

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

### 3.3.2 Hot-Spot Traffic

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 $8 \times 8$-node networks (see Figure 3.2). For networks of size $10 \times 10$ and $12 \times 12$-nodes, node 11 and 88 and node 13 and 130 are used as hot-spot nodes respectively. The hot-spot
probability is \( p = 0.25 \).

Figure 3.2: 8 × 8-node mesh with a one-unit spacing between nodes. Hot spot nodes are marked with a \( \times \).

3.3.3 Transpose Traffic

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 (see Figure 3.2), we used the following node pairs: \((19, 26)\), \((13, 41)\), \((57, 15)\), and \((52, 38)\).

3.3.4 SPLASH-2 Benchmarks

SPLASH-2 [165] is the most commonly used multi-threaded benchmark suite for parallel machines with shared memory in both academia and industry. SPLASH-2 contains kernels and a variety of applications, such as high-performance computing and graphics that run on a shared memory multi-processors. In this dissertation,
we use FFT [11], RADIX [27,76], and LU [166] kernels from the SPLASH-2 benchmark suite as real applications to measure the performance metrics and evaluate the obtained evolved heterogeneous networks. The Fast Fourier Transforms (FFT) kernel is a complex one-dimensional algorithm that computes the discrete Fourier transform, RADIX is an integer sorting algorithm, and LU factorizes a dense matrix into a lower and upper diagonal matrix.

3.4 Network Simulation and Representation

In our C++ Heterogeneous Complex Network (HCN) framework, there are several individual components (see Figure 3.3), such as nodes, links, link types, and packets that are needed to be modeled network to run traffic simulations. This framework is based on the complex network framework [69].

Figure 3.3: Class structure of the Heterogeneous Complex Network (HCN) framework.
3.4.1 Network Abstraction

A basic network is composed of nodes and links. To build a heterogeneous network with different link types, we need another component called link type. The links in the network are modeled as an array of pointers to link type objects, and each link type has pointers to the link objects of that type. The nodes in the network are modeled as an array of pointers to node objects. Major functions of this module are node, link type, and link data which is read in from a network configuration file. When the node data is read from the file, the array of node objects are dynamically created. The same is done for reading in the link type and link data, creating the link type and link objects respectively.

3.4.2 Node Abstraction

The node abstraction encapsulates a node object in the network. In the HCN framework, nodes (IP blocks) are implemented as an array of pointers to node objects consisting of the physical \((x,y)\) coordinates of the nodes on the 2D grid. For each node, there is a routing table and a packet queue. A routing table stores the best route path information such as the next hop node IDs and the interconnect fabrics (links) and its type to route a packet to reach its final destination. A packet queue is a FIFO data structure that queues packets that arrive at the node.

The node object is used in both traffic and routing, therefore, there are two categories of functions, routing functions and traffic functions. The routing functions are used to set the routing table and get the next routing step for a destination node. The traffic functions are used to initialize the traffic for this node, add a packet to the queue, and remove a packet from the queue.
3.4.3 Link Abstraction

The link abstraction encapsulates a link object used for functions specific to links. A link object contains the nodes vector, which is an array of node numbers for the link. The key functions that operate on links are to get the link, set the values for the link, and set the number of node for a link.

3.4.4 Link Type Abstraction

The link type object stores all relevant link properties, such as cost, length, throughput, and energy consumption. Each link type has different values of these properties that distinguish one link type from another link type. A link type object consists of the name of the link type, the minimum and maximum length of the links, and a data structure that holds three parameters ($a$, $b$, and $c$) for computing the cost, delay, and energy of the link type. Based on these parameters, a power law formula as shown in Equation 3.6 is used to compute these metrics. There are three instances of this data structure, one for cost, one for delay, and one for energy. The cost is measured in terms of length, the delay is measured in $1/bandwidth$ of the link, and the energy is measured per bit.

$$f(m) = a \times m^b + c,$$ \hfill (3.6)

where $m$ is one of the metrics of the link type.
3.4.5 Packet Abstraction

This is the lowest level of abstraction of the network. A packet is a simply structure message unit that has a source, destination, and current location. When a packet moves from one node to another in the network, the functions on the packet object track the number of hops it has traveled and the number of iterations it has moved in the network for packet tracking.

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) [51, 56] 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 is then 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 genetic operators to the parent individuals. This process is repeated until the best solution is found. A pseudo algorithm of the basic parallel evolutionary optimization
**Algorithm 1**: Pseudo-code for a parallel evolutionary algorithm.

\begin{algorithm}
\begin{algorithmic}
\State $G \leftarrow \text{number of generation}$, $P \leftarrow \text{population size}$, $O \leftarrow \text{offspring}$, $g \leftarrow 0$
\State Randomly generate initial population $P$;
\While{g < total generation number $G$}
\State $g \leftarrow g + 1$;
\State Evaluate initial population $P(t)$;
\State Crossover:
\State Generate a random number $x$ between $(0,1)$;
\If{$x < P_c$}
\State Randomly select two parents;
\State Recombine to create a new offspring $O$;
\State $P'(t) \leftarrow P'(t) \cup O$;
\EndIf
\State Mutation:
\State Generate a random number $x$ between $(0,1)$;
\If{$x < P_m$}
\State Mutate to create a new offspring $O$;
\State $P'(t) \leftarrow P'(t) \cup O$;
\EndIf
\State Evaluate $P'(t)$
\State Selection:
\State $P(t + 1) \leftarrow select(P(t) \cup P'(t))$;
\EndWhile
\end{algorithmic}
\end{algorithm}

The procedure is described in Algorithm 1, which is inspired by [64].

The strength of EAs is that they perform well with problems that have multiple local optima [174]. EAs are typically used to find best solutions for given problems that cannot easily be solved by using other optimization techniques. They were successfully used in a variety of optimization problems, such as scheduling, routing, transportation problems, and engineering design [100,106]. 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 may look somewhat unstructured to the human eye, and it can thus be hard
to understand and analyze them. In Section 5 and also in a recent paper [35], 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 [30], which is a C++ white-box object-oriented framework dedicated to the reusable design of metaheuristics. The design flow of Heterogeneous Complex Network Evolutionary Algorithm (HCNEA) framework is shown in Figure 3.4. Details on implementation of the framework is described in following sections.

### 3.5.1 Initialization

The evolutionary algorithm starts by generating an initial population. Each member in the population has a genotype and a phenotype. The genotype is often a string of Booleans, but it can actually take the form of any data structure required. The phenotype is a translation of the genotype according to a set of translation rules. The main difference between the phenotype and the genotype is that the genotype is the data structure that is combined with others using crossover and also mutated according to the mutation function and the phenotype represents the solution to the given problem [61]. In our framework, we do not use a genotypical representation for our evolutionary algorithm, instead, all genetic operators work directly on the network level. The individuals of the initial population are randomly generated by interconnecting nodes randomly with different length and types of links.
3.5.2 Fitness Function

When calculating how well an individual solves a target problem, a genetic algorithm uses a fitness. For most of our problems, we consider multiple network performance factors. To consider two 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 a following:

$$f(w) = w \times WiringCost_{\text{network}} + (1 - w) \times ASP_{\text{network}}$$ (3.7)
Here, $w$ is the weight factor that allows us to determine the importance of either of the two factors, and $\text{WiringCost}$ and the $\text{ASP}$ 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 $\text{WiringCost}$ and $\text{ASP}$ are equally important. The aggregate objective function can readily be extended to include additional factors a designer may want to consider, such as throughput or energy as a following:

$$f(w) = w_1 \times \text{WiringCost}_{\text{network}} + w_2 \times \text{ASP}_{\text{network}}$$

$$- w_3 \times \text{Throughput}_{\text{network}} + w_4 \times \text{Energy}_{\text{network}}$$

(3.8)

where $w_1$, $w_2$, $w_3$, and $w_4$ are the weights for $\text{WiringCost}$, $\text{ASP}$, $\text{Throughput}$, and $\text{Energy}$ factors respectively. Each factor gives different value ranges, therefore, in order to use the weight factors precisely, we normalize the values to map the minimum and maximum value to the range $[0, 1]$. To normalize each factor, we run a minimum of 10 and a maximum 100 evolutionary runs to get the maximum value and divide the factor value by the maximum value.

3.5.3 Crossover

To generate new individuals for the next generation, the crossover genetic operator is first used. We perform crossover in the following way: two individuals are randomly selected from the current population. The selected individuals $p_1$ and $p_2$ have 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 ($pt_1$, $pt_2$) and then perform standard two-point crossover and generates two new individuals as
shown in Figure 3.5. We do this for each link type to generate new individuals. Unless otherwise specified, the crossover rate is 0.6.

![Two-point crossover operator](image)

**Figure 3.5: Two-point crossover operator.**

### 3.5.4 Mutation

Another very common genetic operator is mutation. Most of the time mutation simply changes one or more values of a gene to another valid value. In our framework, we use three mutation operators. 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 a different location in the network. The third mutation selects a link type and change the type of link with equal length. Unless otherwise specified, the mutation rate is 0.4.
3.5.5 Selection

Selection is the next step after all the individuals have had their fitness calculated. In this step a subset of the population is chosen, based on their fitness and passes on their individuals to the next generation after undergoing genetic operators, crossover and mutation. Selection can be done in many ways. A common method is called tournament selection [63, 108]. This is where fitness scores of a number of individuals are compared. The best two individuals are chosen for selection and they reproduce enough children to replace themselves and all the individuals that lost the tournament. In our framework, we use deterministic tournament selection [55] to select new individuals for the next generation.

3.5.6 Population Size and Evolutionary Runs

The number of generations and the population size are genetic parameters which are directly related to the simulation runtime. A large population size provides more diverse candidate solutions, therefore, the algorithm finds the optimal solution sooner than smaller population sizes [158, 164]. However, the simulation runtime is increased to simulate a large number of individuals for each generation. To determine the optimum population size and the number of generation, we ran experiments with different population sizes as shown in Figure 3.6. Figure 3.6 shows the fitness as a function of the generations for each population size, i.e., 80, 200, 400, and 600. The data was averaged over 50 different evolutionary runs. As one can see, the population size of 600 converged to the minimum fitness around 6000 generations while a population size 400 takes 3000 more generations to converge. Unless otherwise stated, for faster convergence with reasonable simulation run time, we use a population of 600 individuals evolved over 10,000 generations.
For all experiments, we did 10 evolutionary runs with 10 different initial populations and averaged the results. These parameters were experimentally determined.

Figure 3.6: Comparison of different population sizing and convergence. We observe that the population size of 600 converged to the minimum fitness around 6,000 generations while a population size of 400 converged around 9,000 generations.

3.5.7 Example

As an illustrative example, we optimized networks by changing the weight $w$ in Equation 3.7 to see what kind of NoC topologies we would obtain. We used the EA as described above with only one link type. Figure 3.7 shows the results for a $4 \times 4$-node evolved network with different weights $w = 0$, $w = 0.5$, 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 fewer hops. However, when we only consider wiring cost ($w = 1$), the network becomes a sparsely connected tree and only uses local connections to keep the
network cost as low as possible. For $w = 0.5$, a hub in node 9 evolves. The hub allows to keep the average path length low while minimizing the network cost. Note that we often get many networks that have the same cost and performance, yet have a different network topology. For example, when we optimize for cost, there are many tree-like networks that offer the same performance at the same cost while having different topologies. One reason for this are the many symmetries. The network we present here are therefore simply one instance out of many “optimal” networks.
Figure 3.7: $4 \times 4$-node evolved networks. (a) $w = 0$: performance is favored. The result is an almost fully connected network. $WiringCost = 100$, $ASP = 1.6$. (b) $w = 0.5$: wiring cost and performance are equally favored. A hub in node 9 evolves. $WiringCost = 25.4$, $ASP = 1.9$. (c) $w = 1$: wiring cost is favored. The resulting network is a sparsely connected tree. $WiringCost = 15$, $ASP = 3.9$. Only one link type was used for this example.
Performance Evaluation of NoC Architecture with Three Different Technology-driven Heterogeneous Link Types

In this chapter, we present the performance evaluation experiments in detail by using a technology-agnostic complex network approach that targets NoC architectures with various emerging communication link types. The goal is to show that networks with three different technology-driven heterogeneous link types can be beneficial in terms of cost, performance, and energy compared to homogeneous architectures and a regular 2D mesh. This will be shown by a series of experiments with increasing complexity.

4.1 Link Type Definitions

To explore heterogeneous complex networks with different types of abstract communication links on the same network, we defined three types of links as shown in Table 4.1. As one can see, each link is defined by a different value for the maximum wire length, the wiring cost, the energy consumption, and the throughput. We created these link characteristics by extrapolating the performance of each link type with respect to metal wires. The values also follow current and predicted data regarding the various interconnect technologies. As shown in [48], the area overhead introduced by the photonic links in a NoC is higher than for wireless links. Thus, we have chosen a higher cost for link type 3 compared to link type 2. Similarly, the achievable bandwidth using photonic links is higher than wireless for links. Hence, the throughput of link type 3 was chosen to be higher. Further, the
Table 4.1: Definition of the three different types of abstract links

<table>
<thead>
<tr>
<th></th>
<th>Link type 1</th>
<th>Link type 2</th>
<th>Link type 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum length</td>
<td>1 unit</td>
<td>3 units</td>
<td>20 units</td>
</tr>
<tr>
<td>Variable cost</td>
<td>1</td>
<td>2× actual length</td>
<td>4× actual length</td>
</tr>
<tr>
<td>Energy</td>
<td>1 unit/packet</td>
<td>0.1 unit/packet</td>
<td>0.05 unit/packet</td>
</tr>
<tr>
<td>Throughput</td>
<td>1 packet/clock</td>
<td>2 packet/clock</td>
<td>15 packet/clock</td>
</tr>
<tr>
<td>Candidate technology</td>
<td>metal wire</td>
<td>wireless</td>
<td>photonic</td>
</tr>
</tbody>
</table>

energy dissipation of the photonic links is the lowest of all link types. Lastly, the maximum wire length is motivated by the maximum distance a packet can travel in one clock cycle.

While these links are abstract on purpose, they can nevertheless rather straightforwardly be mapped to actual physical interconnects realized with current technology. Link type 1 corresponds to traditional metal wires, link type 2 to THz wireless links realized by means of carbon nano tubes (CNT) [152], and link type 3 to photonic links [90]. As mentioned above, the energy and throughput values correspond to realistic estimates.

By using the abstract links as defined in Table 4.1, the research goal then becomes to find an optimal heterogeneous network that has a lower cost, a higher throughput, and a lower energy consumption compared to a homogeneous link type network.

For most of our experiments, we limit the number of each link type. The reason for that is twofold: (1) The network cost would explode if the optimization algorithm is able to place unlimited numbers of links when cost is not or only weakly considered; (2) Technological limitations. Link type limitations are specified in each experiment.
4.2 Optimal Number of Links

The goal of the first experiment was to determine the optimal number of each of the different link types 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 also limited the number of each link type to 112, the number of links that would be required for a complete local 2D mesh of size 8 × 8 nodes. As a baseline for comparison, we also used an optimal network that was evolved without traffic.

We will first present the results individually before comparing them in Section 4.2.5.

4.2.1 Networks Without Traffic

First, we evolve networks without inserting any traffic to establish a baseline. For this experiment, we only considered WiringCost and ASP to see the distribution of the three different link types in the network by changing the importance of WiringCost versus ASP. The aggregate objective function $f(w) = w \times \text{WiringCost} + (1 - w) \times \text{ASP}$ was used in this experiment.

The resulting optimal networks are shown in Figure 4.2 while Figure 4.1 shows the distribution of the number of links as a function of the weight $w$. As one can see, when performance is favored ($w = 0$), all three link types are used in the network. In that case, a highly connected network with a hub (node 37) evolves. The hub serves to lower the average shortest path (ASP), so that between almost any pair of nodes, there is only one hop at most. The hub is only beneficial in a network without traffic because it would lead to significant congestion otherwise.

When we give more weight to WiringCost (i.e., higher $w$), the proportion of
link type 1 increases while the proportion of link type 2 and link type 3 decreases. When wiring cost is the only concern, we can see that only short-range links with lower cost links are used (see Figure 4.2 (c)). The network then basically becomes a sparse tree. For an equal importance of the two factors, one can see that two big hubs evolve in node 18 and 53. A smaller hub is located at node 21. The hubs help to lower the average shortest path while keeping the wiring cost low. The network in Figure 4.2 (a) is about 16 times more expensive than in (c), but the ASP is only less than twice as low. Note that compared to Figure 3.7 (b), two hubs evolved in 4.2 (b) due to the bigger network size.

![Figure 4.1: Heterogeneous link type distribution as a function of the weight \( w \). Depending on the importance of performance versus cost, a different number of each link type is used. A tree evolves when cost is favored \((w = 1)\), while a highly connected network with, wiring one link type only, significant hub (node 37) evolves if performance is favored \((w = 0)\).]
Figure 4.2: 8×8-node evolved networks without traffic. (a) \( w = 0.05 \): \( ASP \) only is important. \( WiringCost = 1002.8, ASP = 1.9 \). (b) \( w = 0.5 \): wiring cost and \( ASP \) are equally important. \( WiringCost = 248.6, ASP = 2.8 \). (c) \( w = 1.0 \): wiring cost only is important. \( WiringCost = 63, ASP = 3.6 \). Black links: type 1; green links: type 2; red links: type 3.
4.2.2 Networks with Uniform Random Traffic

Next, we added random uniform traffic to study how the heterogeneous link type distribution changes compared to the no-traffic scenario. For that purpose, we randomly generated packets with source and destination nodes selected with a uniform probability. The injection rate for this experiment was $iR = 0.6$ as used in [127], which is where the network starts to saturate.

Figure 4.3 shows the resulting link type distribution for networks optimized for $WiringCost$ and throughput $TP$ with the aggregate function $f(w) = w \times wireCost + (1-w) \times TP$. Figure 4.4 shows the corresponding networks. Compared to Figure 4.1, link type 1 is not used in most of the networks, except for the two networks where cost matters the most. This can be explained by link type 2 and 3 having a much higher throughput than link type 1. Note the absence of hubs, which would lead to congested nodes. As opposed to a network with hubs, a tree-like network based solely on link type 1 can be seen in Figure 4.4 (c) because that is the cheapest possible way to build a network with the link resources we provide.

We also evaluated the energy $E$ combined with $WiringCost$. As one can see from Figure 4.5 and Figure 4.6, the results are similar to the throughput experiments. The network is constructed with cheaper links that allow to lower the network cost and the expensive links that are used to reduce the network energy.
Figure 4.3: Heterogeneous link type distribution as a function of the weight \( w \). The networks are optimized for \( \text{WiringCost} \) and \( TP \) with uniform random traffic. Injection rate \( iR = 0.6 \). High throughput links of link type 2 and 3 are used when we give more weight to \( TP \) to absorb the network traffic. When we increase the weight to \( \text{WiringCost} \), the networks use a smaller number of link type 3 to reduce cost. When \( \text{WiringCost} \) only is considered, link type 1 only is used in the network.
Figure 4.4: 8 × 8-node evolved networks with uniform random traffic. (a) $w = 0.05$: TP only is important. High throughput links of link type 2 and 3 are used to improve network performance. $WiringCost = 342.8$, $TP = 0.59$. (b) $w = 0.5$: wiring cost and TP are equally important. Link type 2 is used to keep the throughput high while reducing the network cost. $WiringCost = 249.2$, $TP = 0.58$. (c) $w = 1.0$: wiring cost only is important, therefore, link type 1 only is used in the network. $WiringCost = 63$, $TP = 0.02$. Black links: type 1; green links: type 2; red links: type 3.
Figure 4.5: Heterogeneous link type distribution as a function of the weight $w$. The networks are optimized for $\text{WiringCost}$ and $E$ with uniform random traffic. Injection rate $iR = 0.6$. Energy-efficient link of link type 2 and 3 are used to minimize network energy consumption when we give more weight to $E$. When we increase weight to $\text{WiringCost}$, the networks use less expensive link of link type 1 and 2 to balance $\text{WiringCost}$ and $E$. 
Figure 4.6: 8×8-node evolved networks with uniform random traffic. (a) \(w = 0.05\): \(E\) only is important. \(WiringCost = 385.3, E = 0.4\). Long-range link of link type 3 are used to reduce the number of hops to lower the network energy dissipation. (b) \(w = 0.5\): wiring cost and \(E\) are equally important. Link type 2 is used to keep the throughput high while reducing the network cost. \(WiringCost = 154.2, E = 0.9\). (c) \(w = 1.0\): wiring cost only is important, therefore, link type 1 is only used in the network. \(WiringCost = 63, E = 18.1\). Black links: type 1; green links: type 2; red links: type 3.
4.2.3 Networks with Hot-spot Traffic

In this experiment, we used hot-spot traffic [121] as a more realistic traffic pattern. The two hot spots are node 9 and 54 located far apart from each other. The hot spot probability is $p = 0.25$, i.e., 25% of the packets will be sent to the hot-spots.

The results of optimizing the networks for WiringCost and TP are shown in Figures 4.7 and 4.8. The distribution plot shows that almost all optimal networks use three types of links with that traffic scenario. As we will see later in Chapter 5, local links of type 1 are used around the hot-spots to absorb the traffic and long-range link of type 2 and 3 are used to connect the clustered subnets.

![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 $iR = 0.6$. Link type 1, 2, and 3 are all used compared to the evolved networks under uniform random traffic to distribute the network traffic.]

The results for optimizing WiringCost and E with hot-spot traffic are shown in Figures 4.9 and 4.10. As one can see, long-range links of type 2 and 3 are used
more frequently for the majority of the weight values $w$ compared to the optimal network with uniform random traffic.
Figure 4.8: 8 × 8-node evolved networks with hot-spot traffic. (a) $w = 0.05$: TP only is important. $WiringCost = 863.1$, $TP = 0.59$. (b) $w = 0.5$: wiring cost and $TP$ are equally important. $WiringCost = 598.3$, $TP = 0.58$. (c) $w = 1.0$: wiring cost only is important. $WiringCost = 63$, $TP = 0.02$. Black links: type 1; green links: type 2; red links: type 3. Hot spot nodes are marked with a $\times$. High throughput long-range link type 3 is used to communicate between distant nodes located nearby the two hot spots to distribute the network traffic.
Figure 4.9: 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 $iR = 0.6$. Energy-efficient link type 3 is used more frequently for the majority of the weight value $w$ compared to the evolved networks under uniform random traffic to lower energy consumption.
Figure 4.10: $8 \times 8$-node evolved networks with hot-spot traffic. (a) $w = 0.05$: $E$ only is important. $WiringCost = 902.7$, $E = 0.2$. (b) $w = 0.5$: wiring cost and $E$ are equally important. $WiringCost = 189.3$, $E = 0.4$. (c) $w = 1.0$: wiring cost only is important. $WiringCost = 63$, $E = 3.7$. Black links: type 1; green links: type 2; red links: type 3. Hot spot nodes are marked with a ×. Long-range energy-efficient link type 3 is used more frequently for the majority of the weight value $w$ to lower $ASP$ to reduce energy dissipation.
4.2.4 Networks with Transpose Traffic

In transpose traffic [121], 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 \(8 \times 8\)-node networks.

Figures 4.11 and 4.12 show the results 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, long-range link of link type 2 and 3 are used more frequently to absorb the traffic for the majority of the weight values. Especially, when we consider \(TP\) to be more important, the optimal network uses more link type 3 compared to the optimal network with hot-spot traffic.

Figures 4.13 and 4.14 show the results for networks optimized for WiringCost and \(E\). When energy is considered to be more important, the optimal networks use more long-range because these links are more energy-efficient.
Figure 4.11: Heterogeneous link type distribution as a function of the weight $w$. The networks are optimized for $WiringCost$ and $TP$ with transpose traffic. Injection rate $iR = 0.6$. High throughput long-range link type 3 is more frequently used in the network compared to the evolved networks under hot-spot traffic to absorb the network traffic.
Figure 4.12: $8 \times 8$-node evolved networks with transpose traffic. (a) $w = 0.05$: $TP$ only is important. $WiringCost = 955.4$, $TP = 0.59$. (b) $w = 0.5$: wiring cost and throughput are equally important. $WiringCost = 613.4$, $TP = 0.58$. (c) $w = 1.0$: wiring cost only is important. $WiringCost = 63$, $TP = 0.01$. Black links: type 1; green links: type 2; red links: type 3. To support high throughput, a large number of high throughput link type 3 is used in the network.
Figure 4.13: Heterogeneous link type distribution as a function of the weight $w$. The networks are optimized for $WiringCost$ and $E$ with transpose traffic. Injection rate $iR = 0.6$. Energy-efficient evolved networks are generated by using a large number of link type 2 compared to the evolved networks under hot-spot traffic.
Figure 4.14: 8 × 8-node evolved networks with transpose traffic. (a) $w = 0.05$: $E$ only is important. $\text{WiringCost} = 576.3$, $E = 0.3$. (b) $w = 0.5$: wiring cost and $E$ are equally important. $\text{WiringCost} = 185$, $E = 0.4$. (c) $w = 1.0$: wiring cost only is important. $\text{WiringCost} = 63$, $E = 3.8$. Black links: type 1; green links: type 2; red links: type 3. A large number of link types 2 is used to reduce the network energy consumption compared to the evolved network under hot-spot traffic.
4.2.5 Comparison and Discussion

We will briefly discuss and compare the results from Sections 4.2.1 to 4.2.4.

Figure 4.15 shows the total number of links (i.e., the sum of link type 1, 2, and 3) used in the networks from the previous experiments. Note that we had limited the number of each link type to 112, the number of links that would be required for a complete local 2D mesh of size $8 \times 8$ nodes. As one can see in Figure 4.15 (a), the evolved networks with hot-spot and transpose traffic use more than twice as many links compared to uniform random traffic to absorb the traffic. Hot-spot and transpose traffic show very similar results otherwise. When optimized for cost and energy, the results are somewhat different, as Figure 4.15 (b) shows. The total number of links converges to 63 links for weights $w > 0.5$, i.e., when energy becomes more important. Also, considering energy as an optimization factor results in using about half the number of links compared to networks where $TP$ is considered besides the cost.

Figure 4.16 shows the usage of each link type for three different traffic patterns. Figure 4.16 (a) shows the networks optimized for cost and throughput $TP$ and Figure 4.16 (b) for cost and energy $E$. We can see that overall, link type 3 is used the least because it is expensive, yet offers great performance. The more $TP$ and $E$ are important, the more links of type 2 and 3 are used. When we only optimize for cost ($w = 1$), only link type 1 is used because it is very cheap.
Figure 4.15: 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 $E$. When we give 90% weight to network performance, (1) the evolved networks optimized for $WiringCost$ and $TP$ under hot-spot and transpose traffic use 100% and 83% more links respectively compared to the networks under uniform random traffic to absorb the network traffic, (2) the evolved networks optimized for $WiringCost$ and $E$ under hot-spot and transpose traffic use 14% and 29% more links respectively compared to the networks under uniform random traffic.
Figure 4.16: The use of each link type in the network under three different traffic patterns with injection rate $iR = 0.6$. (a) The network is optimized for $\text{WiringCost}$ and $TP$; (b) the network is optimized for $\text{WiringCost}$ and $E$. Hotspot traffic use a large number of link type 1 and 3 compared to the evolved networks under uniform random and transpose traffic to increase throughput at a low cost. When network performance is more considered, high throughput energy-efficient link types are used to support the network to improve the performance.
Discussion: With the traffic patterns and trade-offs under consideration so far, it is beneficial in all cases (except when cost only is important) to make use of heterogeneous links to increase the performance or lower the energy at a low cost. To improve the throughput, a large number of low-cost short-range links with a small number of long-range links are used to absorb the traffic at a low cost. Energy is minimized by using high performance long-range links to reduce the average shortest path. Note that we did not optimize networks for all three factors together, i.e., cost, energy, and throughput, because energy and throughput are essentially proportional.
4.3 Optimal Network for 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 link type 1, 24 for link type 2, and 62 for link type 3 respectively. The specific limits for link type 2 and 3 are motivated by technological considerations. For example, it is currently possible to only create 24 non-overlapping wireless links (link type 2) using CNT antennas [65]. Similarly, using Wavelength Division Multiplexing (WDM), 62 maximum wavelength channels can be created in case of photonic interconnects (link type 3) [134].

We consider 8×8-node networks with three different traffic patterns (see Section 3.3), namely uniform random, hot-spot, and transpose with a target injection rate of \(iR = 0.6\). First we find an optimal network by optimizing cost and throughput for each pattern by equally weighting their importance (\(w = 0.5\)). Figure 4.19 shows the resulting networks for these three different traffic scenarios. A first observation is that no hubs evolve, not even for the hot-spot traffic scenario.

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 Figure 4.17, the network throughput increases and peaks at the target injection rate of \(iR = 0.6\). Naturally, the throughput peaks at \(iR = 0.6\) because the networks were optimized specifically for that target throughput. Because cost is a factor in the optimization, only as many links as needed will be added to the network. Thus, for any traffic that is beyond the target injection rate, the network will not be optimal. Second, the networks seem to lack structure. However, as we will see in Chapter 5, the underlying structure is not easily visible to the human
Figure 4.17: Network throughput of evolved topology and mesh network 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.

In addition, we also compared the network throughput $TP$ of our evolved topologies with a $8 \times 8$-node mesh under the same traffic patterns. Figure 4.17 shows that all of our evolved topologies provide 2.8 times higher throughput than the mesh networks. This shows once again that networks with heterogeneous link types can achieve better performance than regular 2D mesh networks. It is not surprising that adding long(er)-range connections to a mesh network helps to improve the performance. That was successfully shown by Ogras and Marculescu [121], but only for homogeneous networks. Figure 4.18 shows one of their obtained network. Long-range links are inserted at locations where it is most beneficial to improve the network performance. As one can see, the network looks similarly unstructured to
our evolved networks. In Section 4.5 we will show that our heterogeneous networks perform better than any other homogeneous single link type network.

Figure 4.18: 4 × 4-node mesh-based network with long-range links. Long-range links are inserted where it is most beneficial to improve the network performance. Redrawn from: [121].
Figure 4.19: 8 × 8-node optimal evolved networks under different traffic scenarios. WiringCost and TP is equally weighted (w = 0.5) using an aggregate objective function: $f = w \times \text{WiringCost} + (1 - w) \times TP$. Black links: type 1; green links: type 2; red links: type 3. Hot spot nodes are marked with a ×.
4.4 Optimal Network for Energy Dissipation

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

In this section we use the abstract heterogeneous link types defined in Section 4.1 to find optimal networks under $WiringCost$ and energy $E$ constrains with an equal importance, i.e., $w = 0.5$, $f(w) = w \times WiringCost + (1 - w) \times energy$. We one again limited the maximum number of links that can be used in the network for each link type to 112 for link type 1, 24 for link type 2, and 62 for link type 3 respectively. The same traffic patterns as used before were used and the injection rate was $iR = 0.6$.

The resulting optimal networks with heterogeneous links are shown in Figure 4.21. We then took the optimal networks for each weight $w$ and subjected them to traffic with different injection rates. The results were compared with mesh networks. As one can see in Figure 4.20, the evolved topologies consume 2.8 times less energy comparing to the mesh 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.
Figure 4.20: Network energy of evolved and mesh networks with different traffic scenarios. Injection rate $iR = 0.6$. All evolved topologies with heterogeneous links use significantly less energy compared to a regular 2D mesh topology.

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 4.3, the evolutionary algorithm does a splendid job in obtaining networks that meet the target requirements while minimizing both energy and cost.
Figure 4.21: 8 × 8-node optimal evolved networks with different traffic scenarios. WireCost and E is equally weighted (w = 0.5) using the aggregate objective function: \( f = w \times \text{WiringCost} + (1 - w) \times E \). Black links: type 1; green links: type 2; red links: type 3. Hot spot nodes are marked with a ×.
4.5 Network Performance Comparison with Mesh and Homogeneous Networks

Next, we evaluated the performance metrics, such as throughput $TP$ and energy $E$ of networks with heterogeneous link type and compared them with regular 2D meshes and networks with homogeneous link types. We define a *homogeneous link type* network as a network that only uses link type 1, but without a length restriction in order to allow long-range links. This network is based on a regular 2D mesh, but contains additional long-range connections to improve the performance. The network therefore corresponds to the type of network as proposed by Ogras and Marculescu [121]. The only difference is that we consider cost while they did not. In Section 4.6.1, we also explore networks with one type of link only (link type 2 or 3) for long-range links on top of a regular mesh for a detailed comparison of the network performance.

Figure 4.22 shows the comparative results of heterogeneous link type networks with regular 2D meshes and homogeneous link type networks under uniform random traffic. As one can see, both the network throughput $TP$ and energy $E$ of our 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 a higher than 90% importance to cost.
Figure 4.22: 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 $w$. Our evolved topologies with heterogeneous link types under uniform random traffic perform significantly better for the majority of the weight value $w$ compared to regular 2D mesh networks. Moreover, when network performance and wiring cost are equally weighted ($w = 0.5$), our evolved network provides 14% more throughput and 17 times less energy consumption compared to a homogeneous link type network.
Figure 4.23: 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 $w$. Our evolved topologies with heterogeneous link types under hot-spot traffic perform significantly better for the majority of the weight value $w$ compared to regular 2D mesh networks. Moreover, when network performance and wiring cost are equally weighted ($w = 0.5$), our evolved network provides 21% more throughput and 13 times less energy consumption compared to a homogeneous link type network.
Figure 4.24: 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 $w$. Our evolved topologies with heterogeneous link types under transpose traffic perform significantly better for the majority of the weight value $w$ compared to regular 2D mesh networks. Moreover, when network performance and wiring cost are equally weighted ($w = 0.5$), our evolved network provides 10% more throughput and 13 times less energy consumption compared to a homogeneous link type network.
Figures 4.23 and 4.24 show the same performance comparison for hot-spot and transpose traffic. Again, one can see that our evolved networks with heterogeneous links result in a higher throughput and a lower energy consumption than 2D mesh networks or homogeneous link type networks. The results in this section indicate that having multiple types of links improves the network performance and lowers the energy at a low cost.
4.6 Model Variation

In this section, we explore more of experiments by changing several model assumptions to illustrate that our outcomes are robust against such variations.

4.6.1 Performance Comparison of Heterogeneous Link Type Networks with Homogeneous Link Types

In order to show that networks with heterogeneous link types provide higher throughput and lower energy consumption compared to networks with homogeneous link types, we have performed additional simulations by using one type of link (only link type 2 or link type 3) for long-range link on top of a regular mesh network. Figure 4.25 shows that networks with three different link types improve the network performance compared to networks with one link type only.

While a single technology may indeed be cheaper and easier to manufacture, our results clearly show that performance can be significantly improved with additional technologies. How much one is willing to pay is ultimately a design decision influenced by the application and the market. We believe that there are plenty of applications (e.g., defense applications, aerospace, bioinformatics, datacenters, etc.) where performance is a constant bottleneck, but cost is not a primary issue of concern. The other advantage of our approach is that we can determine the optimal network for essentially any point in the cost-performance space.
Figure 4.25: Performance comparison between the networks with three different link types vs. a single link type. (a) The networks are optimized for \textit{WireCost} and \textit{TP}; (b) the networks are optimized for \textit{WireCost} and \textit{E} under uniform random traffic by equally weighting their cost-performance weight ($w = 0.5$). Injection rate $iR = 0.6$. Evolved networks with three different technology-driven link types provide higher through and lower energy consumption compared to two different homogeneous networks.
4.6.2 Performance Evaluation with Different Traffic Scenarios

As explained in Section 3.5, our networks so far were evolved for specific traffic patterns. However, it is straightforward to design them for multiple traffic scenarios at the same time. For this experiment, we optimized networks for $WireCost$ vs. throughput ($TP$) and $WireCost$ vs. energy ($E$) under three different traffic scenarios individually and evaluated each network with other traffic. For example, Figure 4.26 is an evolved network under uniform random traffic, which is also evaluated with hot-spot and transpose traffic. As one can see, the results show that the optimal network with uniform random traffic provides higher throughput compared to the other traffic patterns. This is because the network was optimized specifically for this target traffic pattern. However, one can also notice that although the networks were optimized for uniform random traffic, they still perform better on other common traffic patterns, hot-spot and transpose.

The same result is observed in the evolved network under hot-spot and transpose traffic as shown in Figure 4.27 and 4.28. This illustrates that our evolved networks contain structure that is universally beneficial for various scenarios.
Figure 4.26: Performance comparison of networks evolved under random uniform traffic and tested on hot-spot and transpose traffic. (a) network throughput; (b) energy; \( w = 0.5 \). We conclude that our evolved topology is optimized for uniform random traffic, and at the same time outperforms on both hot-spot and transpose traffic patterns.
Figure 4.27: Performance comparison of networks evolved under hot-spot traffic and tested on uniform random and transpose traffic. (a) network throughput; (b) energy; \( w = 0.5 \). We observe that our evolved topology is optimized for hot-spot traffic, and at the same time outperform on both uniform random and transpose traffic patterns.
Figure 4.28: Performance comparison of networks evolved under transpose traffic and tested on uniform random and hot-spot traffic. (a) network throughput; (b) energy; $w = 0.5$. We observe that our evolved topology is optimized for transpose traffic, and at the same time outperform on both uniform random and hot-spot traffic patterns.
4.6.3 Performance Comparison with a Technology-driven 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 wiring cost formula is shown in Equation 3.2, Section 3.2. The fixed cost values $c$ which were used for each link type was set to 1 for link type 1, 2 for link type 2, and 4 for link type 3 respectively. For the sake of this experiment, we assumed that the technology becomes more expensive the more powerful the links are. This is because, for example, we can achieve high-bandwidth low-power consumption by using optical interconnect fabric for NoC design. However, optical NoCs are very expensive because of costly components, such as optical router, packaging, cooling, and system integration.

Figure 4.29 shows the results obtained by optimizing $WireCost$ and $TP$ under uniform random traffic. As one can see, the main results of this dissertation remain unchanged. In particular, the throughput and the link type distribution are almost identical.
Figure 4.29: Performance comparison of networks with and without technology cost. Networks are optimized for WireCost and TP under uniform random traffic by equally weighting their cost-performance weight \((w = 0.5)\). Injection rate \(iR = 0.6\). (a) Throughput comparison of the evolved topologies; (b) heterogeneous link type distribution comparison.
4.7 Performance Evaluation by Using GEM5

In order to validate the results of our abstract framework, we used the GEM5 platform [22], which is an open source modular-based architectural modeling tool that can simulate a complete system with operating systems and devices, CPUs, caches, buses, etc. We ran a full system simulation with the Ruby and Garnet fixed-pipeline network model on a 64-core Alpha architecture. For our results, we used the SPLASH-2 [167] Fast Fourier Transforms (FFT), RADIX, and LU kernels as real application benchmarks to measure the network performance, evaluate our obtained evolved network with three different heterogeneous link types, and compare the network performance with homogeneous networks and a regular 2D mesh.

Figure 4.30: Performance comparison obtained with gem5 between evolved and mesh topologies. Our evolved topology has a 33\%, 41\%, and 37\% lower latency compared to a regular 2D mesh network under FFT, Radix, and LU traffic respectively. The results validate our abstract framework.
We add our obtained heterogeneous topologies in GEM5 and run the SPALSH-2 benchmarks on our topologies. Figure 4.30 shows the network performance comparison between our evolved heterogeneous topology and a regular 2D mesh network. As one can see, the latency with the SPLASH-2 FFT, RADIX, and LU traffic loads are much lower for our evolved topology than for a regular mesh, which confirms the results obtained so far by our abstract framework.

In addition, we also compare the network performance with homogeneous networks, which use a single link type as a long-range link. As one can see in Figure 4.31, our evolved network with three different link types provide lower latency for all three SPLASH-2 applications compared to homogeneous networks. The network latency comparison between two different homogeneous networks shows that a network with link type 3 has lower latency compared to a network with link type 2. This is because link type 3 provides higher throughput over longer wire lengths than link type 2.

These results validate our abstract framework and shows that the obtained heterogeneous network architecture provides high network performance compared to homogeneous networks and a regular 2D mesh.
Figure 4.31: Performance comparison obtained with gem5 between evolved and homogeneous topologies. Under FFT, Radix, and LU traffic, our evolved topology with heterogeneous link types has on average a 31% and 11% lower latency compared to a homogeneous network with link type 1 and link type 2 respectively. The results validate our abstract framework.
4.8 Discussion

In this chapter we presented the benefits of using heterogeneous link types for NoC designs. To prove our hypothesis, we use three different technology-driven link types, which can directly be mapped on actual technology to design NoC architectures, namely metal wires, wireless, and optical links. These heterogeneous links are irregularly placed on the network by using evolutionary optimization techniques to provide the best performance on this network. To design high-performance low-cost topologies, we consider and balance both network cost and performance factors.

We presented a series of experiments to evaluate our heterogeneous topologies. Our results show that our optimal networks with heterogeneous link types provide higher throughput and lower energy dissipation, not only compared to regular 2D mesh networks, but also compared to homogeneous link type networks under uniform random, hot-spot, and transpose traffics. We use the GEM5 framework [22] to run realistic traffic scenarios on our evolved networks. The results also show that our networks with heterogeneous link types provide better performance compared to homogeneous and mesh networks. Our obtained networks is optimized for a specific traffic pattern, however, we showed that they still outperform with other common traffic patterns. Even with new assumptions, assigning a fixed cost for each technology used to test our abstract model against variations in the assumptions, our heterogeneous link type networks still outperform homogeneous and regular mesh networks. By using our framework, we have provided optimal number of link types, optimal wire-length distributions, and optimal link placement for obtained network to show the heterogeneous link type usage.

All the results we have shown in this chapter confirm that irregular networks
with heterogeneous link types improve the network performance in terms of throughput and energy. To the best of our knowledge, no one has explored irregular interconnect-based NoC architectures with three or more heterogeneous link types.
Network Analysis

Designing a large-scale heterogeneous NoC architecture that performs well is a non-trivial task because of the massive design space. Using optimization techniques is a common approach [48, 103] and does a good job for multi-objective problems. However, the resulting networks are often unstructured, making it hard to analyze, particularly as the system size and the number of network links increases. Discovering and investigating the underlying complex network structure is important to describe the network behavior and to deduce general design principles.

Since the beginning of complex network research, several interesting techniques, such as network community detection [28, 67, 124] and small-worldness [8, 139], have been proposed and used for the structural analysis of complex networks. One of the most important methods that is used to analyze the community structure of large-scale networks is community detection.

In this chapter, we analyze the previously obtained optimized evolved network by using the Brain Connectivity analysis toolbox [139]. We use this toolbox to detect community structure and measure the small-worldness. The main goal is to determine interesting network structure and properties, which we might not see by eye, and to find out how heterogeneous link types are used and placed in our evolved networks. We hypothesize that our evolved networks have inherent structure and use the heterogeneous link types optimally under the given constraints.
5.1 Network Community

A network consists of nodes and links which can be naturally partitioned into subgroups, called the communities [117], as shown in Figure 5.1. Each subgroup has a set of nodes connected densely internally and more sparsely between subgroups. Most of the networks evolved a shown in Section 4.2 seemingly very unstructured. To get a better insight into why the evolutionary algorithm comes up with specific networks, we use the *Brain Connectivity toolbox* [139], which is an open source software package. First we explored the evolved networks to find the network community structure by using the algorithm of Clauset *et al.* [38]. The algorithm detects communities that are strongly connected within each community and weakly connected between communities.

Figure 5.1: Network community structure. Three communities are detected in this network. There are many links between nodes within community while fewer links between communities. This shows that the nodes within community are more densely connected. Redrawn from [117].
The community structure is measured by the modularity ($Q$) [38,78] value:

$$Q = \frac{1}{2m} \left( \sum_{i,j \in C} \left( A_{ij} - \frac{k_i k_j}{2m} \right) \right) \delta_{ij}, \quad (5.1)$$

where $m$ is the total number of edges in the network,

$$m = \frac{1}{2} \sum_{i,j \in C} A_{ij} \quad (5.2)$$

$A_{ij}$ is the adjacency matrix, and $k_i$ and $k_j$ are the degrees of the nodes. $\delta_{ij}$ is one if node $i$ and node $j$ are in the same community and zero otherwise. If $Q$ increases, the connectivity within modular regions of a network also increases. In other words, a network has a strong community structure as $Q$ gets closer to 1.

Figure 5.2: Community structure of the evolved network under hot-spot traffic by using the aggregate objective function: $f = w \times WiringCost + (1 - w) \times TP$. $WiringCost$. Throughput $TP$ are equally important ($w = 0.5$). Black links: type 1; green links: type 2; red links: type 3. Hot-spot nodes are marked with a ×. The thick dash-dotted lines show the communities.
Figure 5.2 shows the network community structure of our evolved network from Figure 4.8 (b). This network is optimized for throughput $TP$ and $WiringCost$ with equal importance ($w = 0.5$) under hot-spot traffic. The network community in Figure 5.2 is maximized at $Q = 0.4793$ with a partition into four communities. All three link types defined in Table 4.1 are used to obtain an optimal high throughput network.

To understand how three different link types are placed to support high throughput, we analyze each link type usage in the network as shown in Figure 5.3 (a). As one can see, link type 3, which has a longer range and a higher throughput, is used more frequently for communication between communities while link type 1, which has a shorter range with a lower cost, is used to locally connect inside a community. Interestingly, link type 2 is used within communities as a short-range link while the longer-range version of the same type is used as intra-communities links as shown in Figure 5.3 (b). We obtain these wire length distribution that can be fitted by a Gaussian curve. From this result, we observe that large number of short-range links of link type 2 are placed nearby the two hot-spots to distribute the network traffic and provide high throughput.
Figure 5.3: Link type usage of evolved network optimized for WiringCost and throughput $TP$ with equal importance ($w = 0.5$) hot-spot traffic. (a) Connectivity between and within communities. (b) Wire-length distribution for each link type used between communities and within a community.
Figure 5.4 shows the total number of each link type used in the networks for different cost-weights $w$. As one can see, link type 1 is frequently used in overall cost-weights because of its low cost. This link type is generally used to communicate within community to absorb the network traffic. A large number of link type 2 is used when we give less weight to cost factor. We observe that long-range link of link type 2 is used to communicate between communities and short-range links are used to communicate within community (see Figure 5.3). A high-throughput long-range link of link type 3 are used to communicate between communities.
Figure 5.4: Link type distribution of the evolved network under hot-spot traffic by using the aggregate objective function: $f = w \times WiringCost + (1 - w) \times TP$. A large number of short-range links of link type 1 and 2 are used within community to absorb the network traffic. A small number of long-range link of link type 3 is used to communicate between communities.
Figure 5.5: Community structure of the evolved network under transpose traffic by using the aggregate objective function: \( f = w \times \text{WiringCost} + (1 - w) \times TP \). Throughput \( TP \) and \( \text{WiringCost} \) are equally important \((w = 0.5)\). Black links: type 1; green links: type 2; red links: type 3. The thick dash-dotted lines show the communities.

The network community structure of our evolved network under transpose traffic is shown in Figure 5.5. This network is optimized for throughput \( TP \) and \( \text{WiringCost} \) with equal importance \((w = 0.5)\) (see Figure 4.12 (b)). The community of the network is maximized at \( Q = 0.4615 \) with a partition of five communities. As one can see, the link usage for each link type is similar to the evolved network under hot-spot traffic. Expensive long-range link of link type 3 is frequently used to connect between the communities to provide high throughput while cheap short-range links of link type 1 are used more frequently to connect within a community. For link type 2, it is both frequently used within communities as a short-range link and between communities a the long-range link as shown in Figure 5.6. The wire length distribution of link type 2 and 3 becomes a Gaussian. When we look at the number of link type used in Figure 5.6 (a), we observe that
the traffic of evolved network under transpose traffic is absorbed by using a large number of low-cost link type 1 compared to hot-spot traffic (see Figure 5.3 (a)).
Figure 5.6: Link type usage of evolved network optimized for WiringCost and throughput TP with equal importance ($w = 0.5$) transpose traffic. (a) Connectivity between and within communities. (b) Wire-length distribution for each link type used between communities and within a community.
Figure 5.7: Link type distribution of the evolved network under transpose traffic by using the aggregate objective function: \( f = w \times WiringCost + (1 - w) \times TP \). A large number of short-range link of link type 1 and 2 are used within community to absorb the network traffic. A small number of long-range link of link type 2 and 3 are used to communicate between communities.
Figure 5.7 shows the total number of each link type used in the evolved networks under transpose traffic for different cost-weights $w$. When we compare the link type usage between hot-spot and transpose traffic, the evolved network under hot-spot traffic uses more number of link type 2 in the network. This is because the networks have a large amount of traffic in the hot-spot area. Therefore, high throughput short-range links of link type 2 are used to absorb the traffic and improve the network performance.

Discussion: From these results we learn that local communities are evolving and that the different link types are used as one would expect, even though the networks shown in Figure 5.2 and Figure 5.5 initially look very unstructured.

5.2 Network Modularity

Modularity measures the strength of a partition in a complex network through connectivity. While it is calculated in various ways, modular networks are generally classified by regions of dense connectivity, or modules that are sparsely linked together. For this reason, modularity is often used in methods for community detection. Clune et al. [39] claimed that as a greater emphasis is put on cost in the evolution of complex biological networks (i.e., $w > 0.5$ in our case), the networks develop into increasingly modular networks in order to maximize the level of performance. Using the Brain Connectivity toolbox [139] (see Section 5.1), which uses standard algorithms to output a fractional measure of modularity, we find that the modularity of our networks generated through evolutionary algorithms indeed also increases with the weight $w$ (see Figure 5.8). The two extreme cases, i.e., $w = 0$ and $w = 1$ are somewhat pathological cases in the sense that extreme networks evolve, i.e., a tree and a fully connected network. Overall, our findings
thus confirm Clune et al.’s results for biological networks, which is another direct indication that the networks we evolve have inherent structure and use the available resources under the given constraints optimally.

Figure 5.8: Network modularity of networks optimized for hot-spot and uniform random traffic. The results confirm Clune et al.’s [39] results on biological networks.

5.3 Small-Worldness

Next, we wanted to determine if the evolved topologies have small-world properties. A small-world network is defined as a network that has a high clustering coefficient but a small average shortest path length [159]. The network can be observed as a middle stage between a regular lattice and a random graph as shown in Figure 5.9. As probability $P$ increases, wires in a ring network (regular lattice) are removed and rewired with long-range wires to a random node. This dramatically reduces the average shortest path length without decreasing the clustering coefficient.
In order to measure the small-worldness of our evolved networks with heterogeneous links, we used the common formula provided in Rubinov and Sporns’ paper [139]:

$$S = \frac{(C/C_{rand})}{(L/L_{rand})} \quad (5.3)$$

Measuring the small-worldness ($S$) is based on the local clustering coefficient ($C$) and characteristic shortest path length ($L$). $C$ is a measure of the neighbor’s node connectivity [139]:

$$C = \frac{1}{N} \sum_i \frac{2t_i}{(d_i - 1)d_i}, \quad (5.4)$$

where $N$ is the number of nodes. $t_i$ is the number of triangles around a node $i$, and $d_i$ is the number of degree of a node $i$. $L$ is the average shortest path length.
Table 5.1: Small-worldness for evolved topologies. If $S$ is greater than 1, the network is said to be a small-world network.

<table>
<thead>
<tr>
<th>Topology</th>
<th>Uniform random</th>
<th>Hot-spot</th>
<th>Transpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Optimized for throughput and cost</td>
<td>$S = 0$</td>
<td>$S = 1.8894$</td>
<td>$S = 1.9196$</td>
</tr>
<tr>
<td>Optimized for energy and cost</td>
<td>$S = 0$</td>
<td>$S = 1.6540$</td>
<td>$S = 0$</td>
</tr>
</tbody>
</table>

($ASP$) between all source and destination pairs in the network [139]:

$$L = \frac{1}{N} \sum_i l_{ij}$$  \hspace{2cm} (5.5)

$l_{ij}$ is the shortest path length between node $i$ and $j$, and $N$ is the number of nodes.

For small-worldness to be high, a network needs to have a high connectivity and low characteristic path length. If $S$ is greater than 1, the network is said to be a small-world network. We use the Erdős-Rényi network for the random network in the formula [57].

The results for our network are shown in Table 5.1. As one can see, only the evolved networks optimized for throughput $TP$ and $WiringCost$ with both hot-spot and transpose traffic show small-worldness as defined by the formula above. As shown in Figure 4.15, the networks optimized for energy $E$ and $WiringCost$ use less wires, and are therefore much sparser, which results in a low clustering coefficient. That is main the reason why these networks do not have the small-world property.
5.4 Discussion

The evolved networks presented in this dissertation may all seem unstructured. To understand the underlying complex network structure, we use the Brain Connectivity analysis toolbox [139] to detect community structure and measure the small-worldness. We show that our evolved networks with heterogeneous link types have an underlaying structure that evolves as a function of the design constrains. Communities are detected in the networks and high-throughput long-range links are used to communicate between communities and low-cost short-range links are used to communicate within communities. Moreover, we observe that a large number of short-range links are used in the network to distribute the traffic and provide high throughput at a low cost instead of using a small number of expensive links.

In addition, most of the networks are indeed small-world. In [154], it was already shown that unstructured small-world interconnect networks can have major advantages over local 2D or 3D regular topologies, however, these networks were not evolved.
Performance Evaluation of NoC Architectures with Ten Different Heterogeneous Link Types

In Chapter 4, we showed that networks with three different technology-driven heterogeneous link types (i.e., metal wires, wireless, and optical) led to a higher throughput and a lower energy dissipation compared to both homogeneous link type networks and regular 2D mesh networks. However, these technology-driven link types may not represent an optimal link library. Thus, the research question we want to address is whether more link types are beneficial.

In order to show this, we (1) obtain optimized networks according to any number and combination of the common network performance metrics, such as the wiring cost, the average shortest path length, the throughput, and the energy and (2) evaluate the network under different realistic traffic models. We show that our proposed heterogeneous architectures outperform heterogeneous architectures with three different technology-driven link types, homogeneous architectures, and a regular 2D mesh in performance, energy, and throughput. The following section presents the performance evaluation experiments in detail. Unless otherwise stated, we used a population of 600 individuals (i.e., 600 different networks) evolved over 14,000 generations. These values were defined experimentally.

6.1 Link Type Definitions

To explore heterogeneous complex networks with a larger number of 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 Figure 6.1. Link type 1 and link type 10 can be thought of as representatives of metal wires and photonic links [48] respectively. Note that all these link types are abstract and technology-agnostic on purpose in order to obtain results that are more broadly applicable. As explained in the introduction, the goal of this section is to find out what level of heterogeneity is the best. We have chosen ten different link types for this experiment because the differences in each metric were still interpretable. Also, it is rather unlikely that a higher number link types will ever become available. More link types would have made the results significantly harder to analyze and interpret. In Section 6.7 we also explore heterogeneous link type networks with a non-linear cost mapping of the links to show that the main results remain unchanged.

![Figure 6.1: Definition of the different types of abstract links.](image)

The y-axis in Figure 6.1 shows the value of each metric. Each link is defined
Table 6.1: Absolute value of each metric for link type 1, 6, and 10.

<table>
<thead>
<tr>
<th></th>
<th>Link type 1</th>
<th>Link type 6</th>
<th>Link type 10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum length</td>
<td>1 unit</td>
<td>11.5 units</td>
<td>20 units</td>
</tr>
<tr>
<td>Variable cost</td>
<td>1</td>
<td>2.65× actual length</td>
<td>4× actual length</td>
</tr>
<tr>
<td>Energy</td>
<td>1 unit/packet</td>
<td>0.48 unit/packet</td>
<td>0.05 unit/packet</td>
</tr>
<tr>
<td>Throughput</td>
<td>1 packet/clock</td>
<td>9 packet/clock</td>
<td>15 packet/clock</td>
</tr>
</tbody>
</table>

by a different value for the maximum wire length, the wiring cost, the energy consumption, and the throughput. The wiring cost is defined by using Equation 3.2. 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 6.7.2. Table 6.1 shows the absolute values of each metric for link type 1, 6, and 10. The rest of the link type values can be seen in Figure 6.1. Note that we do not make any statement whether our ten link types can be implemented in some future technology. We are solely interested in—as stated in the introduction—finding 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 low cost, high throughput, and low energy consumption.

For most of our experiments, we limit the number of each link type. The reason for that is that the potential for the network cost can 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.

6.2 Optimal Number of Links

The goal of the first experiment is to determine the optimal number of each link type for three different 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 $8 \times 8$ 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.

### 6.2.1 Networks with Uniform Random Traffic

First, we evolved networks under random uniform random 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 $iR = 0.6$.

![Figure 6.2: 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 $iR = 0.6$. All ten different link types are used when we give more weight to $TP$. Short-range links of type 1, 2, and 3 are frequently used over the majority of the weight value $w$ to absorb the network traffic.](image)
Figure 6.2 shows the resulting link type distribution for networks optimized for *WiringCost* and throughput *TP* with the aggregate objective function $f(w) = w \times \text{WireCost} + (1 - w) \times \text{TP}$. As one can see, optimal networks use a large number of low cost links, such as link type 1 and 2, to distribute the traffic and provide high-throughput at a low-cost instead of using a small number of expensive link types. Figure 6.4 shows the corresponding networks for one of the evolutionary runs. To distribute the network traffic, less expensive high throughput links of type 7 are frequently used for long distance communication and a large number of low-cost short-range links of type 1 and 2 are used for local communication to lower the cost.

![Diagram](image)

Figure 6.3: 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 $iR = 0.6$. A large number of high performance long-range links of type 8, 9, and 10 are used to reduce energy consumption when we put more weight in *E*. These long-range links reduce the average shortest path length.
We also evaluated the energy $E$ combined with $WiringCost$. As one can see from Figure 6.3, 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. Figure 6.3 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.
Six link types were used in this network. Among the others, a large number of short-range links of type 1 and 2 and less expensive high throughput links of type 7 are frequently used to distribute the network traffic.
6.2.2 Networks with Hot-spot Traffic

Next, we used hot-spot traffic [121] as a more realistic traffic pattern to compare with other traffic patterns. The two hot spots are defined as node 9 and 54. We set the hot spot probability to $p = 0.25$, i.e., 25% of the packets will be sent to the hot-spots.

![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 $iR = 0$. A higher number of high performance links of type 7, 8, 9, and 10 are used to improve the network performance compared to the evolved networks under uniform random traffic.](image)

Figure 6.5: 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 $iR = 0.6$. A higher number of high performance links of type 7, 8, 9, and 10 are used to improve the network performance compared to the evolved networks under uniform random traffic.

The result of optimizing the networks for WiringCost and TP are shown in Figure 6.5. The distribution plot shows that for an optimal network, a large number of short-range links are used throughout the weight range, to absorb the traffic at a lower cost.

The result for optimizing WiringCost and $E$ with hot-spot traffic are shown in Figure 6.6. Long-range links of type 9 and 10 are used more frequently for the
majority of the weight values $w$ to use less energy.

![Figure 6.6: 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 $iR = 0.6$. A large number of high-performance long-range links of type 9, and 10 are used to reduce energy consumption.](image)

6.2.3 Networks with Transpose Traffic

In this experiment, we evolve the network under transpose traffic [121] and compare the link type usage with other traffic patterns. In transpose traffic, the source $(i, j)$ and destination $(j, i)$ nodes pairs are located symmetrically to the diagonal in a matrix. We used a roulette wheel 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 $8 \times 8$-node networks.

Figure 6.7 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.

Figure 6.7: Heterogeneous link type distribution as a function of the weight $w$. The networks are optimized for $WiringCost$ and $TP$ with transpose traffic. Injection rate $iR = 0.6$. A higher number of high performance link types are used in the networks to provide high throughput compared to the evolved networks under uniform random and hot-spot traffic.

Figure 6.8 shows the result for networks optimized for $WiringCost$ and $E$. When energy is considered to be more important, the optimal networks use more long-range links because these are more energy-efficient. However, when we give more weight to cost, we see less long-range and more short-range links to lower the network cost.

### 6.2.4 Comparison and Discussion

We will briefly discuss and compare the results from Sections 6.2.1 to 6.2.3.
Figure 6.8: Heterogeneous link type distribution as a function of the weight $w$. The networks are optimized for $\text{WiringCost}$ and $E$ with transpose traffic. Injection rate $iR = 0.6$. A large number of high performance long-range links of type 9, and 10 are used to reduce energy consumption.

The total number of links (i.e., the sum of ten link types) used in the networks from the previous experiments is shown in Figure 6.9. 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 $8 \times 8$ nodes. As one can see in Figure 6.9 (a), the evolved networks with transpose traffic use slightly more 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 Figure 6.9 (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 the traffic patterns. Also, considering energy as an optimization factor results in using about half the number of links compared to networks where $TP$ and $\text{WiringCost}$ are considered.
Figure 6.9: 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 $E$. The evolved networks optimized for $WiringCost$ and $TP$ under transpose traffic use slightly more links compared to the evolved networks under uniform random and hot-spot traffic to absorb the network traffic. The total number of links used in the evolved networks optimized for $WiringCost$ and $E$ under three different traffics are similar. This suggests that our results are independent of the traffic scenarios.
Figure 6.10: 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 $E$. For weights $w$ between 0.2 and 0.5, the evolved networks optimized for $WiringCost$ and $TP$ use an average of 5.5 link types while the networks optimized for $WiringCost$ and $E$ use an average of 2 link types. Energy-optimized networks use high performance links of type 9 and 10 to reduce the energy consumption. When we give more weight to $WiringCost$, a lower number of various link types are used to lower the energy.
Figure 6.10 shows the total number of link types used in the networks. When the network is optimized for $TP$ and $WiringCost$, 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 $E$ and $WiringCost$, the total number of link types is different from the throughput experiment. As one can see from Figure 6.10 (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 network cost.

**Discussion:** 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 three or more heterogeneous links to increase the performance or lower the energy while keeping the cost low. To improve the network throughput at a low cost, a large number of short-range links are used to distribute the network traffic and a small number of long-rang links are used to communicate long distance. However, to lower the energy consumption, a few high performance long-range link types are also used. When we put more weight on $WiringCost$, the number of link types increases, but the number of links for each link type is reduced. The optimal number of different link types that is used in the majority of the weight $w$ is above 3.
6.3 Optimal Network for Throughput

In this section, we investigate the optimal network throughput for various traffic patterns and injection rates and compare the results with the networks with three different technology-driven link types and regular 2D mesh networks. We hypothesize that the network with ten different link types provide better performance compared to the other two networks. 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 $8 \times 8$-node networks with three different traffic patterns (see Section 3.3), namely uniform random, hot-spot, and transpose with a target injection rate of $iR = 0.6$. First we find an optimal 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 Figure 6.11, 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 technology-driven link types and with a $8 \times 8$-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, we do
not claim to model these link types accurately, which is beyond the scope of this dissertation. The goal is to work with abstract and technology-agnostic links to obtain results that are more fundamental and more broadly applicable.

Figure 6.11 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 an average of 12% better performance compared to the networks with three link types under three different traffic scenarios. 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 [121], but only for homogeneous networks. In Section 6.5 we will show that our heterogeneous networks perform better than any other homogeneous single link type networks.
Figure 6.11: 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. The evolved networks with ten different link types improve the throughput by 12%, 7%, and 17% under uniform random, hot-spot, and transpose traffic respectively compared to the networks with three different technology-driven link types.
6.4 Optimal Network for Energy Dissipation

As current technology allows us to integrate a large number of cores on a single chip, energy dissipation is another key 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 6.1 to find optimal networks under wiring cost and energy constraints with equal importance, i.e., \( w = 0.5 \), \( f(w) = w \times \text{WiringCost} + (1 - w) \times \text{energy} \). 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 an injection rate of \( iR = 0.6 \) 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 technology-driven link types and mesh networks.

As one can see from Figure 6.12, 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, Figure 6.12 shows that our evolved topologies with ten different link types consume less energy than the networks with three different link types.
Figure 6.12: Network energy of evolved topologies and mesh networks with different traffic scenarios. Injection rate $iR = 0.6$. The evolved networks with ten different link types show 66%, 25%, and 66% less energy consumption under uniform random, hot-spot, and transpose traffic respectively compared to the networks with three different technology-driven 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. Moreover, a network with more different types of link can reduce an average of 52% energy dissipation. Thus, as seen here and in Section 6.3, the evolutionary algorithm does a splendid job in obtaining networks that meet the target requirements while minimizing both energy and cost.

6.5 Network Performance Comparison with Mesh and Homogeneous Networks

Next, we evaluated the performance of networks with heterogeneous link type and compared them with both homogeneous link type networks and regular 2D meshes. In addition, we compared the results with networks with three different technology-driven link types. We define a homogeneous link type network as a network that only uses link type 1, but without a length restriction to also 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 [121]. The difference is that we consider cost while they did not.

Figure 6.13 shows the comparative results of ten different heterogeneous link type networks with regular 2D meshes, homogeneous link type, and three different technology-driven 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 three different link
type networks. Figure 6.13 (a) shows that our proposed networks provide similar or higher throughput compared to the networks with three different link types. Figure 6.13 (b) shows that when the weight value $w$ is less or equal than 5, our evolved topologies use less energy.
Figure 6.13: Performance comparison of heterogeneous link types 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 $w$. Our evolved topologies with heterogeneous link types under uniform random traffic perform significantly better for the majority of the weight value $w$ compared to regular 2D mesh networks. Moreover, when network performance and wiring cost are equally weighted ($w = 0.5$), our evolved network provides 14% more throughput and 17 times less energy consumption compared to a homogeneous link type network and the same or better performance compared to a three link type network.
Figure 6.14: Performance comparison of heterogeneous link types 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 $w$. Our evolved topologies with heterogeneous link types under hot-spot traffic perform significantly better for the majority of the weight value $w$ compared to regular 2D mesh networks. Moreover, when network performance and wiring cost are equally weighted ($w = 0.5$), our evolved network provides 21% more throughput and 13 times less energy consumption compared to a homogeneous link type network and the same or better performance compared to a three link type network.
Figure 6.15: Performance comparison of heterogeneous link types 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 $w$. Our evolved topologies with heterogeneous link types under transpose traffic perform significantly better for the majority of the weight value $w$ compared to regular 2D mesh networks. Moreover, when network performance and wiring cost are equally weighted ($w = 0.5$), our evolved network provides 10% more throughput and 13 times less energy consumption compared to a homogeneous link type network and the same or better performance compared to a three link type network.
Figures 6.14 and 6.15 show 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 addition, the networks with ten different link types perform better or similar than the networks with technology-driven heterogeneous link types.

The results in this section indicate that having multiple types of links improves the network performance and lowers the energy at a low cost.
6.6 Scalability

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

We are interested to find how the cost, throughput, 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 $8 \times 8$-node, $10 \times 10$-node, and $12 \times 12$-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 $8 \times 8$-node, 264 for a $10 \times 10$-node, and $12 \times 12$-node network respectively.

Figure 6.16 shows the results for the network throughput $TP$ and the energy $E$ of different sizes. As one can see, networks with ten different 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 due to network congestion. The total number of links and a longer length of the links increases the network cost, therefore, there is a limitation of inserting links to balance both performance and wiring cost.

Figure 6.17(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 Figure 6.17(c), our evolved topologies perform better than 2D regular mesh networks for large-scale networks.
Figure 6.16: 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. Our evolved networks outperform regular 2D mesh networks even on larger systems. However, with larger systems, the performance is not up to the level of smaller systems because the networks evolved with 50% weight on cost limits the use of long-range links.
Figure 6.17: Scalability for network sizes under hot-spot traffic. *WireCost* and *TP* is equally weighted \((w = 0.5)\) using the aggregate objective function: \(f = w \times \text{WiringCost} + (1 - w) \times \text{TP}\). For the network throughput per cost \((\text{TP}/\text{WireCost})\), our evolved topologies perform even better than 2D regular mesh networks for large-scale networks.
Figure 6.18: Scalability for network sizes under hot-spot traffic. WireCost and $E$ is equally weighted ($w = 0.5$) using the aggregate objective function: $f = w \times \text{WiringCost} + (1 - w) \times E$. For the network energy per cost ($E/\text{WireCost}$), our evolved topologies have lower energy requirements compared to 2D regular mesh networks and scale better.
The evolved topologies for optimizing energy $E$ and $WiringCost$ use significantly less energy compared to 2D regular mesh networks, but they also become expensive as system size increases. However, when we analyze the network performance in terms of energy per cost ($E/WireCost$), our evolved networks used less energy and also scale well compared to mesh networks as shown in Figure 6.18.
6.7 Model Variation

In this section, we change several model assumptions to illustrate that our obtained results are robust against such variations.

6.7.1 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. Figure 6.19 shows the non-linear cost mapping of each link type (see Figure 6.1 for a comparison with the linear mapping). In order to reduce 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.

![Figure 6.19: Definition of the ten different types of abstract links with a non-linear (i.e., exponential) cost mapping.](image-url)

Figure 6.19: Definition of the ten different types of abstract links with a non-linear (i.e., exponential) cost mapping.
We then compared the results with the linear model shown in Figure 6.1 in Section 6.1. For this experiment, we evolved optimal $8 \times 8$-node networks under uniform random 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. Figure 6.20 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.
Figure 6.20: 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. Even though the cost function has changed, the number of link types used is the same, but 1% throughput improvement is shown.
Table 6.2: Fixed technology cost for link type 1, 6, and 10.

<table>
<thead>
<tr>
<th>Technology cost $c$</th>
<th>Link type 1</th>
<th>...</th>
<th>Link type 6</th>
<th>...</th>
<th>Link type 10</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.5</td>
<td></td>
<td>3</td>
<td></td>
<td>5</td>
<td></td>
</tr>
</tbody>
</table>

6.7.2 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 this new and more realistic model. The cost formula is shown in Equation 3.2, Section 3.2. Table 6.2 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.

Figure 6.21 shows the results by optimizing $WireCost$ and $TP$ under uniform random traffic. As one can see, the main results of this dissertation remain unchanged. In particular, the throughput is almost identical and the link type distribution differs by one link type.
Figure 6.21: 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. One less link type is used compared to the evolved network without technology cost. The network throughput decreased by 1%.
6.8 Discussion

We presented a library of ten link types to design heterogeneous NoC architectures and showed that the current technology of interconnect fabrics is non-optimal for a heterogeneous setup.

Our results have shown that our optimal networks with ten different link types provide 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. The networks provided an average 12% more throughput and used 52% less energy compared to networks with three different technology-driven link types.

We provided the optimal number of link types, wire-length distributions, and optimal link placement for our evolved heterogeneous networks. From the wire-length distribution results, we observed that as system size increases, the number of long-range links increases to communicate between distant nodes, reducing the number of hops and increasing the network performance. In addition, a large number of short-range link types were used to absorb the traffic and provide high network throughput at a lower cost. To minimize the network energy, a small number of long-range link types were used to reduce the average shortest path. Our results also confirmed that our evolved heterogeneous networks scale up significantly better in terms of cost and performance.
Core Assignment for Heterogeneous NoC Architectures

In the Network-on-Chip (NoC) community, mapping an application onto a physical on-chip network structure in an optimal way is an important open problem [13, 18, 77, 105, 112, 161]. The reason is that the network performance will vary highly depending on where each core of the application is located. For example, two cores of the application which frequently communicate with each other, if placed far apart, will increase energy consumption and decrease the network throughput. Many research groups have presented different approaches to solve the core assignment problems as we discussed in Section 2.2. However, all their target network topologies were 2D regular mesh networks. To the best of our knowledge, no one has mapped applications onto irregular fabrics.

In Chapter 4 and Chapter 6, we showed that our irregular network topology with heterogeneous link types provides higher throughput and lower energy consumption at a lower cost compared to a regular mesh networks. In this chapter, we find an optimal resource mapping for our application-specific optimized evolved networks, obtained through our HCNEA framework (see Chapter 3) as target topologies and show the performance improvement compared to regular mesh networks. We will present the basic architecture of our core assignment framework and describe the experiments and results in the following sections.
7.1 Finding Optimal Placement in Heterogeneous NoC Architectures

An application is composed of a set of heterogeneous cores, such as processors, memories, I/O devices, etc., and each core of the application needs to be placed to a node in a physical target NoC architecture as shown in Figure 7.1. Finding an optimal placement for a given application onto a target topology to achieve high throughput and low energy dissipation is a key issue in the NoC community [13, 18, 77, 105, 112, 161]. For example, an application with 10 cores mapped to a target NoC architecture with 25 nodes has over 1.86 trillion permutations.

![Application core assignment on a target mesh NoC topology. This diagram illustrates one possible mapping scenario. Five cores in the application are mapped to 3 × 3-node mesh network.](image)

Figure 7.1: Application core assignment on a target mesh NoC topology. This diagram illustrates one possible mapping scenario. Five cores in the application are mapped to 3 × 3-node mesh network.

In order to find an optimal placement of an application onto our irregular evolved heterogeneous NoC networks, we use Evolutionary Algorithms (EAs) [51,
EAs have been successfully used in various complex problems where there is no good method for finding optimal solutions. In addition, EAs work well with problems where multiple solutions are required [62, 83, 85, 100, 174], for example, to find an optimal placement solution to meet several requirements, such as high throughput and low energy consumption.

Figure 7.2: Design flow of the Heterogeneous Complex Network Mapping (HCN-Map) framework. Our evolutionary algorithm platform is based on the ParadisEO framework [30] and we combine this framework with our own C++ HCN framework to evaluate the network metrics.

Our optimal core assignment framework, named Heterogeneous Complex Network Mapping (HCNMap), is shown in Figure 7.2. It has a similar structure to the HCNEA framework described in Section 3.5. The evolutionary algorithm platform is based on the ParadisEO [30] framework, which is a C++ while-box object-oriented framework dedicated to the reusable design of metaheuristics. We
combine this framework with our own C++ HCN framework (see Chapter 3) to evaluate the networks. The framework starts from placing application cores randomly in a target NoC architecture as an initial solution. Genetic operators, such as crossover and mutations, are then applied to generate new possible solutions with the goal to find the best solution. The crossover operator combines two parent individuals and creates two new children. The mutation operator randomly changes an individual to explore the design space. We have implemented the regular genetic operators, namely single-point crossover and simple mutation. We use our defined aggregate objective function that considers multiple factors (see Section 7.1.2) to evaluate the optimized mapped NoCs. Two factors, throughput and energy, are measured for our purpose. Details on the implementation of the framework is described in the following sections.

7.1.1 Initialization

The first step of the framework is generating a random initial population. The individuals of the initial population are randomly generated by placing application cores randomly on the nodes in the target NoC topology. In order to accomplish the optimal core assignment, there needs to be an appropriate genetic representation that is capable of handling all possible solutions to the problem.

The genome for each individual is a string of \( n \) integers on chromosome, where \( n \) is the number of nodes in the target network (node ID). The possible chromosome values range from 0 to \( n \). For example, the length of the individual shown in Figure 7.1 is 9. As multiple cores cannot be assigned to the same node in a NoC, no number can be used twice in an individual and all numbers (from 0 to \( n \)) must be used once. Figure 7.3 shows some of the possible initial solutions of Figure 7.2.
Each number in the genotype from 1 to $m$, where $m$ is the number of cores in the application, determines the core ID in the application mapped to the target NoC architecture. For example, as one can see from the above example (Figure 7.1), application core1 and core5 are placed on node8 and node7 in the target mesh network respectively.

After initial populations are randomly generated, a traffic matrix is generated to simulate traffic on the network to evaluate the network performance metrics.

### 7.1.2 Fitness function

In order to find optimal solutions for the target topology, we need to define a measure for the performance metrics and evaluate the performance of the optimized NoC architecture. A solution, e.g., a mapping, is the average network throughput is maximized and the energy dissipation is minimized. We then calculate the fitness value as a following:

$$f(w) = w \times \text{Throughput}_{\text{network}} - (1 - w) \times \text{Energy}_{\text{network}}, \quad (7.1)$$
where \( w \) is the weight factor that allows us to determine the importance of either of the two factors, and Throughput and the Energy are normalized.

In the HCNMap framework, our C++ HCN framework, (described in Chapter 3) is used to measure the network performance by running the traffic simulation on the optimized NoC topology.

Note that not all the nodes in the target NoC architecture are being mapped, therefore, in order to generate the traffic, source and destination nodes are only selected from the nodes that have been used for mapping.

### 7.1.3 Genetic Operators

We use the two standard genetic operators, crossover and mutation. The crossover operator combines two selected parent individuals and generates new children. In our framework, however, general crossover operator is not suitable. The reason is that the newly generated individuals may have repeated numbers in genomes, which would not represent valid mapping. For example, as one can see in Figure 7.4, both values of gene 9 and 2 appears twice in \( \text{child}_1 \). This means that for core9 in application is mapped on both node0 and 7 in the target topology. Moreover, values of gene 4 and 5 do not appear in \( \text{child}_1 \) which means that core4 and core5 in the application are not assigned on target network at all.

To ensure that the genomes are valid, we adopted a new crossover operator introduced by Poon and Carter [133]. The new crossover operator starts by choosing two parents randomly from the current population and constructs offsprings \( A_1 \) and \( A_2 \) by selecting a random crossover point in the first parent. The first part
Figure 7.4: General one-point crossover operator. A random crossover point is selected within a chromosome, then the chromosomes beyond this point are interchanged between the two parents to generate new children.

<table>
<thead>
<tr>
<th>Node ID</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parent₁</td>
<td>9</td>
<td>2</td>
<td>3</td>
<td>6</td>
<td>4</td>
<td>7</td>
<td>8</td>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td>Parent₂</td>
<td>8</td>
<td>5</td>
<td>4</td>
<td>3</td>
<td>7</td>
<td>1</td>
<td>2</td>
<td>9</td>
<td>6</td>
</tr>
<tr>
<td>Child₁</td>
<td>9</td>
<td>2</td>
<td>3</td>
<td>6</td>
<td>7</td>
<td>1</td>
<td>2</td>
<td>9</td>
<td>6</td>
</tr>
<tr>
<td>Child₂</td>
<td>8</td>
<td>5</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>7</td>
<td>8</td>
<td>5</td>
<td>1</td>
</tr>
</tbody>
</table>

of the genes from the first parent is assigned to offspring $A_1$ and the rest of the genes are assigned to offspring $A_2$. Then, we construct the second offsprings $B_1$ and $B_2$ by selecting all the genes that have not appeared in offspring $A_1$ and $A_2$ respectively in the order from the second parent as shown in Figure 7.5. As one can see, none of the genes in offspring $A_1$ and $A_2$ appear in $B_1$ and $B_2$ respectively. Then, by randomly selecting one element at a time from offspring $A_1$ and $B_1$, a new individual $child_1$ is generated. $child_2$ is generated from offspring $A_2$ and $B_2$.

The second genetic operator we use is mutation. The mutation operator randomly swaps the values of two genes from a selected parent in order to explore the design space. As one can see in Figure 7.6, node ID 1 and 6 are selected for mutation and the values of location 2 and 8 are swapped to generate a new individual, $child_1$. 
7.1.4 Selection

The ParadisEO platform supports various functions of selection. In our HCNMap framework, we calculate the fitness value for each individual and use deterministic tournament selection to select individuals with the best fitness for the next generation by comparing the fitness value of each individual.
7.1.5 Population size and evolutionary runs

For all our experiments, we observe the generations to converge for populations of 60 within 7,000 and 8,000 generations as shown in Figure 7.7. Therefore, unless otherwise stated, we use a population of 60 individuals evolved over 10,000 generations. The results are averaged by running 10 evolutionary runs with 10 different initial populations.

Figure 7.7: Fitness as a function of generation. Using a population of 60, convergence occurs in an average of 7,500 generations.
7.2 Network Performance Comparison with Mesh-based NoC Architecture

The goal of this section is to determine that application-specific NoC architectures provide better performance in terms of throughput and energy compared to regular 2D mesh networks. To show our hypothesis, we (1) find an optimal network with heterogeneous link types for a given application, (2) use the obtained application-specific heterogeneous NoC as a target network to find optimal mapping solution for a given application, and (3) evaluate the optimal mapped NoC and compare the network performance with a regular mesh network.

For the heterogeneous link types, we use three different technology-driven link types (see Section 4.1), which can directly be mapped on actual technology to build optimal heterogeneous NoC architectures.

In this chapter, we consider a multi-media SoC design of a 38 cores TVOPD (Triple Object Plane Decoder) where the communication graph of the benchmark presented in Figure 7.8 is derived from [114]. Each node represents a core, and the weight of link represents the bandwidth between the cores. There are three decoders that work in parallel with two extra memories to improve the performance.

We first use our HCNEA framework (see Section 3.5) to find an optimal heterogeneous NoC architecture for the given TVOPD application. We evolve $8 \times 8$-node networks under TVOPD communication traffic with an injection rate of $iR = 0.6$. Figure 7.9 shows the network optimized for WiringCost and throughput $TP$ by equally weighting importance ($w = 0.5$) using an aggregate function $f(w) = w \times wireCost + (1 - w) \times TP$. After an evolved network is obtained, we use the heterogeneous NoC architecture as the target system to map the TVOPD
application. To find an optimal solution for mapping, we use our HCNMap framework discussed in Section 7.1. We use the aggregate function defined in Equation 7.1 to obtain optimal mapped NoC architectures that support high throughput and low energy consumption.

Figure 7.8: Communication graph for the 38 cores TVOPD. Node and weight of the edge represents core and the communication bandwidth respectively. Redrawn from: [114].
Figure 7.9: 8 × 8-node optimal evolved network for TVOPD application. WiringCost and TP is equally weighted (w = 0.5) using an aggregate objective function: \( f = w \times \text{WiringCost} + (1 - w) \times TP \). Black links: type 1; green links: type 2; red links: type 3.

Figure 7.10 shows the network performance comparison of two different target NoCs, heterogeneous and mesh networks. The given application is mapped on the 8 × 8-node evolved network for optimizing throughput TP and energy E. As one can see from Figure 7.10, when the application is mapped on the evolved networks with heterogeneous link types, the evolved networks provide higher throughput and lower energy consumption compared to regular 2D mesh networks. When TP and E are equally favored for mapping, our evolved heterogeneous network has a near 48% throughput improvement and 100% less the less energy dissipation over a mesh network.

Next, we consider the SPLASH-2 benchmark FFT [165] application as another test case. A 8 × 8-node network is evolved under FFT traffic by optimizing WiringCost and TP with equal importance (w = 0.5). As one can see from
Figure 7.11, the result for optimizing $TP$ and $E$ with equally weighted provides 45% higher throughput and 49% lower energy consumption compared to a mesh network. When $TP$ is only considered, we obtain 90% throughput improvement. Moreover, 57% less energy consumption is observed in the network optimized solely for energy ($w = 0$).

According to the above experimental results, we observe that application-specific networks, i.e., a network optimized for a given application provides better performance compared to regular 2D mesh networks. This confirms that finding an optimal network for a specific application is important to support the system requirements.
Figure 7.10: Performance comparison of heterogeneous NoC and mesh topologies as target NoCs. The TVOPD application is optimally mapped on two different target NoC architectures, i.e., evolved heterogeneous and mesh networks. $w = 0$ means that throughput only is considered. $w = 1$ means that energy only is considered. All application-specific NoC architectures with heterogeneous link types have a near 48% throughput improvement as well as 100% less energy consumption compared to regular mesh NoC topologies. $WiringCost$ for the evolved network is 207.66.
Figure 7.11: Performance comparison of heterogeneous NoC and mesh topologies as target NoCs. The SPLASH-2 benchmark FFT application is mapped on two different target topologies, i.e., evolved heterogeneous and mesh networks. When throughput only is considered \((w = 1)\), the evolved network provides 90% more throughput compared to a mesh network. Energy-aware evolved network consume 57% less energy than a mesh network. The \textit{WiringCost} for the evolved network is 188.70.
7.3 Network Performance Comparison of Heterogeneous Evolved Networks with Multi-traffics

In this section we consider application-specific heterogeneous NoC architecture optimized for multi-traffic scenarios. The goal is to show that multi-traffic-base evolved networks support better performance for multi-applications.

In Section 7.2, our optimized heterogeneous networks were evolved for one specific traffic, providing a high performance network for the given specific application. When a different application is mapped on the evolved network, the network may not be at its optimal performance for this application. As one can see from Figure 7.12, for example, the network optimized for TVOPD application provides higher throughput and lower energy consumption for the TVOPD application compared to the other two NoC topologies, FFT and mesh evolved network. This is because the network was optimized specifically for the TVOPD application. However, when we map the FFT application on the evolved network optimized for the TVOPD application, throughput drops by 14% and energy usage increases by 18% compared to the evolved network optimized for the FFT application. This confirms that an application-specific NoC architecture provides the best performance for a given application, which was to be expected.

However, our framework can evolve a network under multi-traffic scenarios to obtain a network that supports the system requirements for multiple applications. In the next experiment, we evolve an 8 × 8-node network under both TVOPD and FFT application traffics by optimizing WiringCost and TP equally weighted \( w = 0.5 \). We then used the obtained heterogeneous NoC architecture as a target network to map the given applications one at a time. As one can see from Figure 7.13, the obtained optimal network evolved under multiple traffics provides higher
throughput and lower energy dissipation for both applications. The network provides 6% more throughput and consume 9% less energy for the TVOPD application and 15% more throughput and 18% less energy for FFT application. However, a higher number of links is used to obtain the optimal network for both application traffics, therefore, the evolved network is 25% - 35% more expensive than a single application-specific NoC architecture. This illustrates that the network cost needs to be taken into account to construct a NoC architecture that supports various applications.
Figure 7.12: Performance comparison of two different applications mapped on the application-specific networks. Two applications, TVOPD and FFT, are mapped on three different NoC architectures. The results show that the application-specific NoC architectures support the best performance for the given application.
Figure 7.13: Performance improvement of the obtained heterogeneous network evolved under two different application traffics for given applications. Multi-traffic based evolved NoC architecture provides higher throughput and lower energy dissipation, however, the network becomes more expensive. The \textit{WiringCost} for the evolved network is 256.93.
7.4 Discussion

In this chapter we presented the basic architecture of our heterogeneous complex core assignment framework and show the benefits of using application-specific NoC architecture as a target system. Many research groups have studied finding an optimal mapping solution for a regular 2D mesh network as a target system. We used our evolved heterogeneous NoC architectures as a target system and found an optimal mapping for a given application. Our evolved network is optimized for a specific traffic and it is irregularly interconnected with three different technology-driven link types. To the best of our knowledge, no one has explored irregular NoC architectures with heterogeneous link types as a target topology for application mapping.

According to our experimental results, we showed that the evolved topologies optimized for specific-application provides average 45% more throughput and average 70% less energy consumption compared to regular 2D mesh networks. This verifies that an application-specific NoC architecture improves the network performance and it is necessary to meet the complex system requirements.
8

Conclusions and Future Work

In this chapter, we conclude by summarizing the main results of the work undertaken in this dissertation. We also present possible future research directions.

8.1 Conclusions

In this dissertation, we presented a comprehensive study on the benefits of heterogeneous link types in a generic networks-on-chip 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, we do not construct a network on top of a regular mesh topology. Instead, the evolutionary algorithm can place links without restriction, which allows to explore the entire search space.

We have shown in Chapter 4 that our optimal networks based on three different technology-driven heterogeneous link types provide a higher throughput and a lower energy consumption compared to both homogeneous link type networks and regular 2D mesh networks under uniform random, hot-spot, and transpose traffic patterns. We showed that the obtained optimal network uses high-throughput long-range links to distribute the network traffic. While the link types are kept abstract on purpose to make our results and the framework applicable to a broad range of technologies, the presented technology-driven links can rather straightforwardly be mapped to current interconnect technology. We have also shown
that, although the networks are optimized for a specific traffic pattern, they still outperform when used on other common traffic patterns, i.e., applications. In addition, we used the GEM5 framework [22] to run realistic traffic scenarios on our evolved networks. The results show that our networks with heterogeneous link types provide better performance compared to homogeneous and mesh networks. This confirms the benefits of using heterogeneous link types and verifies our hypothesis that our obtained heterogeneous networks outperform homogeneous and regular 2D mesh networks.

Designing large-scale heterogeneous Networks-on-Chip (NoCs) for irregular structure based applications often involves sophisticated optimization techniques that lead to unstructured networks. Such networks are hard to understand because they were not built with common engineering knowledge. In Chapter 5, we analyzed the network topology and structure by using existing complex network techniques, such as community detection, modularity, and small-worldness, and determined interesting network structure and properties. We have shown that our evolved heterogeneous networks have sub-community structures 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. This confirms that although the topology of our evolved heterogeneous NoCs appears to be unstructured, the networks are in fact very structured. If one use the right techniques to uncover the structure that is not immediately visible to the human eye.

In Chapter 6, we have shown that our optimal networks based on ten different heterogeneous link types provide even higher throughput and lower energy consumption, not only compared to homogeneous link type networks and regular 2D mesh networks, but also compared to the evolved networks with current technology
based interconnect. The networks obtained 12% and 52% improvement in throughput and energy respectively compared to networks with three different technology-driven link types. These results confirm that the current interconnect technology is non-optimal for future heterogeneous NoC communication architectures. We have also shown that the evolved heterogeneous networks perform even better in terms of throughput per cost \((TP/WireCost)\) and energy per cost \((E/WireCost)\) for large-scale networks. This confirms that our evolved heterogeneous networks scale up significantly better in terms of cost and performance.

In Chapter 7, we have shown that the obtained application-specific NoC architectures with heterogeneous link types provide high throughput and low energy dissipation compared to regular mesh-based networks. These results confirm that application-specific NoC architectures are necessary to manage heterogeneity for future SoC designs to support different performance, energy, and memory size of components.

We conclude by giving brief answers of the questions posed in the introduction. First, our results clearly confirm that irregular networks with three or more heterogeneous link type are beneficial compared to homogeneous and regular 2D mesh networks. We have provided optimal number of heterogeneous link types, optimal link type distributions, and optimal link placements in this dissertation. We can also confirm that the networks with heterogeneous link types scale better compared to regular 2D mesh networks. This is partially true because of the small-world property. We were able to confirm that a sub-network structure evolves and that the different link types are efficiently used to connect the sub-networks. Last but not least, we provided an optimal placement of target applications on our evolved heterogeneous NoC architectures and showed that application-specific NoC
fabrics are beneficial for target applications compared to regular mesh networks in terms of throughput and energy.

We have shown that an application-specific NoC architecture with heterogeneous link types can solve multi-hop communication issues along with cost efficiency of high performance NoC design. We believe that our results are relevant for the design of NoCs with emerging interconnect fabrics and will help drive the development of future large-scale NoC design. 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. This, however, is beyond the scope of this dissertation.

8.2 Future Work

In recent years, 3D NoC architectures were proposed as another attractive as promising solution for designing large and complex Systems-on-Chips (SoCs) [47, 58,156]. 3D NoC architecture can be designed by stacking multiple 2D layers with short vertical interconnection between the layers. This 3D NoC structure allows to increase transistor density and reduce the overall wire length while increasing network performance in terms of throughput and energy consumption. However, because of such dense structure from vertical layers and smaller die area, power density of the chip increases. As a result, the chip temperature increases exponentially which has a negative effect on the interconnect delay, power consumption, reliability of the device, and thus requires additional cooling that adds to cost. To solve the thermal challenges in 3D designs, a thermal-aware NoC architecture that provides high performance under a given power budget is essential for designing
future 3D NoCs [123].

We hypothesize that adopting our proposed an irregular application-specific 2D heterogeneous NoC topology in 3D NoC design will enhance the network performance compared to 3D regular mesh-based networks. According to our preliminary experimental results, a 3D mesh-based (4×4×4-nodes) homogeneous network outperforms up to twice the network performance of the 2D mesh-based (8×8-nodes) homogeneous network under uniform random traffic. Therefore, applying our heterogeneous NoC approach to 3D NoC architecture seems a very promising way to further improve the performance of networks. In addition, with our developed framework, designers can explore 3D heterogeneous NoC design space to find optimal number of vertical links and their placement in relation with energy consumption.
References


[88] S Kaushik, A Singh, W Jigang, and T Srikanthan. Run-time computation and communication aware mapping heuristic for NoC-based heterogeneous


time embedded systems. In Proceedings of the International Parallel and
Distributed Processing Symposium, pages 78–85, April 2003.

of an extensible network testbed. In Proceedings of the 2010 Ninth IEEE
International Symposium on Network Computing and Applications, NCA ’10,


programs: Characterization and methodological considerations. In ACM

block data transfer in cache-coherent multiprocessors. AMC SIGPLAN

programs: characterization and methodological considerations. SIGARCH

network-on-chip through contention-aware input selection. In Proceedings of
the 2006 Asia and South Pacific Design Automation Conference, ASP-DAC


