FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Layout aware router design and optimization for Wavelength-Routed Optical NoCs

Alexandre Carvalho Truppel

Mestrado Integrado em Engenharia Eletrotécnica e de Computadores

Supervisor @ TUM: Tsun-Ming Tseng

Supervisor @ FEUP: José Carlos Alves

July 2018

Resumo

Redes óticas em circuitos integrados (*Optical Networks-on-Chip*) são uma solução promissora para conexões de alto desempenho entre múltiplos *cores* de processamento pois oferecem menor latência e maior largura de banda que as tradicionais redes elétricas integradas. *Wavelength-Routed Optical Networks-on-Chip* conseguem adicionalmente dar garantias de desempenho que são especialmente importantes em aplicações onde a minimização da latência é critica.

No entanto, o *design* de *WRONoCs* apresenta novos desafios à área da automatização do projeto de sistemas eletrónicos integrados que ainda estão longe de resolvidos. O processo de conceção de *WRONoCs* passa por várias etapas de síntese, otimização e validação enquanto são simultaneamente considerados vários fatores de qualidade e desempenho tais como interferência, consumo energético nos componentes elétricos e óticos e paralelismo de *bits*. Este processo representa um problema de otimização complexo que, até ao momento, tem sido simplificado consideravelmente executando cada uma das etapas em sequência em vez de todas em conjunto. Todavia, isto leva a soluções cuja qualidade fica aquém do possível e desejável. Esse facto foi já demonstrado em várias publicações.

O presente trabalho propõe uma nova abordagem de otimização que combina as duas primeiras etapas de síntese e otimização do *design* de *WRONoCs* e que também permite melhorias adicionais futuras como, por exemplo, incorporar na abordagem as etapas posteriores de *design*. Este novo processo baseia-se num modelo de programação linear para otimizar um *template* da configuração física da rede. A programação linear tem várias vantagens que a tornam a ferramenta indicada para enfrentar este problema. Neste trabalho diversas técnicas de redução do modelo usado são também apresentadas e testadas. Adicionalmente, um conjunto de ferramentas para apoiar o *design* de *WRONoCs* foi desenvolvido e é também apresentado. Quando comparado com o estado da arte este novo processo de otimização atinge reduções notáveis de 50% nas perdas óticas na rede.

Abstract

Optical Networks-on-Chip are a promising solution for high-performance multi-core integration, with better latency and bandwidth than traditional Electrical Networks on Chip. Wavelength-Routed Optical Networks-on-Chip offer yet additional performance guarantees which are especially sought-after in latency-critical applications.

However, WRONoC design presents new Electronic Design Automation challenges which are currently far from being fully addressed. The design flow of WRONoCs must go through multiple synthesis, optimization and validation steps while simultaneously taking into account various performance factors such as crosstalk, optical power consumption and bit parallelism. This is a complex problem that so far has been considerably simplified by, among other things, considering each step sequentially instead of conducting the entire synthesis and optimization process at once. This leads to substantially sub-optimal solutions, a fact that has been shown multiple times in previous work.

The present research introduces a new optimization procedure that combines the first two synthesis and optimization steps in WRONoC design while also leaving space for future improvement, i.e., for expanding the procedure to also consider the subsequent steps. This new procedure is based on a linear programming model that optimizes a WRONoC physical layout template. The use of linear programming has multiple advantages which make it an appropriate choice to tackle this problem. In addition, multiple model reduction techniques are also presented and tested. A toolchain was also developed to aid in the design process. When compared to the state of the art design procedure this new method shows a remarkable 50% reduction in maximum optical loss.

Acknowledgements

Firstly, I would like to thank my TUM supervisor, Dr. Tsun-Ming Tseng, for being unequivocally the best supervisor one can ask for. His attention to detail and sharp criticism allied to a remarkable patience to listen to my new ideas made it a joy to work in this area. I must also thank Prof. Ulf Schlichtmann for giving me the opportunity to work in his Electronic Design Automation research group.

Secondly, I want to thank my FEUP supervisor, Prof. José Carlos Alves, for his continued interest, feedback and help throughout this whole process.

My most sincere thanks to my family, especially to my grandparents for the financial support and my mother, Margarida Silva, for always being here for me, no matter what. Her unwavering care, kindness, loyalty, assistance and encouragement were crucial. Absolutely none of this would have been possible without her.

Finally, I would like to thank my girlfriend, Inês Nobre, for being just awesome in every way and my mates Luís Abreu and Nuno Fernandes for watching the second season of Westworld with me and for occasionally cooking for me too. Oh, and my CPU for actually doing all the hard work.

Alexandre Carvalho Truppel

"Premature optimization is the root of all evil."

Donald E. Knuth

Dedicated to my grandparents,
whom I deeply love
and admire

Dedicado aos meus avós,
que respeito e admiro
profundamente

Contents

List of Figures

List of Tables

Abbreviations

CPU	Central Processing Unit
EDA	Electronic Design Automation
ENoC	Electronic Network on Chip
FIFO	First-In, First-Out
GRU	General Routing Unit
GPU	Graphics Processing Unit
GWOR	Generic Wavelength-routed Optical Router
IC	Integrated Circuit
IP	Intellectual Property
LP	Linear Programming
MIP	Mixed Integer Programming
MRR	Micro-Ring Resonator
NoC	Network on Chip
ONoC	Optical Network on Chip
OPDN	Optical Power Distribution Network
P&R	Place and Route
PSE	Photonic Switching Element
SNR	Signal-to-Noise Ratio
SoC	System on Chip
TSV	Through-Silicon Via
WDM	Wavelength Division Multiplexing
WRONoC	Wavelength-Routed Optical Network-on-Chip

Chapter 1

Introduction

The extremely fast development of Integrated Circuit (IC) fabrication technologies seen in the last decades, allied to the unending need for more processing power, has made ICs drastically smaller while at the same time increasing their complexity and versatility. Their exponential evolution since the 70s, as anticipated by Moore's Law, has all but allowed mankind to live in a fully digital age.

Many ICs no longer perform one single task, containing instead a multitude of different subsystems which all together allow a single IC to perform very complex and multifaceted jobs. It has become commonplace for a single IC to be composed of such different components as analog or radio frequency signal processing blocks, memory banks, video encoders, dedicated cryptography hardware, multiple CPUs or GPUs, networking interfaces and other peripherals, etc. ICs at this scale of complexity are commonly named System on Chip (SoC).

Any complex system composed of multiple minimally independent modules, each with a well-defined interface and set of functions, requires a communication layer. The more independent and distributed each module is, the bigger the burden placed on that layer. "Simpler" ICs such as embedded micro-controllers commonly use communication buses to connect all components. However, many SoCs have become so complex – they potentially contain such differing components, for example with respect to clock frequencies, or by virtue of being developed by different companies which will not share their intellectual property – that a complete communications network inside the IC [2] becomes unavoidable. A Network on Chip (NoC) is a communications network whose purpose is to interconnect all SoC components in a fast, scalable and energetically efficient way.

Traditionally, NoCs have been limited by the use of electrical connections in their throughput, latency and energy consumption [3]. To overcome these barriers, research has been conducted in the last decade into silicon-photonic technology. This allows optical waveguides and optical signals to replace long metal interconnects in ICs, effectively opening the door to Optical Networks-on-Chip (ONoCs). These can achieve much higher throughput (easily on the order of tens of Gbps *per wavelength*) allied to extremely low signal delay with a much decreased dynamic power consumption and, potentially, lower total power consumption than traditional electrical

NoCs [4, 5]. ONoCs can be categorized into active ONoCs or passive ONoCs, also known as Wavelength-Routed ONoCs[1], depending on how the routing is achieved.

This thesis focuses solely on Wavelength-Routed ONoCs and its intention is to address a substantial gap in their design and optimization workflow. A novel way of thinking about the logical design and physical layout of WRONoCs is proposed and a linear programming model is developed to perform the optimization tasks with significant improvements on the state of the art tools.

As ONoCs become more prevalent, the ideas, algorithms and tools developed in the present research will become increasingly important.

1.1 Thesis structure

Having given a brief introduction to the history of ONoCs and the underlying motivation for their use, Chapter 2 gives the necessary background on ONoCs that is the starting point for this thesis.

Chapter 3 then explains in detail the various facets of the optimization problem at the center of WRONoC design, mentioning previous work in this area and identifying current gaps in scientific knowledge. A formal definition of the layout-aware router design and optimization problem, which is at the core of this thesis, closes the chapter.

Chapter 4 presents the chosen approach, from a theoretical standpoint, to solve the optimization problem, and Chapter 5 then details the linear programming based optimization algorithms and procedures developed to reach optimal solutions.

Chapter 6 explains the toolchain written for the development of WRONoCs that, among others, implements the optimization algorithms. It also includes relevant comments on other aspects of the design workflow of WRONoCs and provides examples.

Chapter 7 compares this new method with the state of the art in WRONoC optimization. It also analyses, from multiple angles, various algorithm performance metrics. A complete example result is presented.

Finally, Chapter 8 draws a global conclusion for this research project and outlines multiple future additions and improvements this new approach could benefit from.

[1] Both categories will be introduced in detail in Chapter 2.

Chapter 2

Optical Networks-on-Chip

In this chapter an introduction to Optical Networks-on-Chip is given. First, the physical construction of ONoCs is described. Then, the major elements that comprise ONoCs are listed and each is characterized. At the end, the two main categories of ONoCs are presented with examples from published research.

2.1 Physical architecture of ONoCs

The physical integration and fabrication of ONoCs in ICs is done by 3D stacking multiple layers. With current technology, the electrical layers are placed first on the pile. These contain, first, the silicon wafer where the transistors are etched, then multiple layers of metal interconnections. This follows the common CMOS fabrication process for ICs. Next, a cladding silicon layer is deposited. Finally, the optical layers are stacked. The connections between the optical layers above the cladding layer and the electrical layers below are done with Through-Silicon Vias. A cut of the whole stack is shown in Figure 2.1.

Many architectural and design decisions influence the exact layers required to build an ONoC, and, as such, the final structure of the IC. An overview of possible layer types is given below.

Optical routing layer. This layer is required on all ONoCs because it is where the optical routing elements are placed, where the network paths between nodes exist and where the routing takes place. In many cases, laser power is also distributed through this layer (see Section 2.3.2).

Control layer. This layer is responsible for activating/deactivating the network paths of the routing layer as needed. Optical networks that require this layer are named *active* networks (see Section 2.4), whereas optical networks that don't are *passive*, or *Wavelength-Routed*, networks (see Section 2.5). This layer can itself be optical [7], in which case it is also present on the optical portion of the stack. However, by far the most common option is to make the control layer electrical, in which case it is placed along with the rest of the logic circuitry on the electrical portion [8, 9].

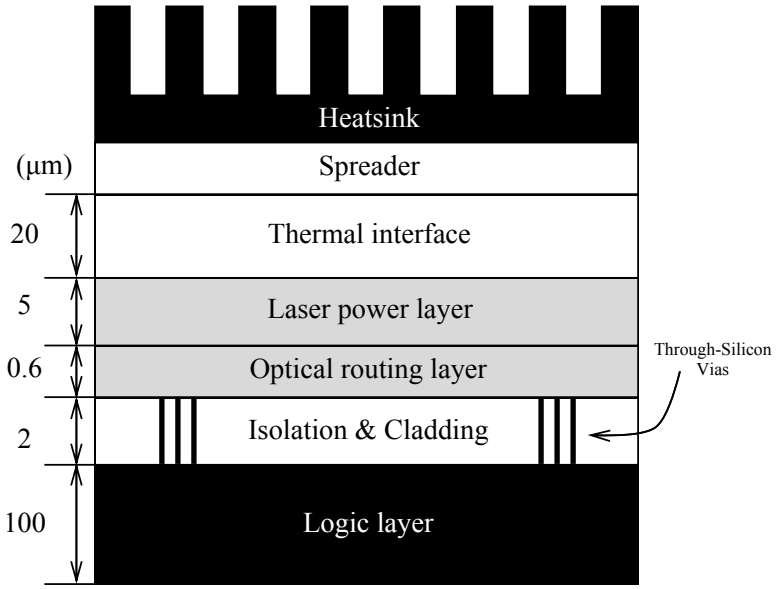


Figure 2.1: ONoC construction through 3D stacking of multiple layers (adapted from [6]).

Laser power layer. Most commonly, ONoC architectures consider the laser sources required for the ONoC to be placed off-chip [3], but some recent work has been done in placing the laser sources on-chip [10]. In these cases the sources are placed on a layer above the optical routing layer and the laser power is routed downwards to the layer below [6].

2.2 Optical routing layer

By far the most important layer for this work is the optical routing layer, since that is where the entire optimization problem unfolds. For successful data routing between network nodes to take effect, four classes of elements must be placed on this layer. These are:

Modulators to perform the conversion from the electrical domain to the optical domain at every transmitting node.

Demodulators to perform the conversion from the optical domain to the electrical domain at every receiving node.

Waveguides that act as optical wires to direct optical signals through the routing layer.

Optical routing elements to perform the routing operations by transferring optical signals between waveguides.

2.2.1 Waveguides

These can be constructed out of multiple materials, such as silicon on a silicon-on-insulator process [2, 11, 12] or gate poly-silicon [11, 13]. Just like optical fibers, they serve as a propagating

medium for an entire range of frequencies of light, thus being responsible for one of the most useful features of ONoCs: multiple data streams transmitted in parallel without interference through the same waveguide, so long as they all use unique wavelengths. This is called Wavelength-Division Multiplexing (WDM) and is one of the primary reasons ONoCs can achieve such high bandwidths. Only one optical signal of each wavelength can be transmitted at the same time through one waveguide, otherwise heavy interference between signals will occur. Waveguides can also propagate light signals in both directions at the same time, once again provided they are of different wavelengths.

2.2.2 Optical routing elements

The optical routing elements in ONoCs are Micro-Ring Resonators (MRRs). These are silicon micro-structures in the form of a ring on the scale of about $10\mu m$ to $50\mu m$ in diameter [14]. Each MRR resonates with a certain set of wavelengths, as shown in Figure 2.2[1]. The values for the resonance wavelengths depend on the material and structural properties of the MRR, one of the more relevant ones being its radius [14]. The injection of electric charge into a p-n junction at the base of the ring can also tune the MRR to different sets of resonance frequencies, which is useful for cancelling temperature-induced changes in the resonance characteristics or to effectively turn on or off the MRR for certain wavelengths [13, 4, 15].

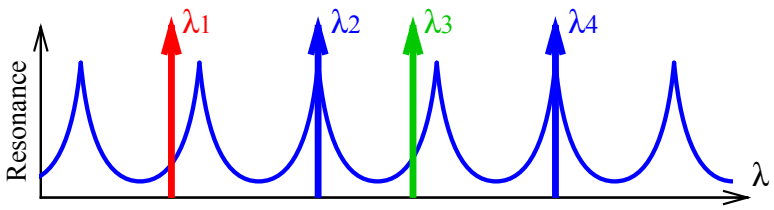


Figure 2.2: MRR resonance characteristic as a function of wavelength [14]. In this example, the MRR resonates with λ_2 and λ_4, but not with λ_1 or λ_3.

Routing with MRRs works as follows: a light signal with a certain wavelength propagating on a waveguide close to a MRR with a matching resonance frequency will be coupled to the MRR and moved onto another waveguide also close to that MRR (this is regardless of current application to the MRR). Figure 2.3 demonstrates this effect with a MRR next to a crossing between two waveguides. As depicted, the MRR resonates with λ_1, but not with λ_2. Light signals come in from the left and leave through the right or the bottom. When the signals have wavelength λ_2, they stay on the same waveguide but, when they have wavelength λ_1, they switch waveguides, joining any other light signal already passing through the new waveguide. Each MRR can only route one light signal of each wavelength at a time.

[1] In many cases only one of the resonance frequencies of the MRR is considered, so phrases like "the resonance frequency of the MRR" or "the wavelength of the MRR" are correct at that level of abstraction.

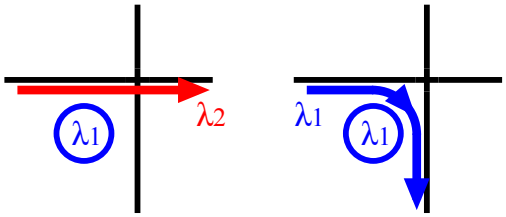


Figure 2.3: Wavelength routing using a MRR next to a crossing.

2.2.3 Modulators

These must be connected to a laser source through optical waveguides and to driver and control circuits on the electrical layer through TSVs. Modulators commonly consist only of MRRs which control the transfer of light from the Optical Power Distribution Network (OPDN – see Section 2.3.2) to the sending waveguide using on-off keying with signals provided by the control circuitry [13, 16, 14] – Figure 2.4.

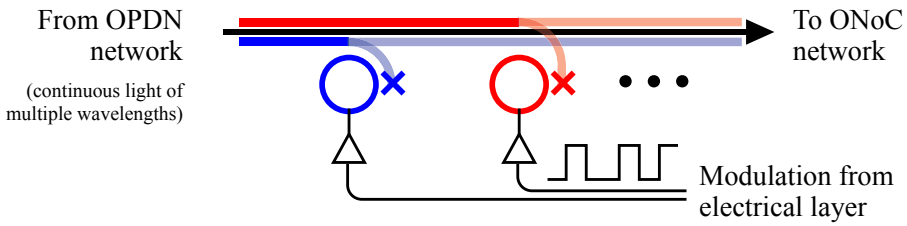


Figure 2.4: Modulator design using MRRs. Color indicates wavelength.

2.2.4 Demodulators

These must be connected to receiver circuits on the electrical layer. These are also commonly MRRs tuned to each received wavelength which redirect each signal to a specific photodetector. The photodetector then transforms the received light into electrical current, which is then picked up by the receiving electrical circuit [11, 14] – Figure 2.5.

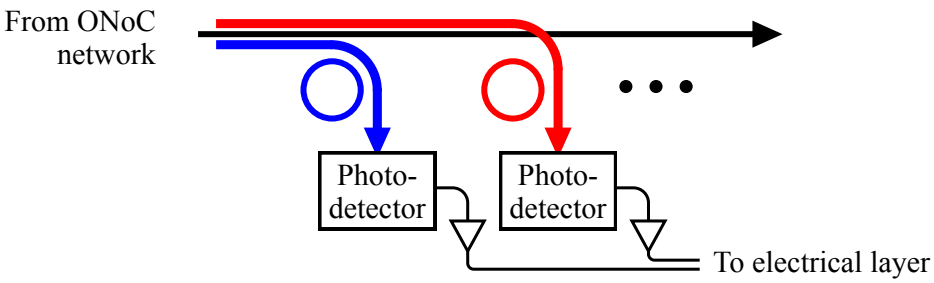


Figure 2.5: Demodulator design using MRRs and photodetectors. Color indicates wavelength.

2.3 Other ONoC components

2.3.1 Laser sources

Below follow the two main approaches for the laser sources used in ONoCs [10].

Off-chip. These are typically comb lasers, i.e., laser sources that emit multiple modes equally spaced along the frequency range. This laser type avoids the complexity and cost of packaging lasers on-chip but the connection from the external optical channel to the chip creates unavoidable coupling losses. Also, the power output for each wavelength emitted by the laser cannot be individually controlled. Hence, if significant power requirement disparities exist between wavelengths, the efficiency of this type of source is decreased.

On-chip. These are single-wavelength distributed-feedback laser arrays. In other words, a separate (on-chip) laser source exists for each wavelength. Each wavelength can have a separate power output as required by the router. However, this has non-trivial implications for OPDN design [3].

2.3.2 Optical power distribution network

According to the placement of the laser sources (on-chip or off-chip), laser power may need to be split and distributed to the nodes of the network. These cases call for an Optical Power Distribution Network (OPDN) to be present on the optical routing layer. This network is made up of waveguides and laser power splitters tuned to deliver the necessary amount of laser power to each transmitting node while minimizing power waste [17, 14, 3]. Figure 2.6 shows an example of an OPDN connecting an off-chip laser source to 11 nodes.

The fact that the OPDN is placed on the same layer as the ONoC makes its design in many cases crucial. This is because the waveguides distributing the laser power may cross with the ONoC waveguides and increase the overall amount of power used [17] (more detail of what factors affect power usage is given in Section 3.1.4).

2.3.3 Optical-electrical network interfaces

ONoCs have most of their control logic outside of the actual network, i.e., on the interfaces between the optical network and the electrical nodes[2] [16]. Because of that, typical duties of the network interface when transmitting include: buffering data to send, transferring/synchronizing data between the clock domains of the Intellectual Property blocks (IPs) and the ONoC, and serializing the data with the desired bit parallelism. The receiving portion of the network interface must perform essentially the same steps but in reverse, i.e., deserializing data and transferring it to a FIFO queue in the electronic clock domain. Both sides also require mechanisms to synchronize the optical transmission. One possibility is to have an extra wavelength transmitting a clock signal

[2] WRONoCs, as explained later, actually have *no* control logic inside the network.

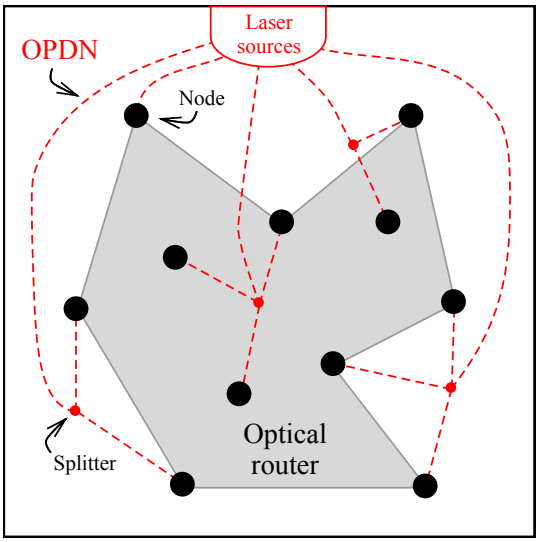


Figure 2.6: Example of an OPDN connecting an off-chip laser source to 11 nodes.

generated by the transmitter [16]. Finally, to avoid overloading the receiving nodes with data, some kind of flow control system must be implemented [16].

2.4 Active ONoCs

Active ONoCs are ONoCs that need a control layer in addition to the optical routing layer. This control layer is responsible for turning on and off the required MRRs so that the correct optical path between the sending and receiving nodes is created before the optical signal is sent.

Many different topologies of active ONoCs have been presented, but most work by having one instance of the same router block per node. This block is commonly a 5x5 router, i.e., it has four bi-directional connections to the North, South, East and West, and one bi-directional connection to its corresponding node (7x7 designs also exist, with additional "Up" and "Down" ports). Figure 2.7 shows the Crux [9] and Cygnus [8] 5x5 routers. These instances are then connected in a mesh [8], torus [18], fat-tree [19] or other configuration. Figure 2.8 shows an example of a mesh configuration with 5x5 router blocks.

In the examples given so far all MRRs are tuned to the same wavelength and only one wavelength is used on the entire network. However, these designs can be adapted to take advantage of WDM, examples of which have also been given in the literature [1].

The main disadvantage of active ONoCs is precisely that the routing is active. The need for a control layer adds extra power usage and latency to the network. Because paths must be set up before being used, conflicts between communication requests may be unavoidable so performance guarantees are few.

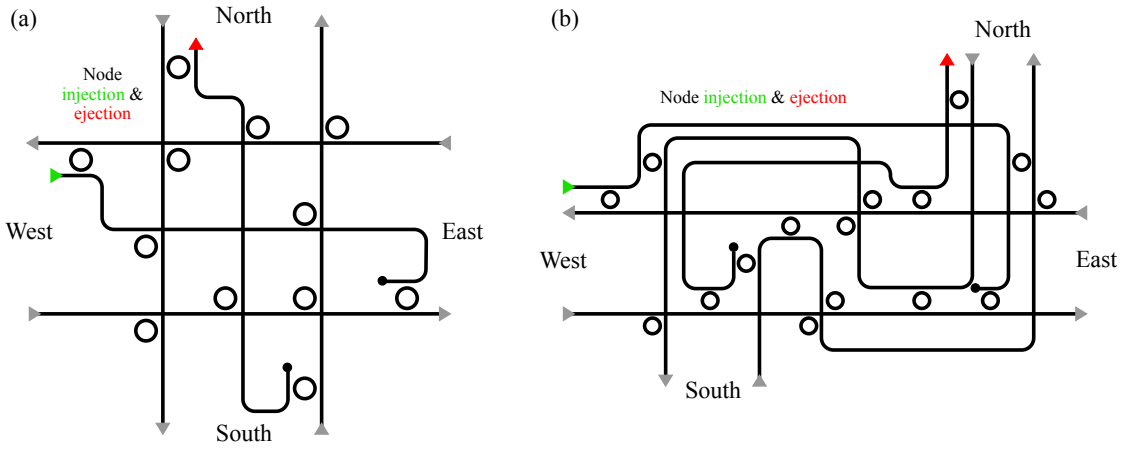


Figure 2.7: Examples of 5x5 router blocks for active ONoCs. (a) Crux router, adapted from [9]. (b) Cygnus router, adapted from [8]. Black dots are waveguide terminators.

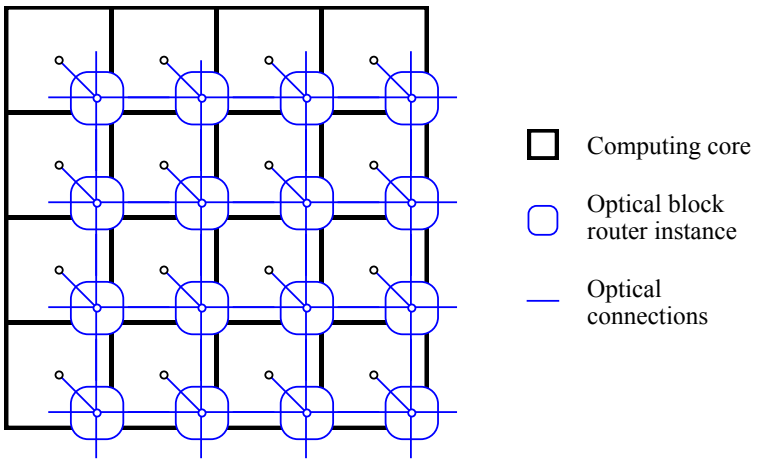


Figure 2.8: Example of an active ONoC using multiple instances of a 5x5 router block in a mesh configuration. Adapted from [8].

2.5 Passive (Wavelength-Routed) ONoCs

Contrary to active ONoCs, passive ONoCs do not require a control layer. Instead they use the wavelength of the optical signal for routing (hence the name "Wavelength-Routed"). As such the path of the optical signal is defined at design time by the signal origin in the router and by its wavelength. This deterministic approach to routing eliminates any latency due to path setup and tear-down present in active ONoCs. The lack of a control layer also lowers dynamic power consumption.

The determinism gives WRONoCs guaranteed performance. Multiple signals can use the same wavelengths, but the WRONoC is always designed such that no two signals using the same wavelength have colliding paths. Because of that, WRONoCs deliver contention-free communications.

2.5.1 Conceptual architecture

In general, any WRONoC router must obey the following three rules:

1. Optical signals cannot have equal wavelengths and colliding paths.
2. Every modulator must send all its optical signals in different wavelengths.
3. Every demodulator must receive all its optical signals in different wavelengths.

Rule one guarantees an absence of conflicts, as explained above. Rule two exists so that a modulator can send each optical signal to a different demodulator[3], and rule three exists so that each demodulator can distinguish which modulator each optical signal was sent from.

Depending on the communication requirements and the physical design of the router, each node of the network may have zero, one, or more modulators and demodulators. Figure 2.9 shows an example of the wavelength mapping for two modulators sending signals to four demodulators. Here, two things are noteworthy. Firstly, rules two and three given above are in fact obeyed. Secondly, this represents a broad WRONoC design and so modulator 1 (M1) and demodulator 1 (D1), for example, may belong to the same node of the network even though they are not drawn next to each other (in which case the wavelength λ_1 coming out of M1 is used for loopback, i.e., self-communication).

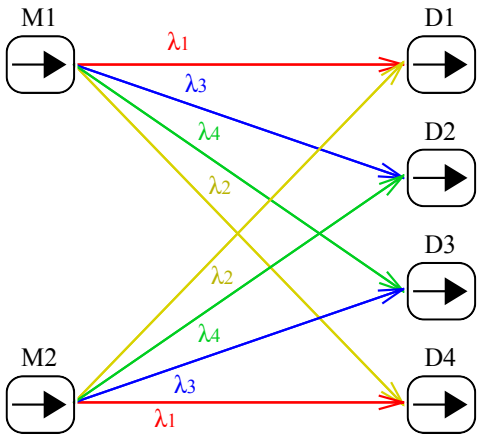


Figure 2.9: Wavelength usage mapping for communications between 2 modulators and 4 demodulators for WRONoCs.

2.5.2 Photonic Switching Element

Most WRONoC router designs presented in the literature are built with multiple instances of 1x2 or 2x2 Photonic Switching Elements (PSEs). These are a crossing between two waveguides where one or two MRRs are present, as shown in Figure 2.10(a) and Figure 2.10(b). Both MRRs have the same radius, and so the same resonance frequencies, which leads to the routing behaviour depicted

[3]Equal wavelengths and the same starting point unequivocally lead to equal paths, and so equal destinations.

in Figure 2.10(c) and Figure 2.10(d). By connecting multiple instances of PSEs between the modulators and demodulators in different ways, different logical topologies of WRONoC routers are possible.

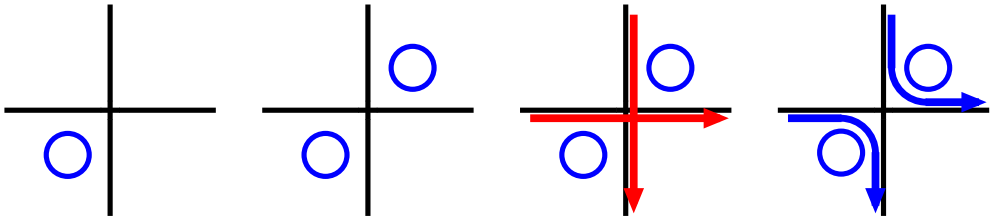


Figure 2.10: PSE structure and routing. (a) 1x2 PSE. (b) 2x2 PSE. (c) Optical path for wavelengths *not* in resonance with the MRRs. (d) Optical path for wavelengths in resonance with the MRRs. Color indicates wavelength.

2.5.3 Logical WRONoC topologies

The logical design space of WRONoC routers using PSEs under complete communication requirements (or complete communication except self-communication) has been explored by Mahdi Tala *et al.* [5]. Some points on that space are important router designs which have been presented and analyzed separately in the literature, as shown below.

Standard crossbar. The standard crossbar is the simplest way to fully connect N modulators and N demodulators. It consists of a square grid with N^2 PSEs of the 1x2 type using N wavelengths – Figure 2.11(a).

λ-router. The λ-router was one of the first topologies proposed in the literature [2]. It uses $\frac{N}{2} * (N-1)$ PSEs of the 2x2 type to connect N modulators to N demodulators with N wavelengths – Figure 2.11(b).

GWOR. The Generic Wavelength-routed Optical Router [20] connects N modulators and N demodulators, but needs only $\frac{N}{2} * (N-2)$ PSEs of the 2x2 type and $N-1$ wavelengths. This is because this topology does not support self-communication, i.e., each modulator has exactly one demodulator to which it cannot send optical signals – Figure 2.11(c).

Snake router. The Snake router [21] was manually designed to be easily placed on the optical routing layer for one specific set of node positions, but has since been cited and analyzed for other use cases [5, 14, 22]. This topology uses $\frac{N}{2} * (N-1)$ PSEs of the 2x2 type with N wavelengths, just like the λ-router – Figure 2.11(d).

Although these logical topologies mostly provide the same connectivity capabilities with small differences in wavelength and MRR usage, their different PSE connection structures lead to widely disparate results after placing and routing. This calls for a broader analysis of WRONoC routers, specifically one which also looks at the physical layout of the router [5, 21, 3].

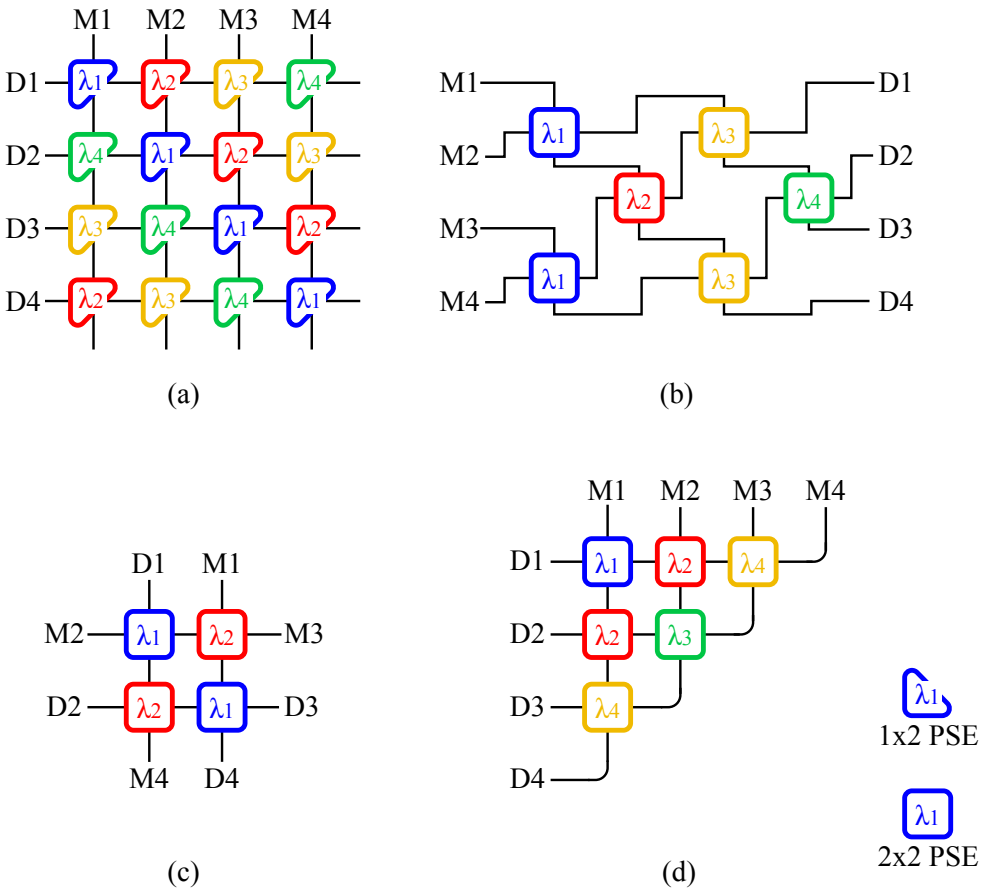


Figure 2.11: Four common PSE-based 4x4 logical WRONoC topologies. (a) Standard crossbar. (b) λ-router. (c) GWOR. (d) Snake router.

Chapter 3

Wavelength-Routed ONoC design problem

This research focuses solely on passive ONoCs, i.e., Wavelength-Routed ONoCs, and looks only at their optical routing layer design. Other topics such as modulation and demodulation techniques, physical integration with the electronic layers, network protocols, flow control, clock synchronization, etc, are out of the scope of this work. Even so, as is the case with any complex system built to satisfy high performance requirements, the various components on the optical routing layer raise many different concerns and optimization opportunities.

Having given an overview of those components and their working principles in the previous pages, this chapter now focuses on the WRONoC optimization challenges when designing this layer. It begins by laying out the major characteristics of WRONoCs influencing most their real-world performance. The main tasks required when designing an WRONoC are then listed and explained. Next, the major gap in the literature on the optimization of these tasks is presented and justified. The chapter ends with a formal definition of the WRONoC design problem tackled in this thesis.

3.1 WRONoC performance factors

3.1.1 Temperature resilience

Critical components in WRONoCs such as waveguides and MRRs are very sensitive to high temperatures and temperature variations. Rising power densities in ICs are making chip temperatures over 90 °C normal [23, 24]. At those temperatures, the optical conducting properties of waveguides may be altered. Temperature variations with respect to the reference temperature assumed at design time also change the MRR resonance frequencies [25, 23, 26]. In such cases the MRRs no longer respond correctly to the wavelengths they were intended for, and may even start acting upon other wavelengths, thus disrupting message flow and leading to a decrease in reliability.

Various possibilities to cope with this undesirable effect have been published. Active thermal management, such as MRR heaters, is commonly used [22, 27, 23]. For active ONoCs, algorithms

for on-line assignment of MRRs to account for red-shifting are a possibility [23]. For WRONoCs, research has looked into customizing P&R algorithms to make sure WRONoC elements avoid hot spots created by the electronic layers [24].

3.1.2 Crosstalk

Crosstalk is the effect by which one communication channel creates unwanted interference in the signals of other, separate, communication channels. In ONoCs there are two types of crosstalk: inter-channel and intra-channel. Inter-channel crosstalk happens when noise is added to a signal of a wavelength different from the wavelength of the original signal. Conversely, intra-channel crosstalk happens when the added noise and the signal have the same wavelengths.

To better explain the importance of this distinction, consider this from the point of view of the demodulator. The demodulator is expecting a signal with wavelength λ_1. Inter-channel crosstalk happens when the demodulator receives the correct signal with λ_1 and other, incorrect, signals with wavelengths $\lambda_i : i \neq 1$. Intra-channel crosstalk happens when the correct and incorrect signals all arrive with λ_1. Because of this, intra-channel crosstalk has worse impact on the Signal-to-Noise Ratio (SNR) of the signal: its effects cannot be alleviated with filtering before reception [28, 1].

Most elements in ONoCs are a source of crosstalk. MRRs, for example, are not perfect filters. If a MRR is configured to route a signal around it, part of the optical power of that signal will instead leak and continue on the same waveguide, thus worsening the SNR of other optical signals traveling on the same path. The same happens when a MRR is configured to not route a signal: part of its optical power will be routed and add to the noise of another waveguide – Figure 3.1(b). Waveguide crossings also add crosstalk, because part of a signal going through the crossing will leak in the two perpendicular directions – Figure 3.1(c).

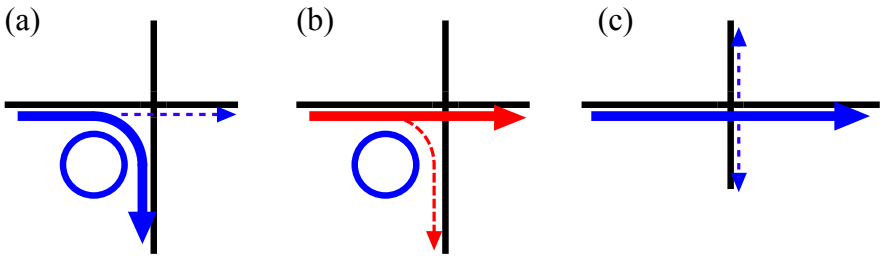


Figure 3.1: Crosstalk sources in ONoCs. (a), (b) Crosstalk due to MRRs. (c) Crosstalk due to waveguide crossings. Color indicates wavelength.

Crosstalk analysis is a crucial part of the development of an ONoC because it can limit network scalability, reliability and performance. Many publications have tackled this issue for active ONoCs [29, 28, 1], but a deep analysis specifically for WRONoCs is still missing.

3.1.3 Bit-rate and bit parallelism

Increased bandwidth is one of the major selling points for ONoCs. In the case of active networks, high bandwidth is achieved using wavelength-division multiplexing: multiple different wavelengths are sent through the same path between the sending and receiving nodes at the same time [1]. This is conceptually equivalent to a digital parallel bus sending one bit through each wire. With WRONoCs, wavelength is used for routing, so having multiple wavelengths follow the same path to increase bit parallelism is possible, but restricted.

As explained in Section 2.2.2, MRRs have a set of resonance frequencies. Because of this, it becomes possible to have a Wavelength-Routed network which allows multiple different wavelengths to follow the same path: the wavelengths following the same path must be in the set of wavelengths of the MRRs used to form that path. To have different paths in the network, the different MRRs must be configured so that their corresponding sets of wavelengths do not coincide. This might not always be possible due to uncertainties in the MRR manufacture or, most commonly, if too many different MRRs are required in the router. Only a careful selection of the radius of each MRR and the wavelengths used in the network can assure bit parallelism in WRONoCs [14].

Even if this performance factor is not directly considered when designing the WRONoC at a higher level of abstraction, one can still prepare for it by minimizing the number of unique MRRs and wavelengths used in the router.

3.1.4 Power usage & optical insertion loss

Power usage for NoCs in general can be divided into two slices: static power consumption and dynamic power consumption. Static power consumption is the amount of energy per second the NoC requires when no data is being transmitted. Conversely, dynamic power consumption is related to the amount of energy needed to transmit one unit of information (for example, bit, flit, etc). On the one hand, WRONoCs require no control layer because all routing is passive, and so all dynamic power usage comes from the optical-electrical network interfaces (which are unrelated to the ONoC technology itself). On the other hand, WRONoCs require always-on laser sources, which directly impact the static power consumption[1].

For information to be transmitted an optical signal must travel from the laser source to the modulator on the transmitter node through the OPDN, and then from the modulator to the demodulator on the receiver node through the actual WRONoC router. This optical path between the laser source and the demodulator is affected by various types of losses, which are highly dependant on the design of the OPDN and the router. To achieve transmission reliability these losses must be compensated by increasing the power of the laser source such that the necessary amount of laser power arrives at the photodetector. The sum of all losses over an optical path is called the optical insertion loss. The required laser power of a laser source therefore increases monotonically with the maximum optical insertion loss over all optical paths powered by that laser source.

[1] MRR heaters are also another important source of static power consumption.

Optical insertion loss contains the components below [1, 30, 3].

Crossing loss when the signal passes through a waveguide crossing – Figure 3.2(a). This happens regardless of whether the perpendicular waveguide has signals going through it or not.

Dropping loss when the signal routes through a MRR – Figure 3.2(b).

Through loss when the signal passes through a waveguide close to a MRR of a different wavelength – Figure 3.2(c).

Bending loss when the signal passes through a bend in a waveguide – Figure 3.2(d).

Propagation loss due to light scattering along the waveguide. This value is proportional to the length of the waveguide path the signal travels.

Modulator and demodulator losses upon modulation at the sending node and detection at the receiving node.

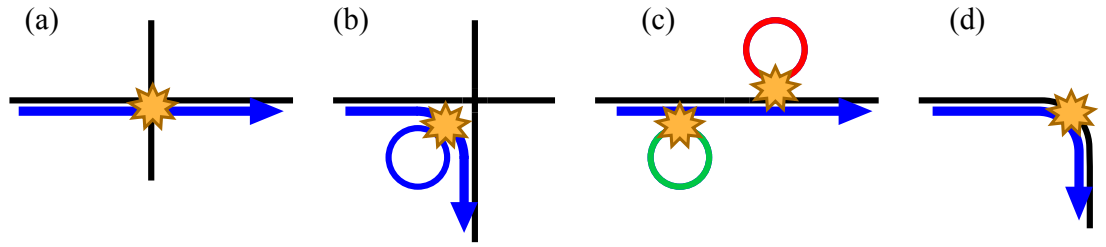


Figure 3.2: Some optical insertion loss sources in ONoCs. (a) Crossing loss. (b) Dropping loss. (c) Through loss. (d) Bending loss. Color indicates wavelength.

Here all types of losses are considered except for the last two (modulator and demodulator losses), because these are constant and equal for all optical paths and thus can be ignored from an optimization perspective. For the others, Table 3.1 presents example[2] values as given by Nikdast *et al.* [1].

Table 3.1: Optical loss values published by Nikdast *et al.* [1].

Type	Value
Crossing loss	0.04 dB
Dropping loss	0.5 dB
Through loss	0.005 dB
Bending loss	0.005 dB/90°
Propagation loss	0.274 dB/cm

[2]These values are bound to change and improve with the development of sillicon-photonic technology.

3.1.5 Wavelength usage

The required number of wavelengths is an important factor in WRONoC design. Fabrication uncertainties and technological restrictions impose a ceiling on the number of available wavelengths. For example, due to the limited frequency range available for transmission on waveguides, a higher number of wavelengths means a smaller distance (in frequency) between each wavelength, which in turn requires tighter fabrication tolerances for MRRs [14]. A limit of 62 wavelengths for transmission on a single waveguide has been reported [31]. Others indicate this maximum to be only 16 [32]. Whatever the case may be, it is clear that minimizing wavelength usage leads to easier fabrication, higher bit parallelism and lower power consumption.

3.2 Design flow of WRONoCs

The design of the optical routing layer of a WRONoC starts with the following data:

- Communication matrix given by the communication requirements of the SoC. The communication matrix is a square binary matrix $CM_{i,j} \in \mathbb{R}^{N \times N}$ with N equal to the number of nodes of the network and where $CM_{i,j} = 1$ if node i needs to send information ("messages") to node j.
- Positions of the modulators and demodulators of each node defined by the location of each SoC component on the electrical layers.

Given that information, the design flow of WRONoCs must include these four major steps:

- Design of the logical topology of the WRONoC router.
- Physical layout of the elements of the router (Place & Route – P&R).
- Design and layout of the optical power distribution network.
- Assignment of physical parameters to the router.

The design of the logical topology concerns itself with creating routers such as the λ-router or GWOR as exemplified in Section 2.5.3 so that the communication matrix is fulfilled. This means connecting the modulators to the demodulators with waveguides and MRRs, and selecting the (symbolic, abstract) wavelengths of the MRRs and messages according to the WRONoC design rules given in Section 2.5.1. If non-complete communication matrices are a possible input (and in general, they very clearly are [33]), then this step presents an optimization opportunity[3]. As shown before, many different logical topologies have been presented but very few have so far attempted to optimize them for non-complete communication matrices [32].

The physical layout of the WRONoC router is about taking the elements used by the logical topology, i.e., waveguides and MRRs, and optimally placing and routing them on the optical

[3]For example, if the network has 10 nodes but only sends 5 messages, then instead of 10 wavelengths, maybe a minimum of only 5 or less is possible. The same optimization opportunities exist for the number of MRRs.

routing plane according to the physical positions of the nodes on that plane. This placement is directly constrained by many of the performance factors listed before, such as power usage and temperature resilience. Some tools have been developed directly for this step, i.e., they take as input the logical topology and place it on the optical plane [34, 3, 35]. Only one specifically considers temperature resilience [24]. Proton+ [3] is considered the state of the art for WRONoC physical layout.

The design of the OPDN deals with selecting the best laser power splitting tree that distributes the correct fraction of the total laser power to each modulator. The layout of the OPDN is about determining where to place the splitters and how to route the waveguides to connect the laser sources to the modulators. Both of these steps have a direct impact on the power usage. The OPDN, for example, uses waveguides to transmit laser power, so its layout may generate extra crossings with the waveguides of the router and thus increase insertion loss[4]. Some work has been done in this area [22], but mostly for specific logical topologies and physical layouts [17, 36].

Finally, it must be observed that the design of the logical topology does not concern itself with assigning real wavelength values to each message, nor actual radius values to MRRs. All examples given in Section 2.5.3 use symbolic placeholders such as λ_1 or λ_2 for both messages and MRRs. These, however, must be assigned actual real values before a WRONoC is built. This assignment of physical parameters is a step at a lower level of abstraction that nonetheless has a direct impact on the bit parallelism of the router and may sometimes invalidate logical topologies if too many different MRRs are used for the given fabrication technology to handle. This has been studied previously [14], but actually implemented only as a validation step after the other steps have been carried out.

3.3 Sequential vs combined optimization

In an ideal world an algorithm would be known that, when given the communication matrix and the node positions, would perform all four steps simultaneously whilst considering all the performance factors, thus reaching the absolute optimal solution.

For lack of an ideal world, the perfect algorithm has been thus far approximated by removing performance factors from the equation and/or, most importantly, by considering these steps in sequence. In other words, a logical design is first created, then this design is placed and routed, the OPDN is then placed and finally the topology is further refined by assigning physical parameters[5].

The main reason for the ineffectiveness of this simplification of sequentially optimizing each step is that many steps have interdependencies. For example, the layout of the router and the OPDN should not be done separately, because either can clash with the other and form extra sources of insertion loss. Therefore, if one step is optimized first then fixed in place, the optimization of the other step is done only around a local optimum of the combined optimization space.

[4] This can, in turn, trigger the need for a redesign of the power splitting tree.

[5] Not *all* published works fall into this *exact* sequence, of course, but the overall argument still stands.

This constrained optimization space is not guaranteed to include the global optimum, hence the reduction in solution quality.

Given these four steps, no interdependency has been more studied or is clearer than the one between logical topology design and physical layout optimization. Published work analyzing these two steps has unanimously pointed out that there is a big difference between designing a logical topology and performing the P&R step on it [5, 21, 3]. For example, if a certain logical topology has a maximum number of crossings on any optical path of 7, good physical layouts of that topology are likely to increase that number to anywhere between 27 and 64 [3], therefore drastically increasing the laser power requirements. This is a consequence of the logical topology not including information on the network node positions. Instead, it makes strong assumptions about them which are almost always wrong. For example, all topologies in Figure 2.11 except for GWOR do *not* have the modulator and demodulator pairs (M_1, D_1), (M_2, D_2), etc, next to each other, whereas a real ONoC will almost certainly have each pair on the same physical location, i.e., the node location. The discrepancy in the number of crossings increases with the number of elements of the logical topology to be placed, thus raising even bigger concerns the more nodes the network has.

In short, the fact that it is very difficult to accurately predict physical parameters (like the number of crossings or message path length) when optimizing only the logical layout (and so before optimizing the physical layout) is what drives the interdependency between logical design and physical layout and thus constitutes another big reason against the sequential solving of this problem.

Yet, no method has been put forth so far to combine the optimization of these two steps. This is exactly the gap to be addressed here and, in doing so, validate or disprove that it is actually possible to get better results by combining the optimization of the two steps compared to optimizing each step sequentially.

In this thesis some simplifications are still made (the world is not yet perfect):

- Step simplifications
  - The OPDN is not optimized, but the implemented approach will provide ways to break the interdependency between the OPDN and the physical layout of the router in many cases. This way, when the OPDN is then later added, no additional power costs will suddenly appear (see Section 6.2.2).
  - Like all other logical topology optimization efforts, the assignment of physical parameters is still not considered and is still left as a further validation/refinement step.
- Performance factor simplifications
  - Crosstalk is not considered, following other optimization efforts for WRONoCs so far. However, the implemented approach allows for the minimization of MRRs and for manually controlling the number of crossings, which indirectly minimizes crosstalk.

- Temperature resilience is not directly considered in the optimization process but the implemented approach will allow this factor to be considered through manual input (see Section 6.2.3).
- Bit parallelism is not considered directly. Because of this, MRRs are abstracted to have only *one* resonance frequency, like in all logical topology designs presented thus far. However, the implemented approach allows for the minimization of MRRs, which indirectly helps bit parallelism.

3.4 Formal definition of the optimization problem

Given the problem statement and the simplifications presented in the last section, the formal definition of the optimization problem for the design of WRONoC routers tackled in this research is as follows:

Input data

- Communication matrix: a square binary matrix $CM_{i,j} \in \mathbb{R}^{N \times N}$ with N equal to the number of nodes and where $CM_{i,j} = 1$ if node i sends a message to node j.
- Physical positions of the modulators and demodulators of each node on the optical plane.
- Technology parameters: power loss values.

Output data

- Wavelength (symbolic) of each message and MRR.
- Placement of each MRR.
- Routing of each waveguide.

Minimization objectives [5, 14, 20, 17, 3, 21]

- Number of wavelengths.
- Message insertion loss.
- Number of MRRs.

The weight given to each objective should be freely controlled by the designer in order to reflect the varying importance of the performance factors. Also, the exact function used for the message insertion loss (maximum insertion loss over all messages, sum of the insertion losses over all messages, or others) depends on the type of laser[6] and other factors, so should be kept as adaptable as possible (or at least a few different cases should be considered).

[6] For example, for comb lasers, this function is the maximum over all messages but, for single-wavelength laser arrays, the function that makes the most sense is the sum of the maximum over each wavelength.

Chapter 4

Methodological approach

The optimization problem for WRONoCs studied in this work was introduced in the previous chapter. This chapter presents the implemented approach. First, the underlying rationale is explained. Then the approach is defined in detail. The final section offers some additional comments.

4.1 Physical layout template

Ideally, a design tool would take as inputs the communication matrix and the physical positions of the nodes, as mentioned in the last chapter, and, by optimizing both the logical topology and physical layout simultaneously, produce a fully-optimized fully-custom solution. It is clear, however, that even with the simplifications already made, an optimization problem like this one is very complex, combining both synthesis and optimization requirements, which results in a solution space that is discouragingly vast for any but the simplest cases. Published work has so far solved this optimization problem by reducing its complexity considerably: first a logic topology is chosen, then the elements of that topology are placed on the optical plane with the use of a P&R tool. Here the optimization problem is tackled differently. The chosen approach also constrains the optimization problem from Section 3.4 but does it in an insightful way, such that:

1. The developed algorithms are still given enough flexibility to design the logical topology and the physical layout together, letting any restrictions, choices or optimization opportunities from one aspect influence the choices made on the other.

2. Whatever constraints are placed to make the problem more manageable can be directly designed in accordance to other, possibly informal, sources of information, such as heuristics or designer experience (or intuition).

3. It is possible to gradually loosen those constraints as better algorithms are developed (or more CPU power is made available), in effect allowing the constrained solution space to incrementally and controllably approach the solution space of the complete problem.

Characteristic one is required given the problems that arise from separating the two aspects. Characteristics two and three are not required but are nonetheless highly desirable. Characteristic

two allows other sources of information to easily instruct the optimization algorithms in such a way that the constrained solution space is more likely to include some or all of the best solutions from the complete solution space. Characteristic three is beneficial because it "future-proofs" the optimization procedure. Researchers can focus on improving the optimization algorithms such that more of the solution space can be searched, rather than re-writing the problem statement entirely and potentially starting from scratch.

The chosen approach, which has all of these characteristics as will be explained later, is to consider a new input to the optimization process: a **physical layout *template***. This input consists of a collection of WRONoC router elements (modulators, demodulators, waveguides and MRR placeholders – see Section 2.2) already placed and routed on the optical plane, to which the solution must conform.

This new input constrains the problem because the synthesis from scratch of a physical layout is turned into an optimization of the given template. In other words, to design the physical layout of the solution, the algorithm is now required to decide only on which elements (waveguides and MRR placeholders) to keep and which to remove from the template. Thus, most importantly, it will never be asked to place any new elements in new locations.

A more detailed explanation of what a physical layout template is, along with examples, is given next. Then, a brief word is said about the other major input to the WRONoC design problem, the communication matrix. Some concluding remarks close the chapter.

4.2 Template elements

Physical layout templates are composed of multiple instances of three basic elements, each having a fixed location on the optical routing layer. These elements closely match the four WRONoC components that are placed on that layer (see Section 2.2):

Endpoints represent modulators and demodulators. They are placed wherever the modulators and demodulators for each node are and connect to one waveguide section.

General Routing Units (GRUs) are elements that connect through *ports* to multiple waveguide sections, called the *edges* of the GRU, and contain MRR placeholders to be populated by the algorithm as needed. They are the only template element that can contain MRRs, making them the routing building blocks of the template. They are described further in Section 4.2.1.

Waveguide sections connect two GRUs or a GRU and an endpoint. They are described further in Section 4.2.2.

An example of how these elements are interconnected to form a layout template is shown in Figure 4.1. Here the template generalizes the 4x4 GWOR topology [20], which uses 4 PSEs to connect 4 nodes. In both cases each node is given one modulator and one demodulator which are placed on the same side of the router.

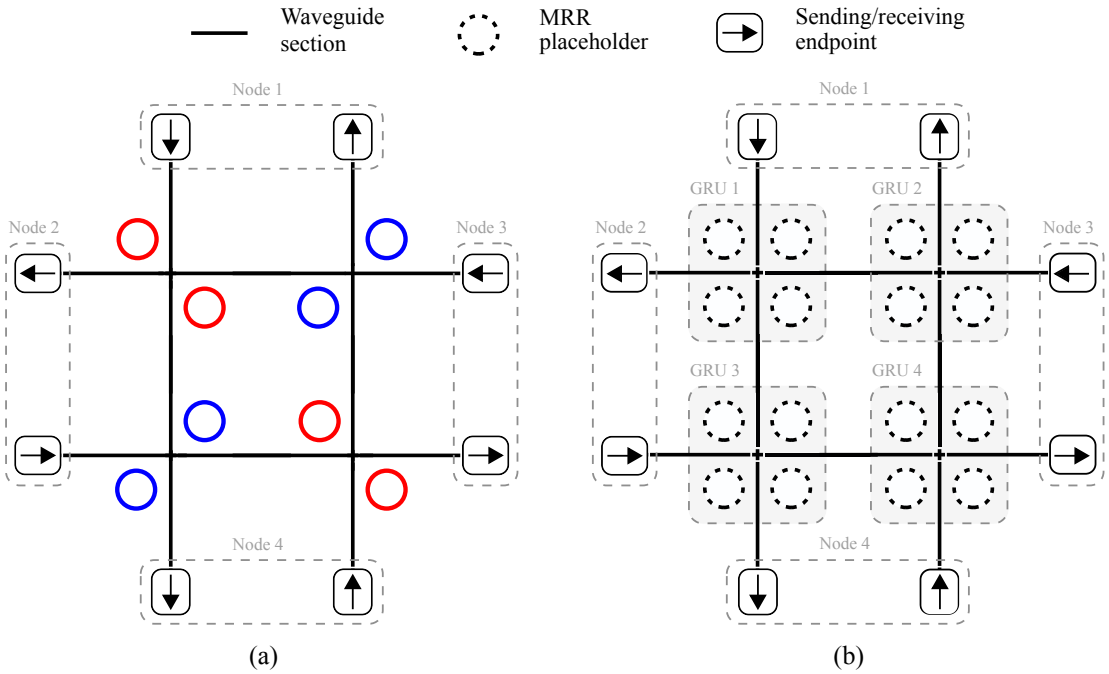


Figure 4.1: Example of a physical layout template. (a) The 4x4 GWOR topology [20] (color indicates wavelength). (b) The generalization using endpoints, GRUs and waveguide sections.

4.2.1 General Routing Unit

PSEs are commonly applied in WRONoC routers [2, 20, 21, 5, 3, 24, 35]. Yet, PSEs have some distinct shortcomings:

- They only have one or two MRRs, where in fact it is possible to place up to four MRRs on a single crossing (one on each corner).
- Both MRRs always have the same resonance frequency, where in fact all *four* MRRs on a crossing can have different resonance frequencies.
- Their waveguide structure is fixed – PSEs always have a crossing – where in fact other routing designs are also possible.

To solve this inflexibility a new type of optical switch is proposed: the ***General* Routing Unit (GRU)**. Externally, GRUs still have four ports to which waveguides are connected to, like PSEs. However, in contrast to PSEs, the internal structure of GRUs is not inherently constrained to a specific configuration, as shown in Figure 4.2. Internally, only MRR placeholders are predefined, which can be populated with MRRs of independent resonance frequencies based on problem needs. Additionally, different instances of GRUs can have different connection arrangements between the four ports. This provides more flexibility in the resulting WRONoC design.

To start with, a specific set of GRU configurations is presented which generalize the simple structure of a PSE. But GRUs are designed to be forward-thinking: in the future, more GRU designs can be researched, implemented and analyzed.

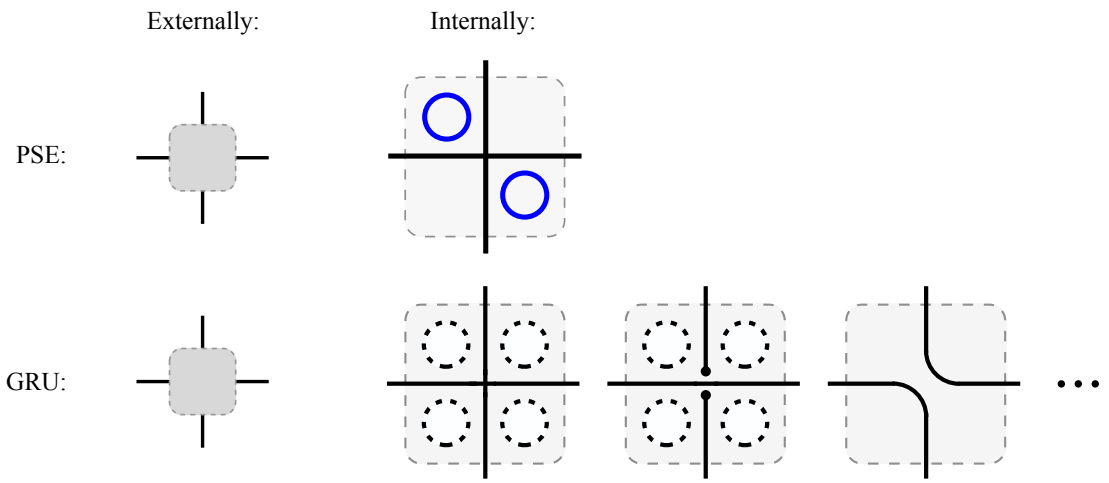


Figure 4.2: External and internal comparison between PSEs and GRUs.

4.2.1.1 Structure

The most basic version of a GRU is based on the PSE: the four waveguide sections come together at the center to form a crossing, as shown in Figure 4.3(a). In this case, any of the four corners of the crossing can have one MRR.

As explained in Section 3.1.4, crossing loss happens when a message goes through a crossing with a perpendicular waveguide. This happens regardless of whether messages are going through the perpendicular waveguide or not. Therefore, avoiding the center crossing in GRUs, like in Figure 4.3(b), is advantageous and is possible when messages only go through the center of the GRU horizontally or vertically.

A third structure variation is considered, called *corner bending*. When active, the GRU contains no MRRs and some corners may be replaced by a bend between the two edges in that corner. Valid examples are shown in Figure 4.3(c). However, not all corners can be bent at the same time on the same GRU. For example, two corners on the same side cannot be both bent. Also, when one or more corners are bent, opposite waveguide sections of the GRU cannot be connected. Invalid examples are shown in Figure 4.3(d).

Corner bending effectively fuses the two waveguides forming the corner, meaning all messages route through the corner regardless of wavelength. This is in opposition to the use of MRRs, which only route one message each. The trade off is that no MRRs can be used in the GRU. Hence, this variation proves useful for sparser templates (low number of messages to number of MRR placeholders ratio) or in cases where multiple messages must be routed through the same corner.

4.2.1.2 Routing

Given the GRU structure variations presented above, a message with wavelength λ can route through a GRU in four different ways:

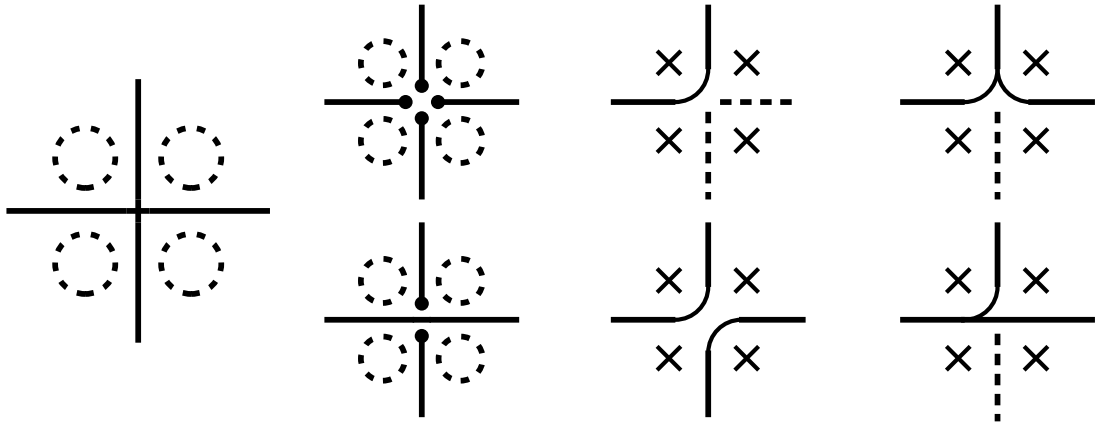


Figure 4.3: Internal structure of a GRU. (a) 4 MRR placeholders and a crossing. (b) Avoiding the crossing when possible. (c) Valid corner bending states. (d) Invalid corner bending states.

- Through the center (direct path) – Figure 4.4(a). In this case MRRs may be present, but their wavelength must differ from λ.
- Through a corner using the MRR on that corner – Figure 4.4(b). In this case a MRR of wavelength λ must be on that corner. The adjacent corners cannot have MRRs of wavelength λ, but the opposite corner may also have an MRR of wavelength λ (just like a PSE).
- Through a corner using the MRR on the opposite corner – Figure 4.4(c). In this case, no other MRR is allowed to have the wavelength λ.
- Through a corner using corner bending – Figure 4.4(d). No MRRs of any wavelength may be present.

One important characteristic of these rules is that the path a message takes through a GRU is always independent of its direction, i.e., all routing features are bidirectional.

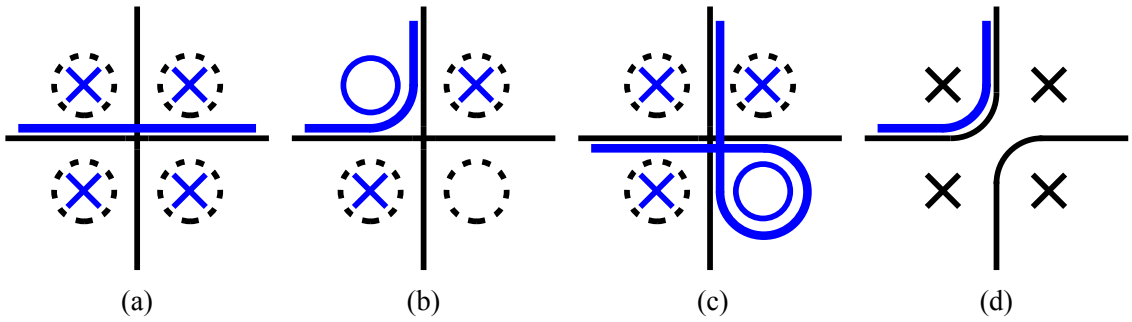


Figure 4.4: Routing possibilities on a GRU. (a) Direct path. (b), (c) Routing through a MRR. (d) Routing through a bend.

4.2.2 Waveguide section

Waveguide sections are not complete waveguides, in the sense that they do not begin and end in modulators, demodulators or waveguide terminators. Instead, multiple waveguide sections are strung together through GRUs to form complete waveguides. This allows the algorithm to connect sections in different ways and to remove sections which are not being used (which have no messages going through). It also allows the template itself to be more flexible and detailed: each waveguide section is assigned a *length* and an *extraloss* value as part of the template construction process. The *extraloss* value can be changed to account for losses other than propagation loss in the waveguide, such as bending loss, as shown in Figure 4.5.

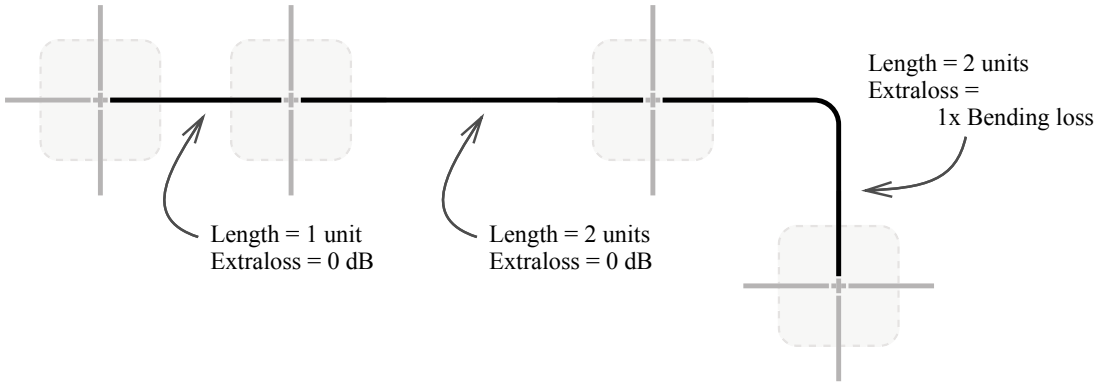


Figure 4.5: Example of how to determine waveguide section parameters.

4.3 Communication matrix

With the layout template, each WRONoC node is now defined by its sending and receiving endpoints. Therefore, the other major input to the WRONoC optimization, the communication matrix, can be translated into a set of messages where each message is an ordered pair of endpoints (E_S, E_R): E_S is the sending endpoint and E_R is the receiving endpoint. One message is added to the set for each nonzero entry in the matrix. Figure 4.6 shows an example of how the conversion is made for a network with three nodes.

4.4 Concluding remarks

The chosen approach is to transform the two major inputs to the WRONoC optimization problem into **i)** a physical layout template and **ii)** an accompanying set of messages defined from the communication matrix and the template. Figure 4.7 shows an example of a complete set of inputs for the constrained WRONoC optimization problem after this transformation.

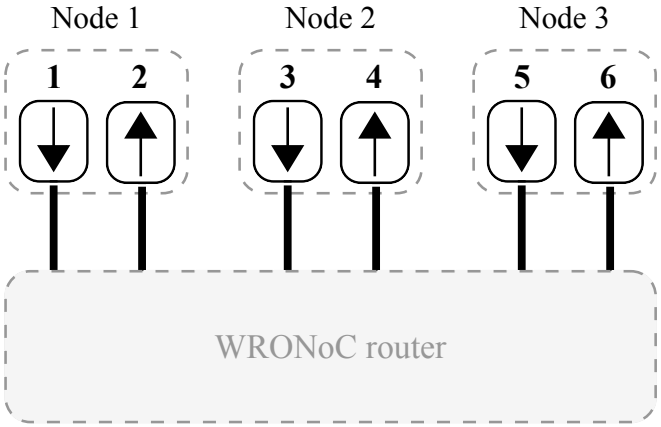
This approach brings about several advantages:

- Node positions are automatically considered in the template through the positions of the endpoints.
- Since the template is fixed and no more elements are to be added, not only is the synthesis problem turned into an optimization problem as stated before, but it becomes possible for the algorithm to calculate with certainty layout-dependant parameters before and during the optimization (such as the number of crossings or the amount of propagation loss of each path).
- It allows for the inclusion of more powerful routing primitives such as GRUs, whose flexibility can only be taken advantage of if both the logic topology and the physical layout are being optimized simultaneously.

The design of the physical layout template itself has not yet been mentioned. In general, the template can be created manually by the WRONoC designer or it can be generated by some kind of synthesis tool. However, results show very clearly that the template does not need to be intricate or sophisticated, i.e., it can easily be created manually. In other words, the intuitive knowledge of the designer about the structure of the router to create is more than enough to provide a good template.

Lastly, not only does this approach have the more specific advantages outlined above, it also fulfills the three general goals for constraining the optimization problem as stated at the beginning of this chapter:

1. The algorithm has the flexibility to simultaneously optimize layout aspects, such as the physical path of the messages, and logical aspects, such as the wavelengths of the messages and the MRRs.
2. The physical layout template given as input can be directly influenced by the experience of the designer. It also allows for template synthesis tools to be developed in the future.
3. By controlling the size of the template, i.e., mainly the number of GRUs and waveguide sections, the size of the constrained solution space can be managed. By solving for larger templates, more of the complete solution space is being looked at. Ultimately, big enough templates allow the constrained space to approximate the complete space with negligible error.



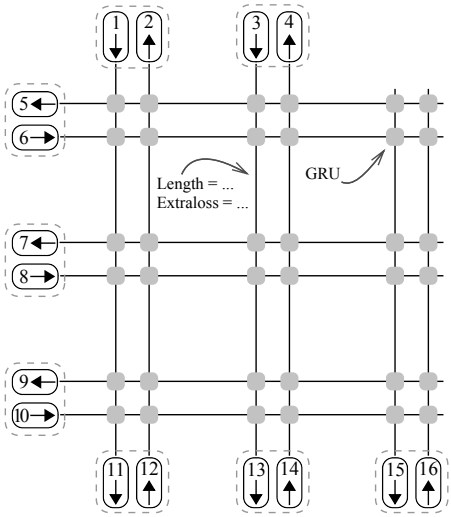
$$CM_{i,j} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \end{bmatrix}$$

$$\Downarrow$$

$$(E_S, E_R) = \{(1,4), (3,4), (3,6), (5,2)\}$$

Figure 4.6: Converting a communication matrix into a set of pairs of endpoints given a physical layout template.

i) Physical layout template:



ii) Message list:

$$(E_S, E_R) = \{(1,4), (6,2), (6,4), (8,9), (12,2), (12,4), (16,9), \ldots\}$$

iii) Loss parameters:

Crossing = 0.04 dB
Dropping = 0.5 dB
Through = 0.005 dB
Bending = 0.005 dB/90°
Propagation = 0.274 dB/cm

Figure 4.7: Example of a complete set of inputs for the constrained WRONoC optimization problem.

Chapter 5

Optimization algorithm

Having defined the constrained optimization problem in Chapter 4, this chapter now focuses on the design of one possible algorithm to solve it. To start with, the requirements of any algorithm for this problem are laid out. Then, a short overview of combinatorial optimization algorithms, including Mixed Integer Programming, is given. Next, the WRONoC model used to solve this problem is described in full. Finally, some accompanying techniques to the core of the algorithm are presented.

5.1 Algorithm requirements

Any optimization algorithm for this problem has to specifically fulfill the following major tasks:

Route messages. The algorithm is required to give a path for each message through the template that starts and ends at the correct endpoints.

Assign wavelengths. The algorithm must assign a wavelength to each message such that no interference between messages exists.

Activate routing features. The algorithm has to configure GRUs according to the chosen message paths and wavelengths.

This is in essence a combinatorial problem. However, these tasks actually have a deep interdependence, i.e., the decisions made on one task influence the possible outcomes of the others, to the point where some decisions made on one task might make the solution unfeasible through restrictions imposed by other tasks.

As an example, consider a simple greedy algorithm that builds a feasible solution by choosing a path, a wavelength and activating the routing features for each message in sequence. Such an algorithm would consistently find itself stuck with no feasible solution. For example it is very likely that, after the algorithm performs the three steps for the first 10 messages, the 11th message no longer has the necessary GRU routing features available to reach the correct endpoint. Because of this such an algorithm would be constantly backtracking, which is extremely inefficient[1] unless

[1] In the worst case, this is no better than a brute-force approach.

carefully controlled. This shows how the difficulty of this particular problem goes beyond the actual optimization: unlike many other combinatorial problems, here the construction of the first feasible solution alone is already complicated.

5.2 Combinatorial optimization

A whole range of algorithms and methods exist to efficiently and effectively solve combinatorial problems, all of which make a trade-off between solve time and the quality of the solution. On the one hand, using Linear Programming (LP) will give optimal solutions in a finite amount of time. On the other hand, if LP is too slow or if the problem is impossible to model linearly, other methods such as local search (and variations thereof) and genetic algorithms may produce "good-enough" feasible, albeit not guaranteed to be optimal, solutions faster.

Here, a short overview of some of those methods is given. Linear Programming was chosen for this particular problem and is thus explained in more detail later in Section 5.3.

5.2.1 Local search and metaheuristics

A basic local search algorithm is comprised of three steps:

1. An initial feasible solution is generated (commonly with the use of a greedy algorithm).
2. A neighborhood of feasible solutions around the current best solution is constructed and explored.
3. The current best solution is replaced with a better solution from that neighborhood.

Steps two and three are repeated until a satisfactory solution is found or the optimization "time budget" is spent.

This procedure assumes that an initial feasible solution can be found easily and also that an adequate neighborhood of feasible solutions can be constructed and explored quickly. In its simplest form, each iteration updates the current solution with the best solution from the neighborhood. This leads to a so-called "hill-climb" around the local optimum closest to the starting solution. The problem with this simple approach is that very frequently only one local optimum of the solution space is explored. When reaching this optimum the algorithm will most likely get stuck and no longer improve the solution[2].

To solve this issue multiple variations around the core optimization principle of local search exist and are briefly presented below. The main goal of these metaheuristics is to introduce variety in the local search procedure such that the solver does not get stuck, thus exploring more of the solution space while using as little time as possible.

Iterated local search resets and repeats the local search algorithm with a new, different, starting solution whenever the entire neighborhood is worse than the current best solution.

[2]This is unless the the considered neighborhood is big enough to "see far and over the hill".

Simulated annealing chooses a random solution from the neighborhood at each iteration. If it improves upon the current solution, it replaces the current solution. If not, it may still replace the current solution with probability p, where p is gradually lowered towards zero as the optimization runs.

Tabu search makes two changes to the core principle. Firstly, the search can move to a worse solution if no superior solution is available in the neighborhood. Secondly, a list of previously explored solution "moves" is recorded to avoid the algorithm retreading its own steps.

5.2.2 Genetic algorithms

Genetic algorithms work by mimicking the process of evolution of living beings by natural selection. A simple explanation of the procedure is given next.

1. A starting pool of feasible solutions is generated.
2. A selection of some solutions from the pool is made (commonly, the selected subset mostly contains the best solutions).
3. The selected solutions go through a breeding process in which multiple random pairs of solutions are combined to create new solutions that are selectively accepted to form the new solution pool

Steps two and three repeat with the same stopping condition as local search.

Ideally, the breeding procedure should be designed such that child solutions keep the most important characteristics from their parents. After breeding, invalid child solutions may be thrown out or made feasible through small modifications when possible. Also, other entirely different solutions may be generated through mutation procedures on currently available solutions to keep the solution pool from stagnating.

5.2.3 The choice for Linear Programming

As explained before, LP is the perfect method for solving linear combinatorial problems except for when it is too slow for the application context where the problem must be solved. The main advantage of other algorithms lies precisely in forgoing guarantees of optimality in favor of optimization speed. Even so, as advisable as these are in many cases, the choice was made for linear programming because it was not at all certain *a priori* if LP modelling would be too slow for this particular problem. When in doubt, starting with LP modelling is a good option:

- If LP is fast enough, then there is no need for another algorithm since it already provides optimal solutions.
- If LP is too slow for some problem sizes, it can still serve as a benchmark for speed and solution quality should other optimization algorithms be developed.

Furthermore, with LP modelling the difficulty in generating feasible solutions for this problem (which are crucial for the algorithms presented above, for example) is bypassed. Finally, there exist methods for adding metaheuristics such as tabu search to LP [37, 38], which once again make LP a good starting point.

5.3 Linear and Integer Programming

Linear Programming is a method whereby a linear function is optimized (minimized or maximized) within the constraints imposed by a linear mathematical model. All LP problems are described using real variables, linear constraints on those variables and a linear optimization function of those variables. The canonical form of LP problems is:

$$\text{max or min} \quad C^T \cdot \boldsymbol{x} \tag{5.1}$$

$$\text{subject to} \quad A \cdot \boldsymbol{x} \leqslant B \tag{5.2}$$

$$\boldsymbol{x} \in \mathbb{R}^N \tag{5.3}$$

where $\boldsymbol{x}$ is a column vector of the decision variables, B and C are column vectors of constants and A is a square matrix of constants.

In the particular problem for WRONoCs there is a need for binary variables. These are *integer* variables whose values are only allowed to be 0 or 1. A generalized version of LP problems exists, where some variables are forced to have integer values: Mixed Integer Programming (MIP) problems. Their canonical form is equal to LP problems except for Equation 5.3, which is replaced by:

$$x_i \in \mathbb{R} \quad \forall i \in X_R \tag{5.4}$$

$$x_j \in \mathbb{Z} \quad \forall j \in X_Z \tag{5.5}$$

With MIP models the effort of solving an optimization problem is shifted from designing a specialized algorithm for solving the problem to describing the problem through variables and constraints. Then, one of the already existing solvers can be used to solve the model – and thus the problem.

MIP solvers such as Gurobi [39] and CPLEX [40] are very advanced and use a plethora of solving algorithms, helping heuristics, cutting-plane methods and are built to take advantage of the parallelism in modern CPUs. Because of this, these solvers can actually find good results fast similarly to other algorithms like the ones presented in Section 5.2 even when faced with complicated combinatorial problems. Some MIP solvers allow the user to provide a feasible solution to *warm-start* the optimization process. In many cases, this can substantially reduce solve times. More importantly, some solvers even allow for the given starting solution to be incomplete (in that case, taking more the role of a hint on how to build a feasible solution). This capability will prove itself advantageous when optimizing the WRONoC model.

Besides giving access to years of expert algorithm design, MIP models also offer other valuable advantages in the context of the WRONoC design problem:

Optimality/error bound. A MIP model can give optimal solutions, or the optimization can be stopped mid-way such that a feasible solution is given along with a worst-case bound to how far that solution is from optimality.

Flexibility. The optimization function can be modified while keeping the constraints unchanged. The (nearly) same[3] MIP model can optimize entirely different objectives.

Adaptability. MIP models are generally easier to change than specialized algorithms. As such, new GRU designs, routing features or other modifications can easily be added.

The last two points are especially useful given that WRONoC research is quite young and still constantly evolving. Therefore, algorithms developed at this stage would do well not to bind themselves too much to specific technologies or practices.

The main disadvantage of MIP models is that they are NP-hard (except in special cases) so no guarantees on the time needed to solve them can be given. This can be alleviated by carefully designing the models (which can increase the effectiveness of solver heuristics, for example) or by not requiring the optimization to reach an optimal solution.

5.3.1 Modeling techniques

Oftentimes when describing a problem (and as is the case with the WRONoC model presented next) the required constraints do not naturally appear in the form of $\leqslant$ or $\geqslant$. They are more easily expressed in other formats which must then be translated to the canonical form. As such, a brief overview of the modeling techniques needed for the WRONoC model must be given first.

5.3.1.1 Equality

An equality constraint can easily be mapped into two inequality constraints:

$$A \cdot \boldsymbol{x} = B \qquad \mapsto \qquad \begin{cases} A \cdot \boldsymbol{x} & \leqslant B \\ A \cdot \boldsymbol{x} & \geqslant B \end{cases} \tag{5.6}$$

5.3.1.2 Maximum value

Minimizing the maximum value over a set of variables is very common in LP models. In those cases, the following mapping can be used:

$$x_m = \max_{i \in S} x_i \qquad \mapsto \qquad x_m \geqslant x_i \qquad \forall i \in S \tag{5.7}$$

This holds if the increase of x_m is penalized in the optimization function.

[3] In almost all cases, the complexity of changing the optimization function pales in comparison to the challenge of choosing the right variables and constraints to correctly model the problem. In essence, changing the optimization function does not really count as changing the model, except in mathematical terms.

5.3.1.3 Boolean expressions

The use of binary variables leads to the use of boolean expressions as constraints. The techniques used in the WRONoC model involving binary variables are presented next.

- Boolean negation[4]:

$$\neg\, x_i \quad \mapsto \quad (1 - x_i) \tag{5.8}$$

- Boolean *or*:

$$\bigvee_{i \in S} x_i \quad \mapsto \quad \sum_{i \in S} x_i \geqslant 1 \tag{5.9}$$

- Boolean *unique-xor*[5]:

$$\bigoplus_{i \in S} x_i \quad \mapsto \quad \sum_{i \in S} x_i = 1 \tag{5.10}$$

- Boolean implication:

$$\bigwedge_{i \in S_1} x_i \Rightarrow \bigvee_{i \in S_2} x_i \quad \mapsto \quad \sum_{i \in S_1} x_i - (|S_1| - 1) \leqslant \sum_{i \in S_2} x_i \tag{5.11}$$

$$\bigwedge_{i \in S_1} x_i \Rightarrow \bigwedge_{i \in S_2} x_i \quad \mapsto \quad \sum_{i \in S_1} x_i - (|S_1| - 1) \leqslant x_j \qquad \forall j \in S_2 \tag{5.12}$$

5.3.1.4 Activation of constraints

Binary variables can be used to turn on/off other constraints in the model. If x_b is a binary variable:

$$x_b \Rightarrow (A \cdot \boldsymbol{x} \leqslant B) \quad \mapsto \quad A \cdot \boldsymbol{x} \leqslant B + M * (1 - x_b) \tag{5.13}$$

$$x_b \Rightarrow (A \cdot \boldsymbol{x} \geqslant B) \quad \mapsto \quad A \cdot \boldsymbol{x} \geqslant B - M * (1 - x_b) \tag{5.14}$$

where M is a constant bigger than the maximum absolute value that the expression $A \cdot \boldsymbol{x} - B$ is allowed to have. In this case the constraints must be fulfilled if x_b is 1, but have no effect otherwise.

5.4 WRONoC model

Having given a brief introduction to and motivation for MIP models, the specific model used for the WRONoC design problem is now presented. First, the constants and indices are defined. Then, the constraints and optimization function are explained along with the relevant variables. Lastly, some model reduction techniques are outlined.

The constraints presented in this section are not in their final format, meaning the techniques listed in Section 5.3.1 still have to be applied. This is to make the purpose of the constraints clearer. The complete and final version of the model is given in Appendix A.

[4] This is not a constraint itself but a mapping that can be applied to the variable x_i in any constraint.

[5] The *unique-xor* of 2 or more variables is true if and only if exactly one of those variables is true.

5.4.1 Constants & indices

The optimization takes as input the physical layout template, the communication matrix and the loss parameters. This information is encoded in the model with various constants and indices, which are then used when formulating the constraints.

The constants are as follows:

- N_{gru}, N_{wg}, N_m, N_{ep}, N_λ - Total number of GRUs, waveguide sections, messages, endpoints and wavelengths respectively. Note that it only makes sense to consider $N_\lambda \leqslant N_m$, given that there will never be a need for more wavelengths than messages.
- L^P, L^C, L^B, L^D, L^T - Values for propagation, crossing, bending, drop and through loss respectively.
- L_{wg}, L^E_{wg} - Length and extra loss values for waveguide section wg.

Indices are used in the subscripts of variables and refer to elements in the layout template:

- W^T_g, W^B_g, W^L_g, W^R_g - Waveguide section connected to GRU g to the top, bottom, left and right respectively.
- W^E_{ep} - Waveguide section connected to endpoint ep.
- E^S_m, E^R_m - Sending and receiving endpoints for message m.

Constants L_{wg}, L^E_{wg} and indices W^*_i collectively describe the physical layout template. Indices E^*_m define the communication matrix.

5.4.2 Variables & constraints

5.4.2.1 Message routing

From a routing standpoint the physical layout template can be interpreted as a graph, where endpoints and GRUs are the nodes and the waveguide sections are the edges. The routing features in GRUs are bidirectional. Hence the direction of a message does not influence its path, thus making the graph undirected.

To describe the path of each message through the graph the set of binary variables $mwg_{m,wg}\ \forall m = 1...N_m, wg = 1...N_{wg}$ is created, where $mwg_{m,wg} = 1$ if the path of message m includes waveguide section wg.

To model the path of each message, three sets of constraints are needed as described below.

- The path must start and end at the correct endpoints. To guarantee this, each message must be present on the waveguide section (W^E_{ep}) connected to the sending (E^S_m) and receiving (E^R_m) endpoints:

$$mwg_{m,W^E_{E^S_m}} = 1 \quad mwg_{m,W^E_{E^R_m}} = 1 \qquad \forall m = 1...N_m \tag{5.15}$$

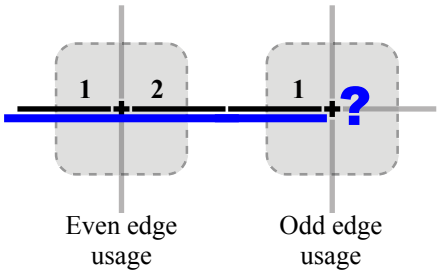


Figure 5.1: Even vs odd edge usage for GRUs.

- The path must be continuous from the sending endpoint to the receiving endpoint. Gaps in the path form when a message does not exit a node once for every time it enters it, as seen in Figure 5.1. Therefore, to remove gaps, constraints must be added to ensure that each message uses an even number of edges on each node.

  GRUs have 4 edges, so messages can use either 0, 2 or 4 edges on each. However, the choice was made to restrict those possibilities to only 0 and 2, for two reasons:

  1. **It simplifies the model.** The constraints for the activation of the routing features (see Section 5.4.2.3) and for the calculation of the insertion loss of each message (see Section 5.4.2.4) become much simpler if a message is restricted to use each GRU at most once (two edges) instead of twice (four edges).
  2. **It is very unlikely to appear in optimized solutions.** A path that uses a GRU twice can always[6] be simplified into a path that only uses it once, as shown in Figure 5.2. Also, a path that uses it twice has necessarily a bigger insertion loss (it has twice the loss on the GRU and the path must be longer, so it also has an increased propagation loss). Therefore, good solutions are unlikely to feature this case.

  The constraint is thus expressed as:

$$mwg_{m,W_g^T} + mwg_{m,W_g^R} + mwg_{m,W_g^B} + mwg_{m,W_g^L} \in \{0,2\} \qquad (5.16)$$
$$\forall m = 1...N_m$$
$$\forall g = 1...N_{gru}$$

[6]There are extreme edge cases that are exceptions to this "always". It is indeed true that any path, *considered in isolation*, can always be simplified in that way, and that it is always in the interest of the optimization function to do so. The edge cases happen when two or more paths interfere with each other. It is possible for two messages to want paths that use the same MRR, in which case one of the two must change its path to leave the MRR available for the other. One of the options is to use 4 edges on the GRU of that MRR instead of 2 (others are, for example, to use corner bending or to avoid using that GRU altogether).

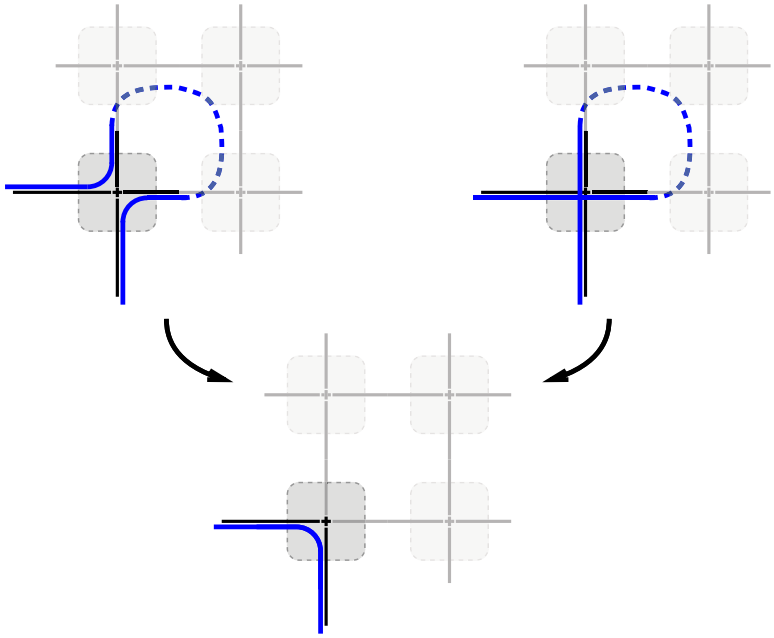


Figure 5.2: Path simplification from usage of 4 edges to 2 edges of a GRU.

The same constraint can also be written as:

$$
\begin{aligned}
& mwg_{m,W_g^T} \Rightarrow mwg_{m,W_g^B} \vee mwg_{m,W_g^L} \vee mwg_{m,W_g^R} \\
& mwg_{m,W_g^B} \Rightarrow mwg_{m,W_g^T} \vee mwg_{m,W_g^L} \vee mwg_{m,W_g^R} \\
& mwg_{m,W_g^L} \Rightarrow mwg_{m,W_g^T} \vee mwg_{m,W_g^B} \vee mwg_{m,W_g^R} \\
& mwg_{m,W_g^R} \Rightarrow mwg_{m,W_g^T} \vee mwg_{m,W_g^B} \vee mwg_{m,W_g^L} \\
& mwg_{m,W_g^T} + mwg_{m,W_g^B} + mwg_{m,W_g^L} + mwg_{m,W_g^R} \leqslant 2 \\
& \quad \forall m = 1 \ldots N_m \\
& \quad \forall g = 1 \ldots N_{gru}
\end{aligned}
\tag{5.17}
$$

This format promotes a different way of thinking about how correct routing paths are achieved. Constraints in 5.15 force some $mwg_{m,wg}$ variables to 1. Then, the constraints in 5.17 force, through boolean implications, some other $mwg_{m,wg}$ variables to be 1 too. This sequence of implications will, GRU by GRU, force the start and end portions of each path to meet.

- The path must not reach an incorrect endpoint. This set of constraints deter situations like the one depicted in Figure 5.3 from happening:

$$
\begin{aligned}
mwg_{m,W_{ep}^E} = 0 \qquad & \forall ep = 1 \ldots N_{ep} \setminus \{E_m^S, E_m^R\} \\
& \forall m = 1 \ldots N_m
\end{aligned}
\tag{5.18}
$$

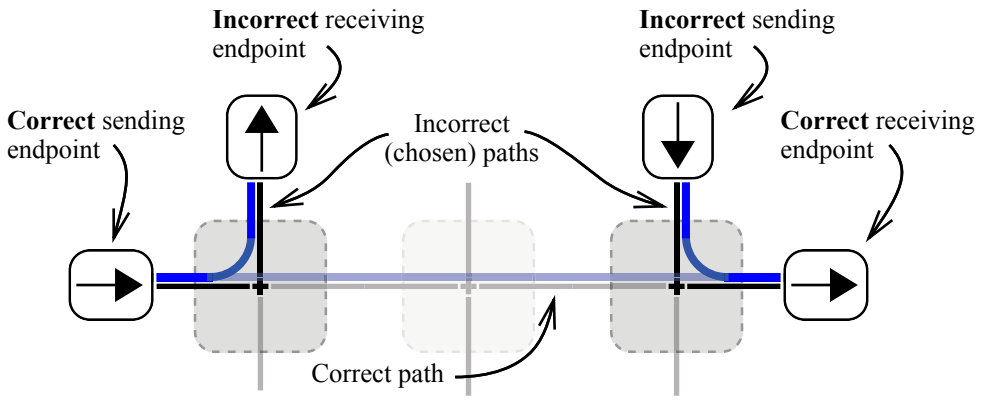


Figure 5.3: Possible incorrect results if message paths are allowed to use waveguide sections connected to endpoints which are neither sending nor receiving endpoints for the message.

5.4.2.2 Wavelength assignment

The first step in assigning wavelengths to messages is to create the set of binary variables $mwl_{m,\lambda}\ \forall m = 1...N_M, \lambda = 1...N_\lambda$ to indicate which wavelength each message is using, and to add the following set of constraints to force each message to use exactly one wavelength:

$$\sum_{\lambda=1}^{N_\lambda} mwl_{m,\lambda} = 1 \qquad \forall m = 1...N_m \tag{5.19}$$

Wavelength assignment to messages has a deep connection to the paths the messages are allowed to take. The model must ensure that no two messages share the same waveguide section *and* have the same wavelength. There are various ways to enforce this rule, four of which will be presented next.

Possibility 1. Use the following set of constraints:

$$\begin{aligned} &mwg_{m_1,wg} + mwg_{m_2,wg} + mwl_{m_1,\lambda} + mwl_{m_2,\lambda} \leqslant 3 \\ &\forall wg = 1...N_{wg} \\ &\forall \lambda = 1...N_\lambda \\ &\forall m_1, m_2 = 1...N_m : m_1 < m_2 \end{aligned} \tag{5.20}$$

If a pair of messages (m_1, m_2) is on the same waveguide section ($mwg_{m_1,wg} + mwg_{m_2,wg} = 2$ for one section wg), then it cannot use the same wavelength ($mwl_{m_1,\lambda} + mwl_{m_2,\lambda} \leq 1$ for all λ). Conversely, if it is using the same wavelength, it cannot be on the same waveguide section.

Possibility 2. Create binary variables $mwleq_{m_1,m_2}\ \forall m_1, m_2 = 1...N_m : m_1 < m_2$ to indicate if

two messages share a wavelength:

$$\begin{aligned} & mwl_{m_1,\lambda} \wedge mwl_{m_2,\lambda} \Rightarrow mwleq_{m_1,m_2} \\ & \forall \lambda = 1...N_\lambda \\ & \forall m_1, m_2 = 1...N_m : m_1 < m_2 \end{aligned} \tag{5.21}$$

Then, if two messages share a wavelength, they cannot share a waveguide:

$$\begin{aligned} & mwleq_{m_1,m_2} \Rightarrow (mwg_{m_1,wg} + mwg_{m_2,wg} \leqslant 1) \\ & \forall wg = 1...N_{wg} \\ & \forall m_1, m_2 = 1...N_m : m_1 < m_2 \end{aligned} \tag{5.22}$$

Possibility 3. This is very close to possibility 2, but waveguide sections are compared and binary variables $mwgeq_{m_1,m_2}$ $\forall m_1, m_2 = 1...N_m : m_1 < m_2$ are created instead:

$$\begin{aligned} & mwg_{m_1,wg} \wedge mwg_{m_2,wg} \Rightarrow mwgeq_{m_1,m_2} \\ & \forall wg = 1...N_{wg} \\ & \forall m_1, m_2 = 1...N_m : m_1 < m_2 \end{aligned} \tag{5.23}$$

Then, if two messages share a waveguide, they cannot share a wavelength:

$$\begin{aligned} & mwgeq_{m_1,m_2} \Rightarrow (mwl_{m_1,\lambda} + mwl_{m_2,\lambda} \leqslant 1) \\ & \forall \lambda = 1...N_\lambda \\ & \forall m_1, m_2 = 1...N_m : m_1 < m_2 \end{aligned} \tag{5.24}$$

Possibility 4. Use the definitions of $mwleq_{m_1,m_2}$ and $mwgeq_{m_1,m_2}$ together from possibilities 2 and 3 (constraints 5.21 and 5.23 respectively) but go instead with the following set of constraints to enforce exclusivity:

$$mwleq_{m_1,m_2} + mwgeq_{m_1,m_2} \leqslant 1 \qquad \forall m_1, m_2 = 1...N_m : m_1 < m_2 \tag{5.25}$$

Comparison. The four possibilities result in different numbers of variables and constraints. A summary of that comparison is presented in Table 5.1. Even though all possibilities represent the exact same problem (and, as such, are logically interchangeable), some are likely to be faster to solve than others. The heuristics coded in the MIP solver might be more effective with some possibilities than others, for example. Through testing, possibility 2 turned out to be faster, to some extent, and was the one used for the model.

Table 5.1: Comparison between possibilities in describing wavelength exclusion rules.

	Possibility		
	1	2 and 3	4
# Variables	0	N_{pm}	$2*N_{pm}$
# Constraints	$N_{pm}*N_\lambda*N_{wg}$	$N_{pm}*(N_{wg}+N_\lambda)$	$N_{pm}*(1+N_{wg}+N_\lambda)$

Note: $N_{pm} = \binom{N_m}{2} = \frac{N_m*(N_m-1)}{2}$, which is the number of unordered pairs of messages.

5.4.2.3 Activation of routing features

Having added constraints to enforce the selection of paths for messages, constraints must now be added to the model to activate the correct routing features of each GRU such that those paths actually take place in the solution.

MRR placement. To track the MRRs placed by the solver, variables $rum_{g,p,m}\ \forall g = 1...N_{gru}, p \in \mathbb{P}, m = 1...N_m$ and $ru_{g,p}\ \forall g = 1...N_{gru}, p \in \mathbb{P}$ are created. Variables $rum_{g,p,m}$ and $ru_{g,p}$ indicate if the ring on corner position[7] p on GRU g is being used by message m and by any message, respectively.

Rings can only be used by one message. The following constraint both enforces this restriction and sets the value of $ru_{g,p}$:

$$ru_{g,p} = \sum_{m=1}^{N_m} rum_{g,p,m} \qquad \forall p \in \mathbb{P} \qquad \forall g = 1...N_{gru} \tag{5.26}$$

Now, constraints are added to set the values of $rum_{g,p,m}$ based on the paths of the messages. Looking at one single GRU there are four possibilities for the path of the message through that GRU:

1. A message is **not present on any edges** of the GRU: the message does not influence this GRU, so no constraints are needed to cover this case.

2. A message is **present on two edges** of a GRU that **form a corner**: one of the three options from Figure 4.4(b-d) must be active. The following constraints are added (example given for the top-left corner only):

$$mwg_{m,W_g^T} \wedge mwg_{m,W_g^L} \Rightarrow rum_{g,TL,m} \vee rum_{g,BR,m} \vee cb_{g,TL} \qquad \forall \text{ 4 corners} \quad \forall m = 1...N_m \quad \forall g = 1...N_{gru} \tag{5.27}$$

3. A message is **present on two edges** of a GRU that **form a direct path**: in this case, no rings can be placed on the GRU with the wavelength of the message. However, this does not need to be explicitly enforced with constraints because the combination of other constraints

[7] Index $p \in \mathbb{P}$, $\mathbb{P} = \{TL : \text{Top-Left}, TR : \text{Top-Right}, BL : \text{Bottom-Left}, BR : \text{Bottom-Right}\}$.

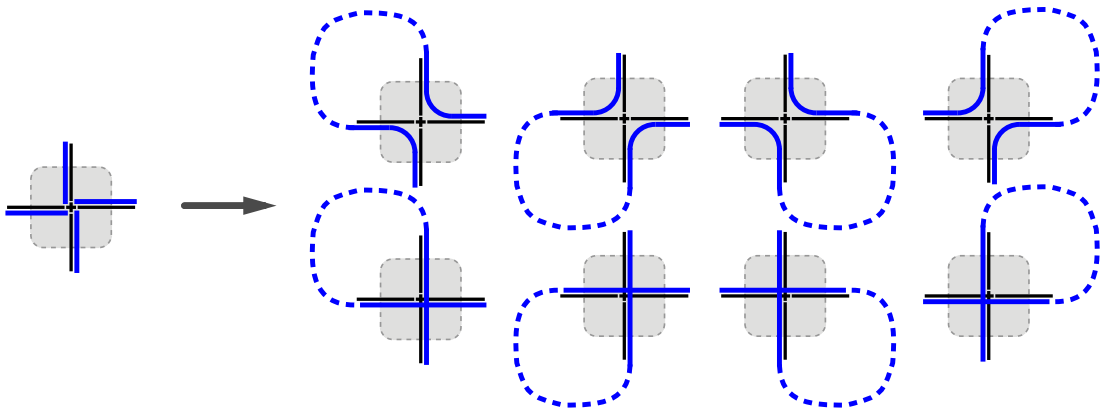


Figure 5.4: Possible 4-edge paths through a GRU.

already implicitly covers this case. In short, because no features need to be turned *on*, no constraints are needed.

4. A message is **present on four edges** of a GRU: this case was removed purposefully, one of the reasons being that it simplified the model. Looking at one GRU locally (i.e., knowing only that its four edges are all used by the message), it is impossible to distinguish between the 8 possible paths, as shown in Figure 5.4, and so it is impossible to know if and where MRRs must be placed or corner bending must be activated. Therefore, to allow four edges, the model could not be written this way. Allowing four edges would lead to a potentially quite different, and certainly more complicated, MIP model.

Corner bending. For corner bending variables $cb_{g,p}$ $\forall g = 1...N_{gru}, p \in \mathbb{P}$ are created[8] to indicate if corner p is bent in GRU g.

Three sets of constraints are needed:

- A GRU cannot have corners bent and rings active at the same time.

$$cb_{g,p_1} + ru_{g,p_2} \leqslant 1 \qquad \forall p_1, p_2 \in \mathbb{P} \qquad \forall g = 1...N_{gru} \tag{5.28}$$

- Corners for the same edge cannot be bent at the same time for the same GRU.

$$\begin{aligned} cb_{g,TL} + cb_{g,TR} \leqslant 1 \qquad & cb_{g,TR} + cb_{g,BR} \leqslant 1 \\ cb_{g,TL} + cb_{g,BL} \leqslant 1 \qquad & cb_{g,BL} + cb_{g,BR} \leqslant 1 \\ \forall g = 1...N_{gru} & \end{aligned} \tag{5.29}$$

[8] This feature can be turned off, if needed, by adding constraints to set all $cb_{g,p}$ variables to zero.

- If a corner is bent, then messages present on one of the edges of that corner must be present on the other (example given for the top-left corner only).

$$cb_{g,TL} \Rightarrow mwg_{m,W_g^T} = mwg_{m,W_g^L} \qquad \begin{array}{l} \forall \text{ 4 corners} \\ \forall m = 1...N_m \\ \forall g = 1...N_{gru} \end{array} \tag{5.30}$$

As stated before, finding feasible solutions requires thinking about multiple messages at once because, for example, not all sets of message paths are possible. By adding these constraints to the model, impossible sets imply contradictions in these constraints. In the case where two messages have paths such that both need to use the same MRR placeholder, these paths automatically become invalid and are no longer a feasible solution to the model. By definition, feasible solutions must comply with all constraints, which means a MIP solver for a model written this way will naturally consider the deep task interdependence, as per the initial goal.

5.4.2.4 Message insertion loss

The model must calculate the insertion loss of each message based on its path. As explained in Section 3.1.4, the insertion loss of a message is the sum of five loss types, so constraints must be added to deal with each. Some types of losses – bending loss, crossing loss and drop loss – only happen on GRUs. For those the restriction introduced in Section 5.4.2.1 is very useful: it limits to one the number of occurrences of each type of loss for each GRU, thus simplifying the model once again.

Bending loss. Only bending loss due to corner bending in GRUs must be considered here, because bending loss due to bends in waveguide sections is considered on their *extraloss* property already. Binary variables $bl_{g,m}$ $\forall g = 1...N_{gru}, m = 1...N_m$ are created to indicate if message m has bending loss in GRU g. To set the values for those variables the following constraints are added (example given for the top-left corner only):

$$mwg_{m,W_g^T} \wedge mwg_{m,W_g^L} \wedge cb_{g,TL} \Rightarrow bl_{g,m} \qquad \begin{array}{l} \forall \text{ 4 corners}, \\ \forall m = 1...N_m, \\ \forall g = 1...N_{gru} \end{array} \tag{5.31}$$

Through loss. If a message is going through the direct path on a GRU then it has through loss for each MRR present on that GRU. Binary variables $tl_{g,p,m}$ $\forall g = 1...N_{gru}, p \in \mathbb{P}, m = 1...N_m$ are created to indicate if message m has through loss due to a MRR in corner p in GRU g. To set the values for those variables, the following constraints are added (example given for

the horizontal direction only):

$$mwg_{m,W_g^L} \wedge mwg_{m,W_g^R} \wedge ru_{g,p} \Rightarrow tl_{g,p,m} \qquad \begin{array}{l} \forall \text{ 2 directions}, \\ \forall m = 1...N_m, \\ \forall p \in \mathbb{P}, \\ \forall g = 1...N_{gru} \end{array} \tag{5.32}$$

Crossing loss. As explained in Section 4.2.1.1, avoiding the center crossing in GRUs can help decrease crossing loss for some messages. The center crossing *cannot* be avoided when there are messages going through the GRU horizontally *and* vertically. The first step in modeling this situation is to create binary variables $mch_g, mcv_g \; \forall g = 1...N_{gru}$. These variables indicate if at least one message is going through the center of the GRU horizontally and vertically, respectively. The following constraints set their values:

$$mwg_{m,W_g^L} \wedge mwg_{m,W_g^R} \Rightarrow mch_g \qquad \begin{array}{l} \forall \text{ 2 directions}, \\ \forall m = 1...N_m, \\ \forall g = 1...N_{gru} \end{array} \tag{5.33}$$

$$mwg_{m,W_g^T} \wedge mwg_{m,W_g^L} \wedge rum_{g,BR,m} \Rightarrow mch_g \wedge mcv_g \qquad \begin{array}{l} \forall \text{ 4 corners}, \\ \forall m = 1...N_m, \\ \forall g = 1...N_{gru} \end{array} \tag{5.34}$$

Then, binary variables $cl_{g,m} \; \forall g = 1...N_{gru}, m = 1...N_m$ are created to indicate if message m suffers crossing loss on GRU g. The value of $cl_{g,m}$ follows:

$$mwg_{m,W_g^T} \wedge mwg_{m,W_g^B} \wedge mch_g \Rightarrow cl_{g,m} \qquad \begin{array}{l} \forall \text{ 2 directions}, \\ \forall m = 1...N_m, \\ \forall g = 1...N_{gru} \end{array} \tag{5.35}$$

Drop loss is proportional to the number of MRRs used by each message and does not require any extra variables or constraints.

Propagation loss is proportional to the length of the waveguides the message goes through and does not require any extra variables or constraints.

To calculate the total insertion loss of a message over all waveguides and GRUs, continuous variables $mil_m \; \forall m = 1...N_m$ are created. Their values are set with the following constraints, which

are just a weighted sum:

$$\begin{aligned} mil_m = & \sum_{i=1}^{N_{wg}} (L^P * L_i + L_i^E) * mwg_{m,i} \\ & + \sum_{g=1}^{N_{gru}} (L^C * cl_{g,m} + L^B * bl_{g,m}) \\ & + \sum_{g=1}^{N_{gru}} \sum_{p \in \mathbb{P}} (L^T * tl_{g,p,m} + L^D * rum_{g,p,m}) \qquad \forall m = 1...N_m \end{aligned} \tag{5.36}$$

5.4.3 Optimization function

The three optimization targets from Section 3.4 are considered. Three functions for the message insertion loss are contemplated, but since the value for the insertion loss of each message is available through the mil_m variables, other functions can be added to the model if needed (assuming they can be linearized).

Number of wavelengths. Binary variables $wlu_\lambda \; \forall \lambda = 1...N_\lambda$ are created to indicate if wavelength λ is used. Their values are set with:

$$\begin{aligned} wlu_\lambda \geqslant mwl_{m,\lambda} \qquad & \forall m = 1...N_m, \\ & \forall \lambda = 1...N_\lambda \end{aligned} \tag{5.37}$$

Integer variable *nwl* is created to hold the number of wavelengths used. Its value is set with:

$$nwl = \sum_{\lambda=1}^{N_\lambda} wlu_\lambda \tag{5.38}$$

Maximum insertion loss over all messages. Continuous variable *maxil* is created to hold the maximum insertion loss over all messages. Its value is set with:

$$maxil \geqslant mil_m \qquad \forall m = 1...N_m \tag{5.39}$$

Sum over all wavelengths of the maximum insertion loss over all messages on each wavelength. Continuous variables $maxilwl_\lambda \; \forall \lambda = 1...N_\lambda$ are created to contain the value for the maximum insertion loss over all messages using wavelength *wl*. These variables are set with:

$$\begin{aligned} mwl_{m,\lambda} \Rightarrow maxilwl_\lambda \geqslant mil_m \qquad & \forall m = 1...N_m \\ & \forall \lambda = 1...N_\lambda \end{aligned} \tag{5.40}$$

Sum of the insertion loss over all messages. No additional variables or constraints are needed.

Number of MRRs. No additional variables or constraints are needed.

Finally, the following objective function is minimized:

$$\alpha_1 * nwl + \alpha_2 * maxil + \alpha_3 * \sum_{\lambda=1}^{N_\lambda} maxilwl_\lambda + \alpha_4 * \sum_{m=1}^{N_m} mil_m + \alpha_5 * \sum_{g=1}^{N_{gru}} \sum_{p \in \mathbb{P}} ru_{g,p} \tag{5.41}$$

where α_i are optimization weights chosen by the designer.

5.4.4 Heuristics and model reduction techniques

5.4.4.1 Path hints for messages

One of the major complexities of the WRONoC design problem with a layout template lies in the combinatorial nature of the search for, and evaluation of, all the possible sets of paths for the messages through the template[9]. In combinatorial problems, heuristics are often the most useful way to find "good-enough" feasible solutions in a timely manner.

Good MIP solvers already have very advanced heuristics put in place to shorten solve times, but these are designed to work on MIP models in general, and so lack the power of a good custom-tailored heuristic that uses specialized knowledge about the problem at hand.

In the case of the WRONoC design problem it is therefore expected that using heuristics to help the MIP solver find good paths fast will cut down on solve times. One of the possible ways to do this is to figure out *a priori*, by looking at the layout template, paths that some messages are very likely to follow. That information can be given to the solver as a hint which will be used to help form the first feasible solution. It is not even required to give a path hint for all messages, although the more the better (assuming they are of good quality). Also, the higher the quality of the hints, the closer the first feasible solution is to the optimal solution. As will be explained later, some templates lend themselves very well to such an approach (see Section 6.2.1.1).

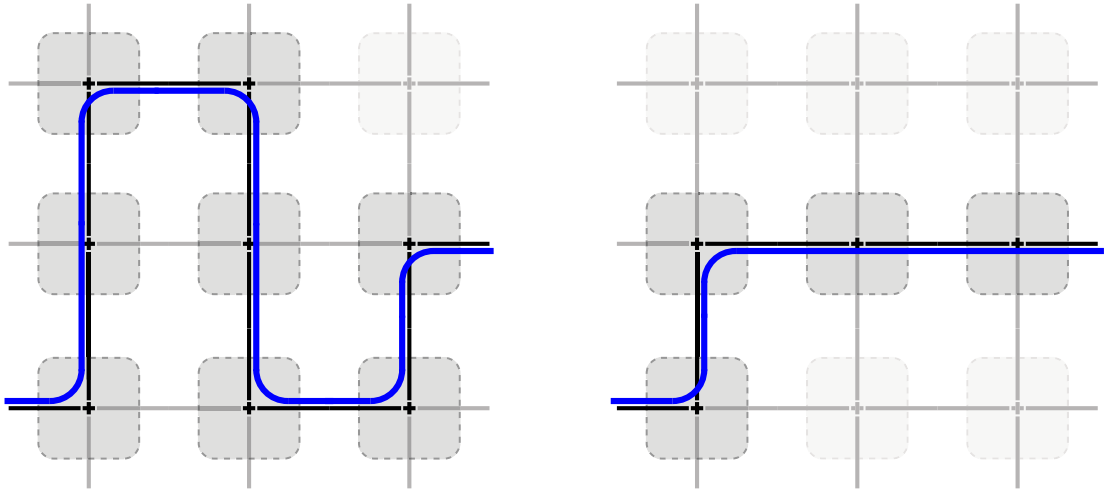
5.4.4.2 Restrictions on usage of MRRs

This next model reduction is an heuristic because it *can* remove the optimal solution from the solution space but testing shows it is *very unlikely* to do so if used correctly (see Section 7.2.3). Yet, contrary to Section 5.4.4.1, this directly changes the way the model is solved by adding constraints.

Empirically, it is found that messages prefer very direct paths, i.e, the optimized solution will almost always choose Figure 5.5(b) over Figure 5.5(a). This is especially true if the direction changes are made using MRRs, because not only does the path in Figure 5.5(a) use more MRR positions in that case (which are valuable in compact templates due to their limited amount), but also because Figure 5.5(a) is very unlikely to result in a smaller insertion loss than Figure 5.5(b).

Yet, if the solver is stopped mid-way through the optimization procedure, paths such as Figure 5.5(a) (and where every bend is achieved with an MRR) are likely to appear in the best feasible

[9] The other is the assignment of wavelengths, also a combinatorial problem.



(a) (b)

Figure 5.5: Example of a convoluted (a) and an equivalent (same start and end) simpler path (b) through a layout template.

solution up to that moment. It appears solvers spend time looking at quite complicated paths when, in many cases, a simpler solution is quite obvious.

To mitigate this issue, constraints can be added to the model that force a maximum number of MRRs per message (R^{max}):

$$\sum_{g=1}^{N_{gru}} \sum_{p \in \mathbb{P}} rum_{g,p} \leqslant R^{max} \qquad \forall m = 1...N_m \tag{5.42}$$

This forces the solver to only consider simpler paths right from the onset, and so leads to noticeably reduced solve times. Corner bending is not restricted because bends are much "cheaper" in terms of insertion loss and so with corner bending active it can actually be counter-productive to avoid convoluted paths (since it can be observed that most optimized solutions will take advantage of corner bending to make messages snake through the template in unexpected ways).

The choice of R^{max} is paramount in defining the usefulness of this heuristic. Too big and this heuristic has little impact, but too small and too much of the solution space might be removed (potentially along with the optimal solution). In some extreme cases (for example, $R^{max} = 1$ or $R^{max} = 0$), it might make the model unfeasible. This choice needs to be done through specific layout template analysis.

For most templates, however, $R^{max} = 2$ turns out to be a good option. This can be justified intuitively with Figure 5.6. In this figure all path types in terms of the number of bends, given the set of possible sender and receiver positions and orientations, are shown. Note that some paths require a minimum of up to four bends. Two remarks are in order:

- Any other path with more bends can be simplified into one of these six types, unless the

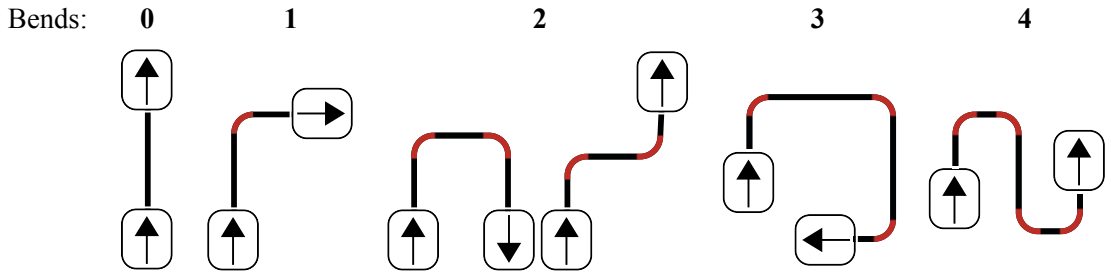


Figure 5.6: Minimum number of bends given each set of sender and receiver positions and orientations.

layout template and/or other messages create restrictions that force the message to take that more complicated path.

- The path types requiring three or four bends arise from difficult positions and orientations of the sender and receiver waveguides which are less likely to occur or can be easily avoided (for example, simply inverting the receiver orientation on the path with 4 bends brings the minimum down to 2).

As mentioned before this heuristic is template specific, but these observations foster the conclusion that most templates end up requiring no more than two bends for all messages.

5.4.4.3 Restrictions on usage of wavelengths

The assignment of wavelengths to messages contains redundancy. In a model with three messages, for instance, both solutions presented below are equivalent:

$$\begin{array}{lll} m_1 \to \lambda_1 & & m_1 \to \lambda_2 \\ m_2 \to \lambda_1 & \text{and} & m_2 \to \lambda_2 \\ m_3 \to \lambda_2 & & m_3 \to \lambda_1 \end{array}$$

Because all wavelengths are symbolic (i.e., $\lambda_1, \lambda_2, \ldots$), the important information is not which wavelength each message uses, but what messages can use the same wavelength. To better understand this fact it helps to view the set of variables $mwl_{m,\lambda}$ as a matrix, where messages are rows and wavelengths are columns:

$$\begin{bmatrix} mwl_{m_1,\lambda_1} & mwl_{m_1,\lambda_2} & \ldots & mwl_{m_1,\lambda_{N_\lambda}} \\ mwl_{m_2,\lambda_1} & mwl_{m_2,\lambda_2} & \ldots & mwl_{m_2,\lambda_{N_\lambda}} \\ \vdots & \vdots & \ddots & \vdots \\ mwl_{m_{N_m},\lambda_1} & mwl_{m_{N_m},\lambda_2} & \ldots & mwl_{m_{N_m},\lambda_{N_\lambda}} \end{bmatrix}$$

This matrix has precisely one nonzero element in each row, and is thus very close to a permutation matrix. The difference is that messages can share wavelengths, and so one column can have more than one nonzero element. In this representation the redundancy in the solutions can be

expressed by stating that swapping the values between any two columns still yields effectively the same solution. From the example above (assuming $N_\lambda = N_m$):

$$\begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \equiv \begin{bmatrix} 0 & 1 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}$$

To reduce some of these meaningless variations around the same effective solution, the following set of constraints can be added:

$$mwl_{m,\lambda} = 0 \qquad \forall \lambda = (m+1)...N_\lambda \qquad \forall m = 1...N_m \tag{5.43}$$

Their effect is to force that matrix to become a lower triangular matrix:

$$\begin{bmatrix} mwl_{m_1,\lambda_1} & 0 & \dots & 0 \\ mwl_{m_2,\lambda_1} & mwl_{m_2,\lambda_2} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ mwl_{m_{N_m},\lambda_1} & mwl_{m_{N_m},\lambda_2} & \dots & mwl_{m_{N_m},\lambda_{N_\lambda}} \end{bmatrix}$$

In doing so many sets of equivalent solutions are reduced in size (in the example above the solution on the right is removed, but the left one stays) and the solver is no longer burdened with exploring redundant solutions. Note that any feasible solution matrix *can* be converted into a lower triangular matrix by swapping columns (proof in Appendix B). Because of that, forcing the matrix to be lower triangular will *never* remove the optimal solution from the solution space.

5.4.4.4 Extra constraints

Some extra constraints shown below were added to the model. These do not alter the model logically, but managed to consistently improve solve times in testing.

- A message cannot have crossing, bending or drop loss at the same time for the same GRU.

$$cl_{g,m} + bl_{g,m} + \sum_{p \in \mathbb{P}} rum_{g,p,m} \leqslant 1 \qquad \forall g = 1...N_{gru}, \quad \forall m = 1...N_m \tag{5.44}$$

5.5 Feasibility proof

It is possible that the chosen layout template cannot satisfy the entire communication matrix. This can happen, for instance, if the template does not have enough MRR placeholders to route all messages. For those cases the WRONoC model above will be unfeasible. Yet, checking for the existence of a solution can actually be done much faster by solving a simplified version of the

model. For this version N_λ must equal N_m, the optimization function is set to a constant value (to stop the optimization process immediately once a feasible solution is found) and a wavelength is uniquely assigned to each message by adding these constraints:

$$mwl_{m,\lambda} = 1 \qquad \forall m = 1...N_m, \lambda = m \tag{5.45}$$

$$mwl_{m,\lambda} = 0 \qquad \forall m = 1...N_m, \lambda \neq m \tag{5.46}$$

This removes one of the three major efforts the solver is tasked with: wavelength assignment. Yet, if the solver is unable to find a feasible solution for this simplified model, the complete model is also unfeasible.

Proof. Assume a feasible solution exists. It will have $nwl \leqslant N_m$. From that solution build another where each message uses its own wavelength (thus either maintaining or increasing nwl). Any message that changes its wavelength in that process must also change the wavelength of the MRRs it uses. This is always possible because each MRR routes only one message. Furthermore, the wavelength/waveguide exclusion rule is still always satisfied. Hence, the feasibility of the complete model implies the existence of a solution for the simplified version. In conclusion, if the simplified model is unfeasible, the complete model is also unfeasible. □

5.6 3-step optimization

Solving the presented MIP model once for the required optimization function is enough to get the optimal solution. However, due to the nature of the problem, it is possible to slightly alter the optimization process yielding more control and faster results. This leads to the 3-step optimization process proposed below. In this process each step optimizes a slightly different version of the model and produces a solution used at the start of the next step.

5.6.1 First step

With big problem sizes, MIP solvers may take a considerable amount of time to find the first feasible solution (assuming one exists). For this problem, the same model simplification used for the feasibility proof in Section 5.5 can also be used as a very fast way to generate the first feasible solution. Therefore, **in the first step**, that simplified model is solved. If a feasible solution exists, it is then used as a warm-start for the remaining optimization process and this can, in some cases, decrease optimization times substantially. If unfeasible, this also has the added bonus of stopping the process as quickly as possible.

5.6.2 Second step

One major source of model complexity is the number of wavelengths considered (N_λ). This number can be anywhere in the range $\{1...N_m\}$. On the one hand, if that number is too low, the model

might be unfeasible[10]. On the other hand, lower numbers are preferable because model solving is faster.

Unfortunately, no guarantee *a priori* can be given about some value N_λ smaller than N_m being feasible[11]. It is quite possible that a certain choice of N_λ will lead to wasted time, as time must be spent solving the model once, ultimately proving unfeasibility, then more time must be spent solving the model again with another choice of N_λ. Also, the first step has given the solver a feasible solution for $N_\lambda = N_m$. So, the best course of action is to consider $N_\lambda = N_m$ and take advantage of the solution already in hand.

At this point the model *can* be solved directly for the optimization function given by the designer. However, it is now time to make one assumption that always holds true: *the designer of the WRONoC will want to use less wavelengths than messages*[12]. Therefore, **in the second step**, the extra constraints from step one are removed and the model is solved a second time, but only the number of wavelengths is minimized. This results in a new feasible solution which will use a reduced number of wavelengths[13], nwl. Now the $N_m - nwl$ unused wavelengths can be removed from the model and solving for the goals of the designer with the smaller $N'_\lambda = nwl$ will be correspondingly faster.

5.6.3 Third step

At this point a feasible solution exists for a small number of wavelengths and the model is also reduced to that number of wavelengths. **The remaining step** is to take that solution as a warm-start to the optimization for the goals decided by the designer (such as minimizing the maximum insertion loss). The final solution has now been reached.

5.6.4 Final comments

In the second step the designer can choose whether to fully minimize the number of wavelengths or to stop the optimization midway when an acceptably low number has been reached. This gives the designer more flexibility in the cases where minimizing wavelengths is more of a secondary goal and it is feared that optimizing the number of wavelengths "too much" might significantly reduce the quality of the final solution.

However, experience shows this is rarely the case anyway. If the quality of the final solution is in fact reduced, it is normally by a negligible amount. Also, when this happens, the better

[10] As an (extreme) example for this, think of a communication matrix that defines an endpoint sending 7 messages (i.e., those 7 messages all go through the same waveguide section of the endpoint), yet the choice is made for $N_\lambda = 5$. The model is clearly unfeasible because it is impossible to avoid interference between the messages.

[11] As exemplified in Footnote 10, there will be a minimum number of wavelengths (which depends on the communication matrix) below which the model is certain to be unfeasible (this number is normally much lower than N_m). But this proves unfeasibility, not feasibility. The physical layout template might still not allow for that minimum value (as is very commonly the case).

[12] If this doesn't hold, then force the model to use N_m wavelengths by keeping the extra constraints from the first step, skip this step and go directly to the third step.

[13] In all the tests and experience gathered during this work the only cases where the number of wavelengths couldn't be lowered below the number of messages were cases where the number of messages was already really small, in which case the model is already fast to solve and so this optimization effort is unnecessary anyway.

solution regularly uses a much bigger number of wavelengths, therefore losing its utility given the assumption stated earlier.

In conclusion, this 3-step optimization procedure is a more efficient way to solve the model with virtually no trade-off in the quality of the final solution.

Chapter 6

WRONoC design workflow

Having described the algorithm in detail in the last chapter, attention must now be turned to the WRONoC design workflow. Here the following two topics, both of which constitute integral parts of this workflow, are covered:

- The toolchain developed to implement the optimization algorithm and to aid in WRONoC design.
- Guidelines for physical layout template design, along with template examples.

6.1 Design toolchain

A specialized toolchain was developed to implement the algorithms presented in the previous chapters and simplify WRONoC design. This toolchain consists of three tools and two file types, all working together to form the WRONoC design workflow. A typical workflow is depicted in Figure 6.1 along with example contents for each file for a small 4x4 WRONoC with 8 messages.

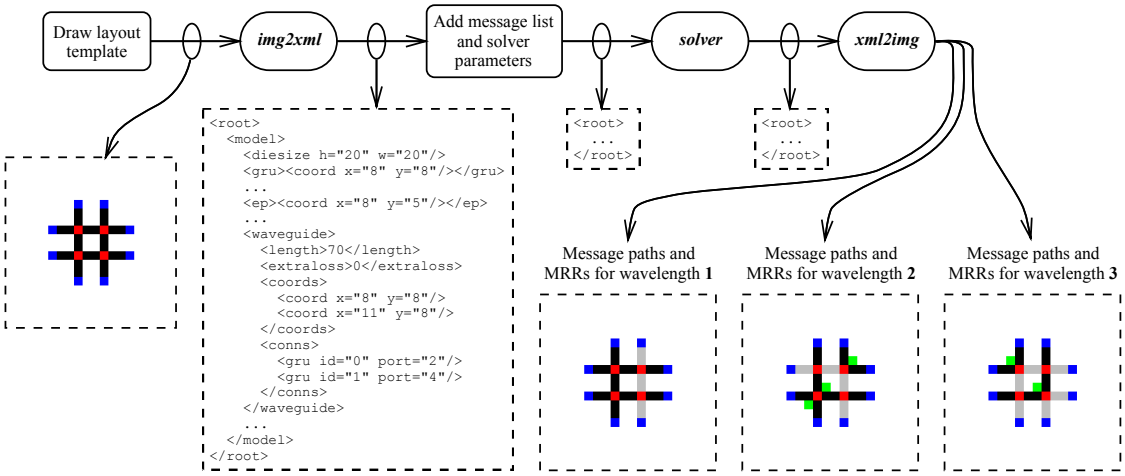


Figure 6.1: WRONoC design workflow.

6.1.1 File types

Two file types are used:

- An XML-based text file to describe in full the inputs and outputs of the optimization procedure.
- An image file[1] to represent a physical layout template and to visualize the optimization results.

The XML files can include the following pieces of information:

- Inputs:
  - Physical layout template: die size, physical locations of GRUs, physical locations of endpoints, physical paths of waveguides, length of waveguides, *extraloss* of waveguides, connections of waveguides to GRUs and endpoints.
  - Communication matrix: sender and receiver endpoints for each message.
  - Technology parameters: values for each type of loss.
  - Solver weights for each target in the optimization function outlined in Section 5.4.3.
  - Solver parameters: explained in detail next in Section 6.1.2.
  - Solver hints: hints for the path/wavelength of some/all messages and the state of some/all GRUs.
  - Solver locks: forced paths/wavelengths for some/all messages and forced state of some/all GRUs.
- Outputs:
  - Message information: wavelength, path through waveguides, insertion loss.
  - GRU state: bent corners or wavelength of each placed MRR.
  - General optimization results: number of used wavelengths, number of used MRRs, maximum insertion loss.

The image files are a pixel-by-pixel scale representation of the WRONoC. The width and height of one pixel in the image is equivalent to a length value of d µm meaning one pixel corresponds to a d^2 µm^2 area of the optical plane. The bounds of the image are the bounds of the entire optical plane. Each non-white pixel on the image is a WRONoC element: **red** pixels are GRUs, **blue** pixels are endpoints and **black** pixels are waveguides. If the image represents an optimization result, unused waveguides are **gray** and MRRs placed by the solver are **green**.

[1] The actual image format (PNG, JPG, BMP, etc) can be any lossless color format.

6.1.2 Design tools

The three developed tools use the aforementioned file types to perform the WRONoC optimization:

img2xml is a tool that takes the image of a physical layout template and transforms it into a XML description of the template to serve as input to the optimization.

solver is the optimization tool that takes an XML file with a physical layout template, the communication matrix and solver parameters and solves it for the optimal WRONoC design, also outputting an XML file.

xml2img is a tool that takes the resulting XML file from the optimization and transforms it into a collection of images visually describing the optimized WRONoC design. One image is created for each wavelength used in the final solution.

The two image tools were written in Python and the *solver* tool is written in C++. The *solver* tool makes use of a WRONoC library developed to create the MIP model and conduct the optimization, which itself uses Gurobi [39] to solve the MIP model. This tool also takes multiple parameters as inputs to control the optimization process which are described in Table 6.1.

Table 6.1: WRONoC solver tool parameter list and description.

Name	Type	Purpose
cornerBending	*bool*	Allow or forbid corner bending in GRUs.
maxRingsPerMessage	*int*	Set value of R^{max}. A value of zero turns this heuristic off.
wavelengthOptimalityThreshold	*int*	How close to the best bound must the solution in step 2 be to start step 3.
wavelengthUsageSlack	*int*	How many extra wavelengths are allowed in step 3 beyond those used by the final solution given by step 2.
finalOptimalityThreshold	*double*	How close must the solution in step 3 be to the best bound to finish the optimization. A value of zero will force the solver to find the proven optimal solution.
maxSecs	*int*	Maximum number of seconds before optimization is stopped.
skipWarmStartGeneration	*bool*	Skip first optimization step.
threadCount	*int*	Maximum thread usage by the MIP solver.

6.2 Layout template design

The first step in the WRONoC design workflow is to create a suitable physical layout template. A brief word is now given about this process.

6.2.1 Centralized grid template

The design of a layout template is highly dependant on the location of the nodes on the optical plane. Nevertheless, some layout template topologies exist that can be applied to virtually any

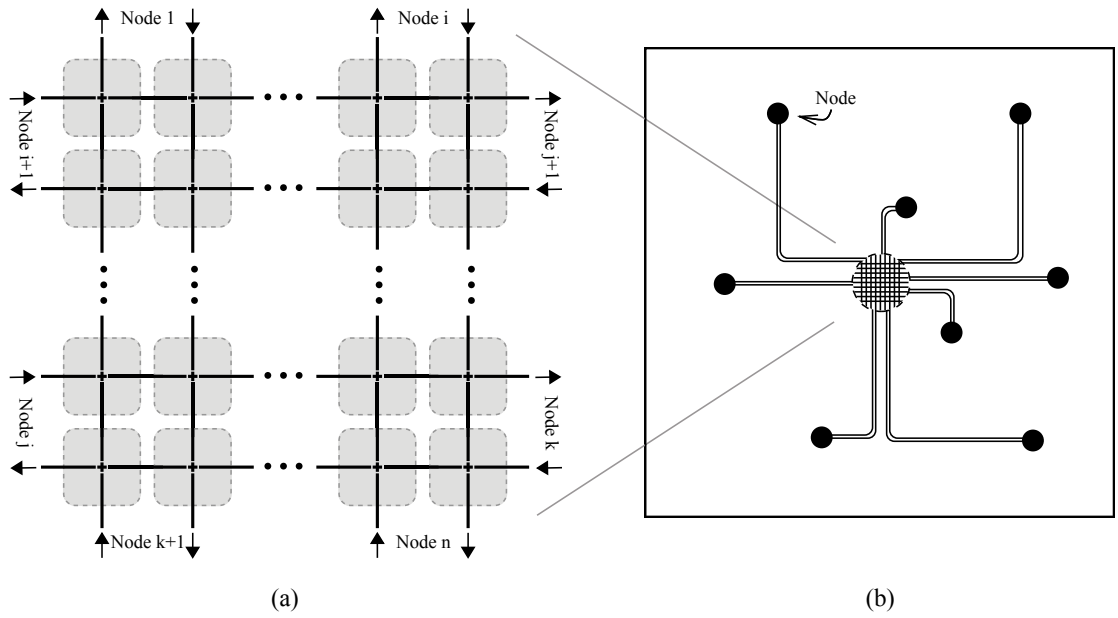


Figure 6.2: Centralized grid layout template. (a) Grid structure. (b) Grid placement on the optical plane and external routing.

node configuration. One such topology is the centralized grid template. This template consists of a grid of GRUs interconnected by waveguides. A $w \times h$ grid has $w \times h$ GRUs, $2w + 2h$ ports and connects $w + h$ nodes. Each node connects to two ports on the grid which are next to each other: one port connects to the modulator and the other to the demodulator – Figure 6.2(a).

The grid itself can be placed anywhere on the optical plane, but in most cases placing on the center of the die or on the center of mass of the nodes is already enough to improve upon the state of the art.

The assignment of grid ports to each node is something that can have a measurable impact on the quality of the solution because it influences the paths the messages take inside the grid. Without deeper analysis on a case by case basis, however, it is very difficult to predict the best assignment. Nonetheless, having decided on a position for the grid on the optical plane, there will exist very few assignments of ports to nodes where no crossings external to the grid are created and which also follow the rule explained next in Section 6.2.2. Since crossing loss strongly influences the overall power usage, the procedure which is in equal parts simple and effective is just to use one of those assignments.

Having made a choice on port assignment, the waveguides connecting the nodes to the grid can then be manually routed to minimize their length and number of bends – Figure 6.2(b).

6.2.1.1 Message path hints

As explained in Section 5.4.4.1, path hints given to the solver before the optimization starts are expected to cut down on solve times and the centralized grid router shows quite an advantage in

this area.

In general, for any template, messages are likely to need only a small number of MRRs (up to 4 MRRs, although 2 is the more frequent maximum – see Section 5.4.4.2). In centralized grid routers messages do *not* need more than 2 MRRs. In fact, many messages use only 0 or 1 MRRs and almost always have very clear paths:

- Messages whose entrance and exit ports are **directly aligned** are *very* likely to take the direct path and use **0 MRRs** – see blue path in Figure 6.3.
- Messages whose entrance and exit ports are **on perpendicular sides of the router** are *very* likely to use only **1 MRR**, making their path obvious as well – see red path in Figure 6.3.

Messages that use 2 MRRs have multiple path options, so path hints are not given for them – see green paths in Figure 6.3.

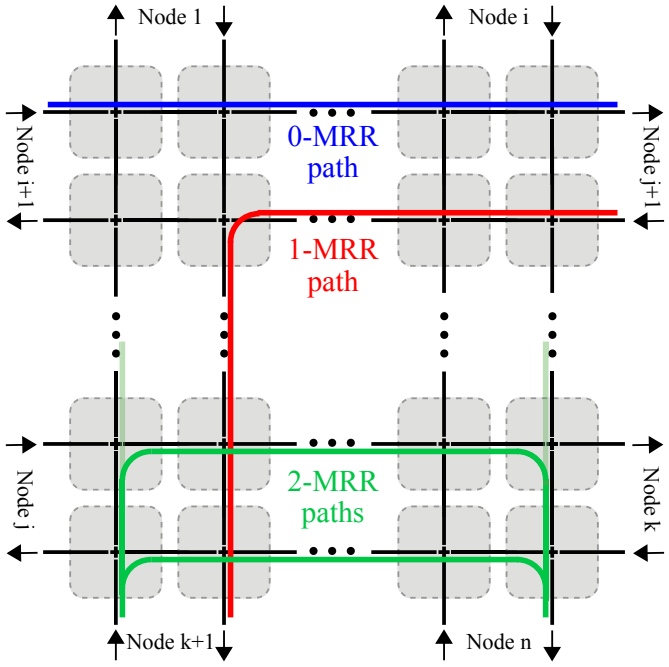


Figure 6.3: Path types through a centralized grid router.

When these simple 0 or 1 MRR paths are less likely to occur is when corner bending is turned on and the communication matrix is very sparse. In those cases, because there are few messages to route, each message has the space and freedom to take much more convoluted paths through the grid (see Section 7.3). Also, 90° turns are "cheap" because they can be done with corner bending instead of MRRs. Even so, these path hints still contribute to valid solutions and, even if they don't appear in the final optimized solution, they will still help the solver in finding good feasible solutions fast.

6.2.2 Accounting for the optical power distribution network

Although the OPDN is not directly considered in this work (see Section 3.3), it is also not completely ignored. Some templates are independent of the OPDN, i.e., when the OPDN is added to the optical plane after the WRONoC router has been optimized, no extra crossings between the router and the OPDN are generated. To build templates with this characteristic there is only one simple rule to follow: there must be a free path from each modulator to the laser source – Figure 6.4. If the template design follows this rule then the interdependency between the OPDN and the physical layout of the router is broken.

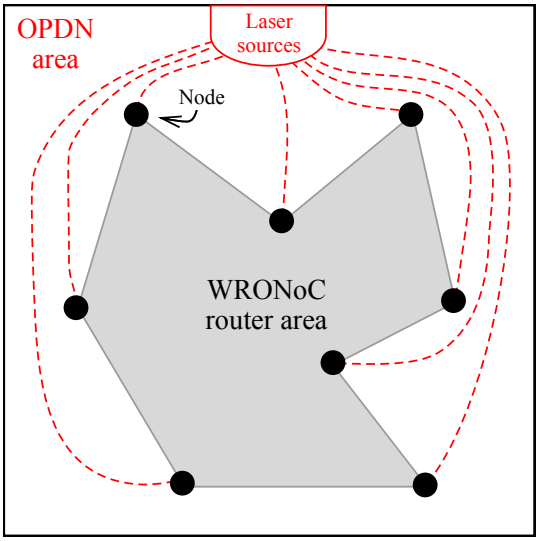


Figure 6.4: Breaking the inter-dependency between the OPDN and the physical layout of the router when using a layout template (with an example for off-chip laser sources) by dividing the optical plane into a “template area” and an “OPDN area”.

6.2.3 Accounting for thermal hotspots

As explained in Section 3.1.1 the WRONoC must be designed with temperature variations and hotspots in mind. These are due to the electrical layers and so are already well characterized when creating the WRONoC. Although the implemented approach does not explicitly optimize the placement of MRRs and waveguides for thermal hotspots, the physical layout template can be designed with that in mind. For example, if the centralized grid template is used, the grid itself can be placed on the coldest place of the optical plane and the waveguides connecting the grid to the nodes can be routed around the major hotspots.

Chapter 7

Results & analysis

In this chapter the implemented approach for optimizing WRONoCs is fully tested and analyzed. First, a comparison is made with the state of the art procedure and tools. Then, a thorough analysis of the centralized grid template is executed. This analysis brings not only useful insight into the effectiveness of the template itself, but also into the performance of the optimization algorithm, the usefulness of its heuristics and the advantage of multiple GRU designs. Finally, a full example result is shown. All tests were conducted on 2.2 and 2.6 GHz CPUs.

7.1 Comparison to the state of the art

The state of the art procedure for WRONoC design is to manually choose a logical topology and then automate its placement and routing using the Proton+ tool [3]. A comparison between the implemented procedure herein and the best results from Proton+ was carried out. Most of the result analysis from Proton+ is dedicated to an 8 node test case with 44 messages. This same test case (considering the same communication matrix, node placement, die size, PSE/GRU size and loss parameters) was solved with the proposed tool. The second step of the optimization fully optimizes for the number of wavelengths and the third step was set to minimize the maximum insertion loss, just like Proton+.

Proton+ compares results originating from P&R of three logical topologies (8x8 λ-Router, 8x8 GWOR and 8x8 Standard-Crossbar), five different sets of node positions for those 8 nodes and various permutations of solver parameters. The present comparison uses the same node positions where Proton+ got the absolute best result, which are shown in Figure 7.1(a).

7.1.1 Layout templates

Three layout templates were manually designed to tackle this test case. All templates share some common features:

- Each node has two endpoints, a modulator and a demodulator, just like the logical topologies used in Proton+.

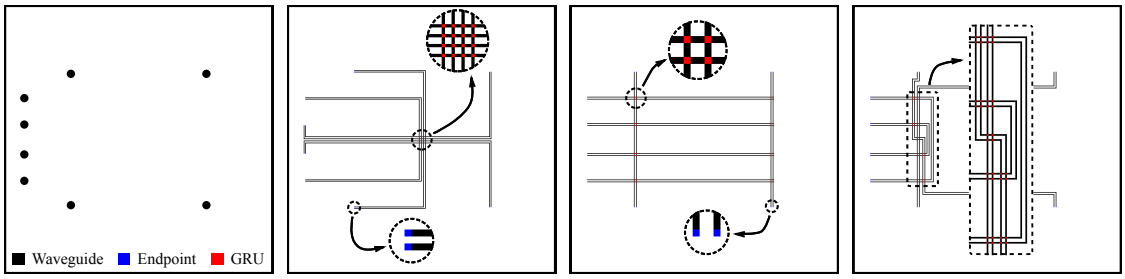


Figure 7.1: Physical layout templates used in Proton+ comparison. (a) Location of the eight nodes that produces the best result in Proton+. (b) A centralized grid template connecting those nodes. (c) A distributed grid template. (d) A custom template.

- All templates avoid extra crossings from the OPDN by following the rule explained in Section 6.2.2.

The **centralized grid** template follows the description in Section 6.2.1. In this case the grid itself was placed on the center of the die – Figure 7.1(b). The **distributed grid** template was built by placing horizontal or vertical pairs of waveguides starting at each node, with a GRU on each crossing – Figure 7.1(c). The **custom** template was built specifically for this test case (i.e., these node positions and communication matrix). In particular, no message needs to use more than one MRR – Figure 7.1(d).

For the first two templates R^{max} was set to 2, but for the third it was set to 1, since it was designed with this in mind. No path hints were given for any template and corner bending was not used.

7.1.2 Results

Table 7.1 presents the various comparisons. Most important are the number of wavelengths and maximum insertion loss, but #MRRs and execution time are also given. Results given by the new

Table 7.1: Results of comparison to Proton+ for 8 nodes and 44 messages.

	#WLs	**Max IL (dB)**	#MRRs	Time (s)	
Proton+				T_{total}	
λ-Router	**8**	**6.6 - 9.0**	56	134	
GWOR	**7**	**8.1 - 11.3**	48	79	
Std. crossbar	**8**	**10.5 - 13.0**	64	601.6	
This approach				T_{opt}	T_{total}
Centralized	**8**	**3.1**	52	178	271
Distributed	**8**	**3.6**	48	37	376
Custom	**7**	**4.1**	40	-	6

T_{opt} is time to find the optimal solution, T_{total} is total execution time.
Implemented approach: $T_{total} = T_{opt}$ + time to prove optimality.
Proton+: T_{total} = time that produces the best IL result.

approach are the optimal solutions.

7.1.2.1 Number of wavelengths

Each node has only one modulator and some send 7 messages. Thus, 7 wavelengths is the minimum. The custom template achieves this value, but the grid templates require 8. However, the new approach can reduce this number if given a sparser communication matrix, whereas the logical topologies used in Proton+ are fixed, i.e., a smaller number of messages will always result in the same amount of wavelengths.

7.1.2.2 Max. insertion loss

The new approach produces results that are *twice* to *three times* better. This proves the substantial benefits of developing a combined logical topology and physical layout optimization algorithm and proves the working hypothesis presented in Section 3.3 which is at the core of this thesis.

7.1.2.3 MRR usage

This was not an optimization objective in these tests. Nevertheless, the comparison to Proton+ remains favourable.

7.1.2.4 Time

Grid templates have a total execution time comparable to Proton+. The custom template is much faster, mostly because of the R^{max} heuristic. Not shown on the table, but nonetheless still relevant, is the fact that solving the custom template with $R^{max} = 2$ is orders of magnitude slower, which proves the effectiveness of this heuristic.

Furthermore, the optimal solution is consistently reached in half or less than the total execution time. Thus, a designer that does not require proof of optimality can end the optimization once a satisfactory solution is found which, based on these results, is likely to appear quickly and be close to optimal. A more definitive argument for this is given in Section 7.2.5.

7.2 Solver performance and centralized grid template analysis

The main purpose of this section is to characterize the implemented solving algorithm in its performance and result quality. Every single performance metric (such as time, number of wavelengths, insertion loss, etc) depends heavily on the chosen layout template. Therefore, for this analysis, only the centralized grid router is used. This helps make results consistent and comparable. Also, because the centralized grid router is a physical layout template that can be applied to virtually any WRONoC use case, this also makes these results representative of the minimum potential of this novel WRONoC design approach in any case.

A series of tests will now be presented and commented upon. Every test aims to analyze one specific characteristic either of the centralized grid template or of the solver itself. In these tests the main sources of change are the number of messages in the communication matrix, solver parameters such as *cornerBending* (see Table 6.1) and the use of specific efforts to improve solve times such as the R^{max} heuristic, path hints and the 3-step optimization procedure. Conclusions are drawn by looking at results such as the required number of wavelengths and MRRs, the message insertion loss and the solve times.

Unless otherwise stated, all tests use an 8 node centralized grid router solved for multiple random sets of N_m messages, with $N_m = 1...56$ (no self-communication was considered), while recording solve time, number of wavelengths, number of MRRs and maximum insertion loss. The 3-step optimization procedure is used (the second step fully optimizes for the number of wavelengths and the third step fully optimizes for the maximum insertion loss), R^{max} is set to 2, corner bending is turned off and no path hints are given. Multiple tests are made for each value of N_m because the results can vary substantially with the actual communication matrix chosen. To get a general trend over all values of N_m, the multiple values over each set of tests for each N_m are averaged[1]. Finally, all tests in this section use the technology parameters presented in Table 3.1.

7.2.1 General results and corner bending comparison

The purpose of this test is to understand how the results for the centralized grid router and the corresponding solve time change with the increase in number of messages and use of corner bending. For this analysis the 8 node centralized grid router was solved twice, one with corner bending turned on and another with it turned off. The results are presented in Figure 7.2.

The following conclusions can be drawn:

- Dependency of solve time with the number of messages is somewhere between $O(n^c)$ and $O(c^n)$, using the standard *Big O* notation where n is the size of the problem and c is some constant, which is to be expected given the combinatorial nature of this problem.

- The maximum insertion loss changes very little above a minimum number of messages. This is very useful to know, because it is very easy to calculate *a priori* an estimate of the maximum insertion loss for a centralized grid template based on the worst possible path through it and, given this insertion loss graph, this estimate is very likely to come close to the actual result for a wide spectrum of number of messages.

- There is a clear linear relationship between the number of messages in the communication matrix and the minimum number of wavelengths and MRRs required. This shows the considerable amount of resources that can be removed from typical logical topologies such as the ones presented in Section 2.5.3 when given non-complete communication matrices.

[1] The combinatorial space for the choice of messages (for example, 28 messages out of 56) is enormous. For obvious reasons, only a very small portion of this space was randomly selected and tested. Some noise in the results is to be expected, but the trends are nonetheless very clear.

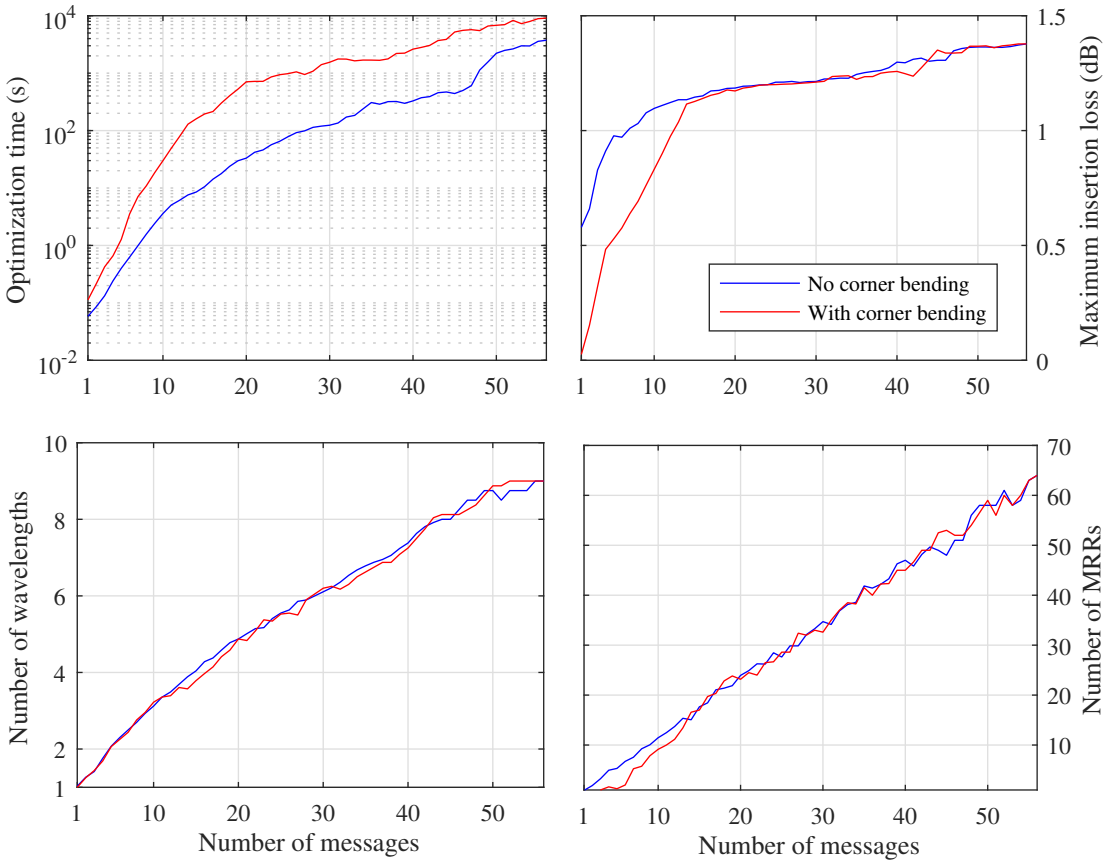


Figure 7.2: Solve time and optimization results vs number of messages in the communication matrix for an 8 node centralized grid router – baseline results and comparison with corner bending.

- Using corner bending takes longer because it considerably expands the combinatorial space. However, it also significantly reduces the maximum insertion loss when using up to 14 messages. Above 14 messages results indicate corner bending is not used. Thus, for the centralized grid router, only sparse[2] matrices will take advantage of this feature.

7.2.2 Time improvement with 3-step optimization

The purpose of this test is to assess the solve time benefit in using the 3-step optimization explained in Section 5.6. For this test the 8 node centralized grid router was solved twice. The first time used the 3-step optimization: first step provided first feasible solution, second step fully optimized for number of wavelengths, and third step fully optimized for maximum insertion loss. For the second run, however, the model was solved only once with $100 \times nwl + 1 \times maxil$ as the minimization function. Given the magnitude of the values of *nwl* and *maxil*, this function mimics the hierarchical optimization given by the 3-step procedure and ensures a fair comparison.

As can be seen in Figure 7.3, using $100 \times nwl + 1 \times maxil$ does not alter the quality of the results, meaning a fair comparison between the solve times for both test runs is in fact possible.

[2]Based on these results, up to about $14/56 = 25\%$ of a full communication matrix.

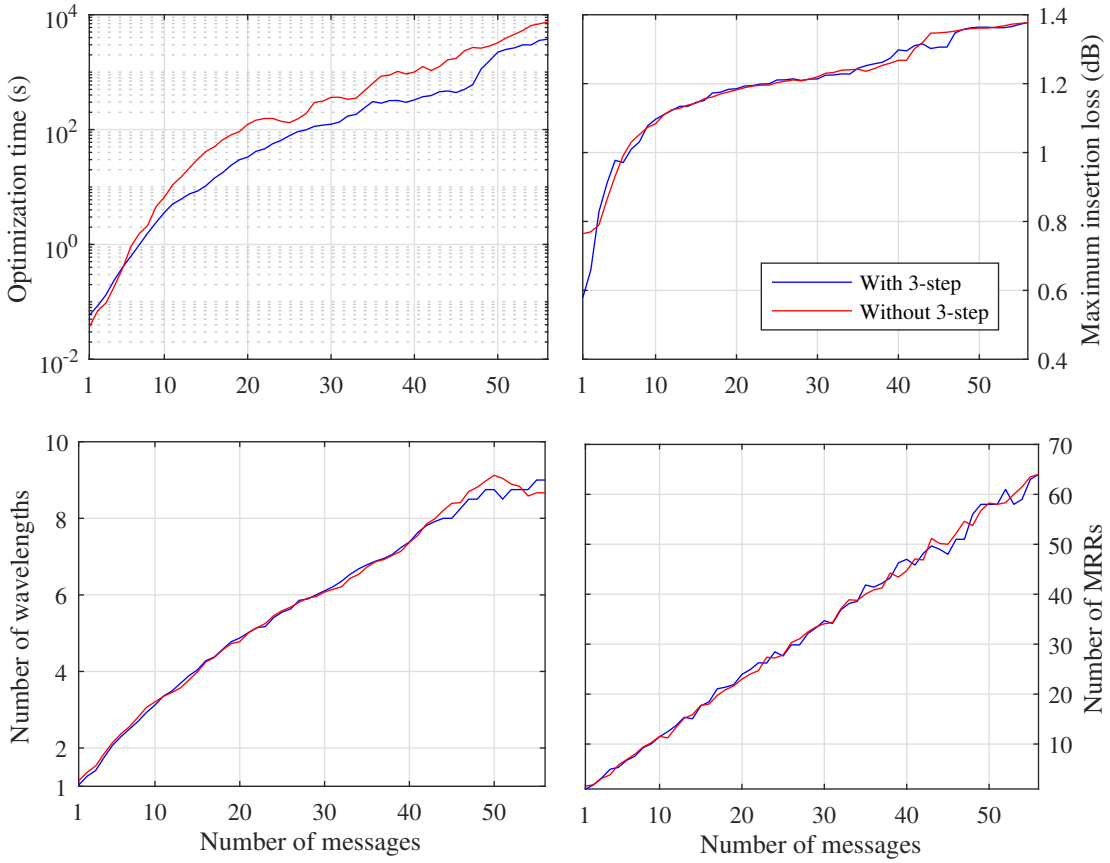


Figure 7.3: Solve time and optimization results vs number of messages in the communication matrix for an 8 node centralized grid router – assessing time benefits of 3-step optimization.

This comparison, however, comes out completely in favor of the 3-step optimization procedure. On average, using this procedure is **2.5×** faster.

7.2.3 Time improvement with R^{max} heuristic

The purpose of this test is to assess the time benefit of using the R^{max} heuristic explained in Section 5.4.4.2. This heuristic has already been proven to have a very positive impact when used for the custom template in the comparison with Proton+ (see Section 7.1.2), but here a more in-depth analysis is made. For this test the 8 node centralized grid router was solved twice: the first time used $R^{max} = 2$ and the second time did not use this heuristic at all.

As can be seen in Figure 7.4, this heuristic does not lower the quality of the results for the centralized grid template, as argued in Section 6.2.1.1. Nevertheless, using this heuristic is **4.5×** faster on average. Therefore, it proves to be an adequate and useful heuristic.

7.2.4 Time improvement with path hints

As explained in Section 6.2.1.1, some messages in centralized grid routers have very clear paths. This information can be given to the solver before starting the optimization. The purpose of this

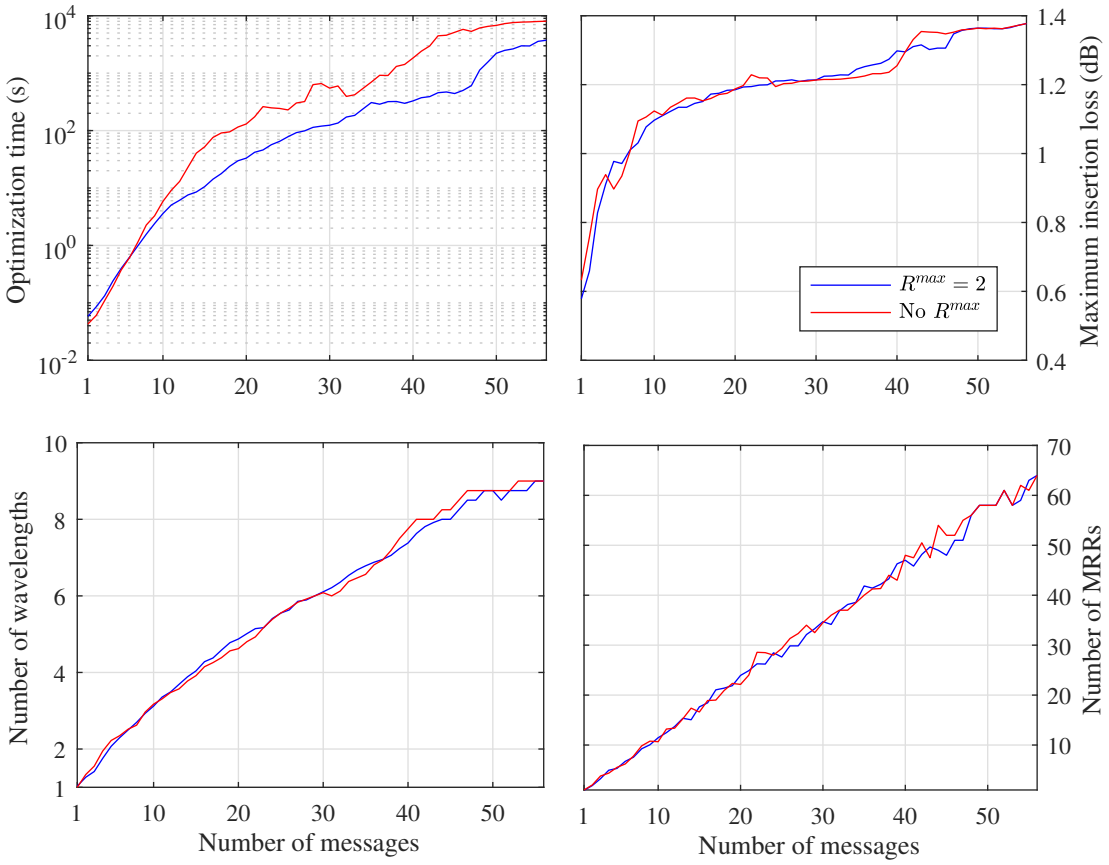


Figure 7.4: Solve time and optimization results vs number of messages in the communication matrix for an 8 node centralized grid router – assessing time benefits of R^{max} heuristic.

test is to measure how much time improvement can be accomplished with this technique.

To make this measurement the 8 node centralized grid router was solved three times. The first time no hints were given; the second time hints were given for all 0-MRR and 1-MRR messages on the communication matrix; the third time, instead of giving hints, all 0-MRR and 1-MRR messages were forced to use those plausible paths.

Results are presented in Figure 7.5. Curiously, just providing path hints does not produce a measurable decrease in total solve times. Not shown in these graphs, however, is that it does produce a considerable speed-up in finding the optimal solution (just not in proving its optimality).

For example, take the 56 message case. Without path hints, it may take about 1500 seconds to find a solution with 9 wavelengths, and then an additional 500 seconds to prove the optimality of that solution. With path hints it takes only 23 seconds to find the 9 wavelength solution. The remaining time is then spent only on proving its optimality. A reasonable explanation for why proving optimality is equally hard in both cases is because optimality can only be proven once a sufficient portion of the decision tree for the problem has been explored, and this tree is the same in both cases[3].

[3] An analogy for this would be an explorer dropped on a mountainous island tasked with finding its highest peak. Even if the explorer has the good fortune of starting on the tallest mountain, he still has to go look at all the other

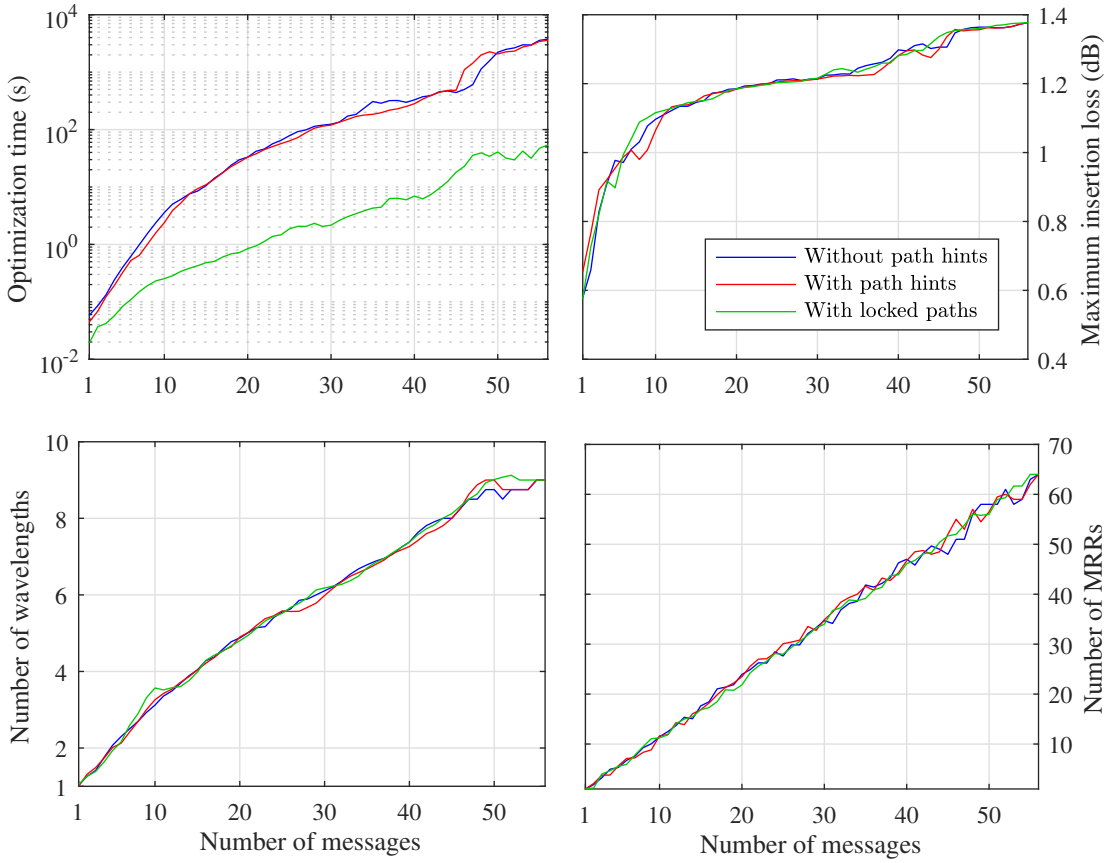


Figure 7.5: Solve time and optimization results vs number of messages in the communication matrix for an 8 node centralized grid router – assessing time benefits of path hints.

The key conclusion here is that there are few shortcuts for making the solver faster at proving optimality[4], but there are still many possibilities of getting to very good solutions (i.e., probably optimal solutions, even though no proof of optimality is available) faster, and path hints definitely succeed in this.

On the other hand, forcing all 0-MRR and 1-MRR messages to have the paths explained in Section 6.2.1.1 makes the entire optimization process (including proving optimality) on average **16.2×** faster. What is the most surprising, however, is that this extreme restriction on the possible paths of some messages does not in any way deteriorate the optimal result. This proves indubitably that 0-MRR and 1-MRR messages do in fact always take those paths in optimal solutions for centralized grid routers when corner bending is turned off.

7.2.5 Maximum bound progression during optimization

This test is designed to measure how fast the best available feasible solution improves during the optimization process.

mountains to be sure.

[4]Without changing the solution space considered by the model, for example, by using the R^{max} heuristic.

Due to the way MIP models are solved, the solver always knows the theoretical best value for the objective function. This best bound is improved during the optimization just like the best feasible solution[5]. Their difference gives an estimate of how far the best feasible solution is from optimality, i.e., the maximum solution error.

In this test an 8 node centralized grid was solved with multiple random sets of 16, 32 and 48 messages. The 3-step optimization procedure was used and the improvements over time in both the optimization for the number of wavelengths and the optimization for the maximum insertion loss were recorded. Results are given in Figure 7.6. Time is given as a percentage of total solve time for the optimization of the corresponding step.

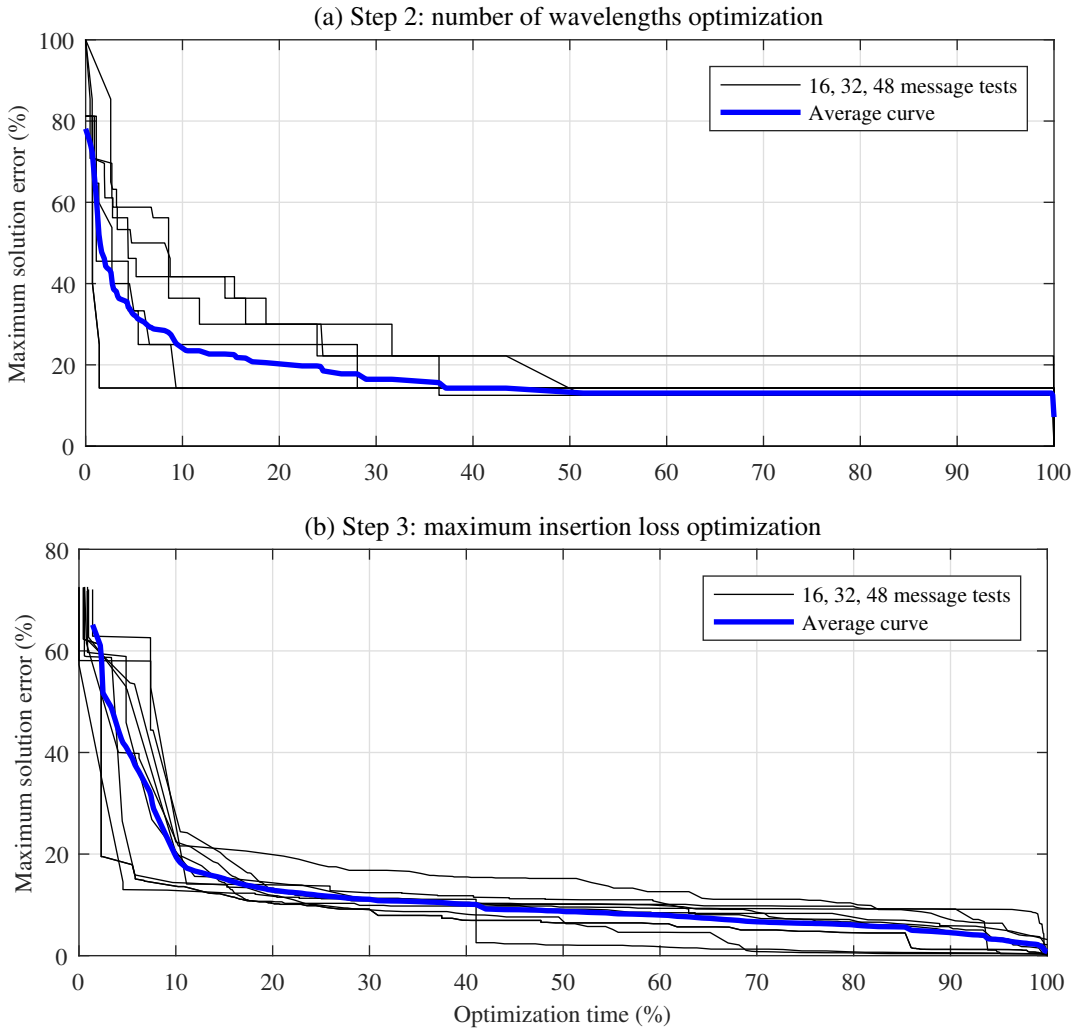


Figure 7.6: Maximum solution error improvement during optimization for an 8 node centralized grid router.

It is abundantly clear that the solver achieves most of its improvement on the feasible solution very quickly. In fact, on average, it takes only one tenth to one fifth of the total optimization time

[5]On a minimization problem the best bound increases during the optimization. When the best bound and the feasible solution meet, by definition of "best bound", the feasible solution is proven to be optimal and the optimization stops.

to get a feasible solution with a maximum error below 20%. So, if a certain problem takes 2000 seconds to fully optimize, a designer willing to sacrifice 10% to 20% of the optimal solution is extremely likely to "solve" the model in just 200 seconds.

7.2.6 Concluding remarks

In this section multiple tests were performed to better understand the performance of the solver and what kind of solutions can be expected from the centralized grid router. Based on the results, general guidelines for an optimization workflow for WRONoCs given any node positions and communication matrix can be established:

- First, try the centralized grid router. Use 3-step optimization, turn off corner bending, set $R^{max} = 2$ and lock the paths for all 0-MRR and 1-MRR messages. The optimization should run extremely fast and the results should already be better than the state of the art, as proven in Section 7.1.2.
- If those results are not yet satisfactory, then either **1)** keep the centralized grid router, remove path locks and turn on corner bending if the communication matrix is sparse or **2)** try other physical layout templates. For option 2, R^{max} and corner bending should be set on a case-by-case basis and, if available, path hints are also beneficial[6].

In all of these cases always be on the lookout for when the solver has found a good-enough solution. This can also substantially reduce total solve times.

7.3 Example of optimized result for 16 nodes, 22 messages

Finally, an example result of a centralized grid router is shown. The communication matrix for this WRONoC was taken from Antti Alhonen *et al.* [41, Figure 10]. That particular NoC has 16 nodes and 22 messages. To connect 16 nodes a centralized grid router with 8 ports (4 input-output node connections) on each side is used.

For this optimization corner bending was turned on, R^{max} was set to 2, the second step fully optimized the number of wavelengths and the third step optimized for the maximum insertion loss. No path hints were given and the technology parameters from Table 3.1 were used. The resulting router is shown in Figure 7.7. The optimization took 5.5 hours, the required number of wavelengths was 7 and the resulting maximum insertion loss was 1.19 dB. Note, however, that after 30 minutes a result for 7 wavelengths and 1.338 dB was already available. The produced router is highly intricate due to corner bending and a sparse communication matrix, which together allow for more aggressive optimizations.

[6] Locking the paths of messages might be too much of a gamble in achieving the optimal solution unless the paths of those messages are *extremely* clear.

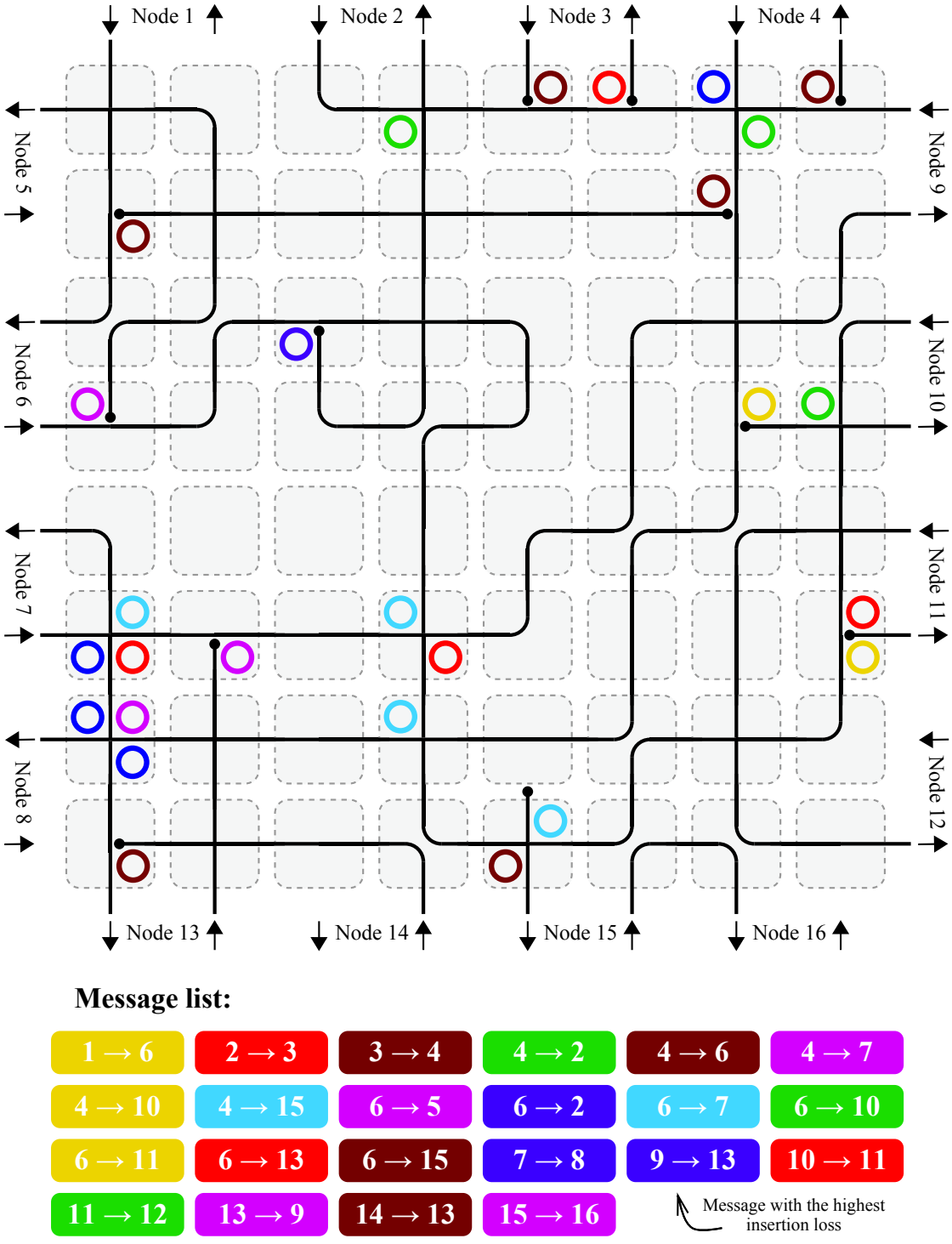


Figure 7.7: Resulting WRONoC design for 16 nodes and 22 messages with corner bending. Color indicates wavelength.

Chapter 8

Conclusion & future work

8.1 Conclusion

As stated at the beginning, the purpose of this research was to improve the state of the art algorithms, procedures and design tools for Wavelength-Routed Optical Networks-on-Chip. More specifically, to combine the optimization of the first two steps in WRONoC design: the logical topology design and the physical layout of the router. This goal was achieved in full. Not only were these two steps combined, the results attainable with this new work easily surpass the current state of the art.

To achieve such results a linear programming model was developed to handle the actual optimization and a design toolchain was built to implement the model. Along the way, multiple ideas such as the use of a physical layout template and the General Routing Unit were presented for the first time. Many accompanying techniques to the core optimization of the programming model were also demonstrated. Finally, multiple tests were performed to characterize the solver performance and the quality of the achieved results.

One final key asset of this work is that the new approach to WRONoC design outlined here was conceptualized with future improvements in mind. In other words, this new approach is not only inspired by the experience gained in previous work, but it also benefits from having a comprehensive view over the major design tasks from WRONoCs right from the onset. It is this broad view of the entire WRONoC design problem that allows other design tasks not considered in this work, such as the optimization of the OPDN, to be added later with little effort.

8.2 Future work

The novelty and uniqueness of this WRONoCs design approach as compared to the state of the art has opened the door to many new optimization possibilities. In the future some of the following areas can be explored within the framework already defined:

Solve times. More effort can be spent on improving solve times. For example, other helper algorithms/heuristics can be designed to give path hints for messages or good starting solutions

to the MIP model for other layout templates. New versions of the 3-step optimization procedure might also result in a faster solver. For big templates an iterative approach to solving the MIP model (where each iteration increases the portion of the template that is considered in the optimization) might prove to be very time efficient.

Optical Power Distribution Network. It is possible to include in the WRONoC layout template a template for the OPDN as well. Then, the MIP model can be modified to optimize both the physical layout of the router and the layout of the OPDN. This will become useful in cases where the designed templates do not follow the design rule outlined in Section 6.2.2.

GRU designs. More GRU designs can be added to the model and their trade-off between solve times and improvement on results characterized.

Layout templates. More layout templates (such as ring-based templates) can be explored. There is a potential for the development of a "library" of layout templates, each with their own characteristics, strong points and weaknesses. It might also be worth exploring the design of automatic synthesis tools for layout templates.

8.3 Scientific publications

This work spawned a scientific publication, annexed in Appendix C, to be submitted to the 56th Design Automation Conference, whose call for papers opens on November 2018.

Appendix A

Complete WRONoC MIP model

A.1 Constants & Indices

Constants	
$N_{gru}, N_{wg}, N_m, N_{ep},$	Total number of GRUs, waveguide sections, messages, endpoints and
N_λ	wavelengths
L^P, L^C, L^B, L^D, L^T	Values for propagation, crossing, bending, drop and through loss
L_{wg}, L^E_{wg}	Length and extra loss of waveguide section wg
Indices	
$W^T_g, W^B_g, W^L_g, W^R_g$	Waveguide section connected to GRU g to the top, bottom, left and right
W^E_{ep}	Waveguide section connected to endpoint ep
E^S_m, E^R_m	Sending and receiving endpoints for message m

A.2 Variables

Binary

$mwg_{m,wg}$	Message m goes through waveguide section wg
$cl_{g,m}$, $bl_{g,m}$	Message m has crossing/bending loss on GRU g
$tl_{g,p,m}$	Message m has through loss due to MRR p in GRU g
$mwl_{m,\lambda}$	Message m uses wavelength λ
$mwleq_{m_1,m_2}$	Messages m_1 and m_2 use the same wavelength
wlu_{λ}	Wavelength λ used by at least one message
$rum_{g,p,m}$	MRR on GRU g, corner p, used by message m
$ru_{g,p}$	MRR on GRU g, corner p, used by a message
$cb_{g,p}$	Corner p on GRU g is bent
mch_g, mcv_g	GRU g has at least one message going through the center crossing horizontally/vertically

Integer

nwl	Number of used wavelengths

Continuous

mil_m	Insertion loss for message m
$maxilwl_{\lambda}$	Maximum insertion loss over all messages using wavelength λ
$maxil$	Maximum insertion loss over all messages

Index $p \in \mathbb{P}$, $\mathbb{P} = \{TL : \text{Top-Left}, TR : \text{Top-Right}, BL : \text{Bottom-Left}, BR : \text{Bottom-Right}\}$.

A.3 Constraints

$$mwg_{m,W^E_{E^S_m}} = 1 \quad mwg_{m,W^E_{E^R_m}} = 1 \qquad \forall m = 1...N_m \tag{A.1}$$

$$\begin{aligned}
&mwg_{m,W^T_g} \leqslant mwg_{m,W^B_g} + mwg_{m,W^L_g} + mwg_{m,W^R_g} \\
&mwg_{m,W^B_g} \leqslant mwg_{m,W^T_g} + mwg_{m,W^L_g} + mwg_{m,W^R_g} \\
&mwg_{m,W^L_g} \leqslant mwg_{m,W^T_g} + mwg_{m,W^B_g} + mwg_{m,W^R_g} \\
&mwg_{m,W^R_g} \leqslant mwg_{m,W^T_g} + mwg_{m,W^B_g} + mwg_{m,W^L_g} \\
&mwg_{m,W^T_g} + mwg_{m,W^B_g} + mwg_{m,W^L_g} + mwg_{m,W^R_g} \leqslant 2 \\
&\quad \forall m = 1...N_m \\
&\quad \forall g = 1...N_{gru}
\end{aligned} \tag{A.2}$$

$$mwg_{m,W^E_{ep}} = 0 \qquad \forall ep = 1...N_{ep} \setminus \{E^S_m, E^R_m\} \qquad \forall m = 1...N_m \tag{A.3}$$

$$\sum_{\lambda=1}^{N_\lambda} mwl_{m,\lambda} = 1 \qquad \forall m = 1...N_m \tag{A.4}$$

$$mwl_{m_1,\lambda} + mwl_{m_2,\lambda} - 1 \leqslant mwleq_{m_1,m_2} \qquad \forall \lambda = 1...N_\lambda \quad \forall m_1, m_2 = 1...N_m : m_1 < m_2 \tag{A.5}$$

$$mwg_{m_1,wg} + mwg_{m_2,wg} + mwleq_{m_1,m_2} \leqslant 2 \qquad \forall wg = 1...N_{wg} \quad \forall m_1, m_2 = 1...N_m : m_1 < m_2 \tag{A.6}$$

$$ru_{g,p} = \sum_{m=1}^{N_m} rum_{g,p,m} \qquad \forall p \in \mathbb{P} \quad \forall g = 1...N_{gru} \tag{A.7}$$

$$\begin{aligned} mwg_{m,W_g^T} + mwg_{m,W_g^L} - 1 &\leqslant rum_{g,TL,m} + rum_{g,BR,m} + cb_{g,TL} \\ mwg_{m,W_g^T} + mwg_{m,W_g^R} - 1 &\leqslant rum_{g,TR,m} + rum_{g,BL,m} + cb_{g,TR} \\ mwg_{m,W_g^B} + mwg_{m,W_g^L} - 1 &\leqslant rum_{g,BL,m} + rum_{g,TR,m} + cb_{g,BL} \\ mwg_{m,W_g^B} + mwg_{m,W_g^R} - 1 &\leqslant rum_{g,BR,m} + rum_{g,TL,m} + cb_{g,BR} \\ \forall m &= 1...N_m \\ \forall g &= 1...N_{gru} \end{aligned} \tag{A.8}$$

$$cb_{g,p_1} + ru_{g,p_2} \leqslant 1 \qquad \forall p_1, p_2 \in \mathbb{P} \quad \forall g = 1...N_{gru} \tag{A.9}$$

$$\begin{aligned} cb_{g,TL} + cb_{g,TR} &\leqslant 1 \qquad cb_{g,TR} + cb_{g,BR} \leqslant 1 \\ cb_{g,TL} + cb_{g,BL} &\leqslant 1 \qquad cb_{g,BL} + cb_{g,BR} \leqslant 1 \\ \forall g &= 1...N_{gru} \end{aligned} \tag{A.10}$$

$$
\begin{aligned}
&cb_{g,TL} + mwg_{m,W_g^T} - 1 \leqslant mwg_{m,W_g^L} \\
&cb_{g,TL} + mwg_{m,W_g^L} - 1 \leqslant mwg_{m,W_g^T} \\
&cb_{g,TR} + mwg_{m,W_g^T} - 1 \leqslant mwg_{m,W_g^R} \\
&cb_{g,TR} + mwg_{m,W_g^R} - 1 \leqslant mwg_{m,W_g^T} \\
&cb_{g,BL} + mwg_{m,W_g^T} - 1 \leqslant mwg_{m,W_g^L} \\
&cb_{g,BL} + mwg_{m,W_g^L} - 1 \leqslant mwg_{m,W_g^T} \\
&cb_{g,BR} + mwg_{m,W_g^B} - 1 \leqslant mwg_{m,W_g^R} \\
&cb_{g,BR} + mwg_{m,W_g^R} - 1 \leqslant mwg_{m,W_g^B} \\
&\quad \forall m = 1...N_m \\
&\quad \forall g = 1...N_{gru}
\end{aligned}
\tag{A.11}
$$

$$
\begin{aligned}
&mwg_{m,W_g^T} + mwg_{m,W_g^L} + cb_{g,TL} - 2 \leqslant bl_{g,m} \\
&mwg_{m,W_g^T} + mwg_{m,W_g^R} + cb_{g,TR} - 2 \leqslant bl_{g,m} \\
&mwg_{m,W_g^B} + mwg_{m,W_g^L} + cb_{g,BL} - 2 \leqslant bl_{g,m} \\
&mwg_{m,W_g^B} + mwg_{m,W_g^R} + cb_{g,BR} - 2 \leqslant bl_{g,m} \\
&\quad \forall m = 1...N_m, \\
&\quad \forall g = 1...N_{gru}
\end{aligned}
\tag{A.12}
$$

$$
\begin{aligned}
&mwg_{m,W_g^L} + mwg_{m,W_g^R} + ru_{g,p} - 2 \leqslant tl_{g,p,m} \\
&mwg_{m,W_g^T} + mwg_{m,W_g^B} + ru_{g,p} - 2 \leqslant tl_{g,p,m} \\
&\quad \forall m = 1...N_m, \\
&\quad \forall p \in \mathbb{P}, \\
&\quad \forall g = 1...N_{gru}
\end{aligned}
\tag{A.13}
$$

$$
\begin{aligned}
&mwg_{m,W_g^L} + mwg_{m,W_g^R} - 1 \leqslant mch_g \\
&mwg_{m,W_g^T} + mwg_{m,W_g^B} - 1 \leqslant mch_g \\
&\quad \forall m = 1...N_m, \\
&\quad \forall g = 1...N_{gru}
\end{aligned}
\tag{A.14}
$$

$$
\begin{aligned}
&mwg_{m,W_g^T} + mwg_{m,W_g^L} + rum_{g,BR,m} - 2 \leqslant mch_g \\
&mwg_{m,W_g^T} + mwg_{m,W_g^L} + rum_{g,BR,m} - 2 \leqslant mcv_g \\
&mwg_{m,W_g^T} + mwg_{m,W_g^R} + rum_{g,BL,m} - 2 \leqslant mch_g \\
&mwg_{m,W_g^T} + mwg_{m,W_g^R} + rum_{g,BL,m} - 2 \leqslant mcv_g \\
&mwg_{m,W_g^B} + mwg_{m,W_g^L} + rum_{g,TR,m} - 2 \leqslant mch_g \\
&mwg_{m,W_g^B} + mwg_{m,W_g^L} + rum_{g,TR,m} - 2 \leqslant mcv_g \\
&mwg_{m,W_g^B} + mwg_{m,W_g^R} + rum_{g,TL,m} - 2 \leqslant mch_g \\
&mwg_{m,W_g^B} + mwg_{m,W_g^R} + rum_{g,TL,m} - 2 \leqslant mcv_g \\
&\quad \forall m = 1...N_m, \\
&\quad \forall g = 1...N_{gru}
\end{aligned}
\tag{A.15}
$$

$$
\begin{aligned}
&mwg_{m,W_g^L} + mwg_{m,W_g^R} + mcv_g - 2 \leqslant cl_{g,m} \\
&mwg_{m,W_g^T} + mwg_{m,W_g^B} + mch_g - 2 \leqslant cl_{g,m} \\
&\quad \forall m = 1...N_m, \\
&\quad \forall g = 1...N_{gru}
\end{aligned}
\tag{A.16}
$$

$$
\begin{aligned}
mil_m = &\sum_{i=1}^{N_{wg}} (L^P * L_i + L_i^E) * mwg_{m,i} \\
&+ \sum_{g=1}^{N_{gru}} (L^C * cl_{g,m} + L^B * bl_{g,m}) \\
&+ \sum_{g=1}^{N_{gru}} \sum_{p \in \mathbb{P}} (L^T * tl_{g,p,m} + L^D * rum_{g,p,m}) \qquad \forall m = 1...N_m
\end{aligned}
\tag{A.17}
$$

$$
\begin{aligned}
wlu_\lambda \geqslant mwl_{m,\lambda} \qquad &\forall m = 1...N_m, \\
&\forall \lambda = 1...N_\lambda
\end{aligned}
\tag{A.18}
$$

$$
nwl = \sum_{\lambda=1}^{N_\lambda} wlu_\lambda \tag{A.19}
$$

$$
maxil \geqslant mil_m \qquad \forall m = 1...N_m \tag{A.20}
$$

$$maxilwl_{\lambda} \geqslant mil_{m} - M * (1 - mwl_{m,\lambda}) \qquad \forall m = 1...N_m \quad \forall \lambda = 1...N_{\lambda} \quad M \to +\infty \tag{A.21}$$

$$\sum_{g=1}^{N_{gru}} \sum_{p \in \mathbb{P}} rum_{g,p} \leqslant R^{max} \qquad \forall m = 1...N_m \tag{A.22}$$

$$mwl_{m,\lambda} = 0 \qquad \forall \lambda = (m+1)...N_{\lambda} \quad \forall m = 1...N_m \tag{A.23}$$

$$cl_{g,m} + bl_{g,m} + \sum_{p \in \mathbb{P}} rum_{g,p,m} \leqslant 1 \qquad \forall g = 1...N_{gru}, \quad \forall m = 1...N_m \tag{A.24}$$

A.4 Optimization function

$$\min \quad \alpha_1 * nwl + \alpha_2 * maxil + \alpha_3 * \sum_{\lambda=1}^{N_{\lambda}} maxilwl_{\lambda} + \alpha_4 * \sum_{m=1}^{N_m} mil_m + \alpha_5 * \sum_{g=1}^{N_{gru}} \sum_{p \in \mathbb{P}} ru_{g,p} \tag{A.25}$$

Appendix B

Lower triangular matrix proof

The goal is to prove that any matrix that has exactly one nonzero element in each row can be converted into a lower triangular matrix by swapping columns. Start by looking only at the first row of the matrix. Swap columns such that the nonzero value for that row is on the first column:

$$\begin{bmatrix} 0 & \dots & 1 & \dots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots \end{bmatrix} \equiv \begin{bmatrix} 1 & 0 & \dots & \dots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots \end{bmatrix}$$

Notice that row 1 now satisfies the lower triangular matrix requirement. Now look at the second row. If the nonzero value is on column 1, do nothing (in that case, the requirement is already met). Otherwise, swap columns such that the nonzero value for that row is on the second column:

$$\begin{bmatrix} 1 & 0 & \dots & \dots & 0 \\ 0 & \dots & 1 & \dots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots \end{bmatrix} \equiv \begin{bmatrix} 1 & 0 & \dots & \dots & 0 \\ 0 & 1 & 0 & \dots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots \end{bmatrix}$$

Repeat the process until all rows have been checked. The result is a lower triangular matrix.

Appendix C

Scientific publication

Combining logical topology and physical layout optimization for Wavelength-Routed ONoCs

Alexandre Truppel[♯], Tsun-Ming Tseng[†], Davide Bertozzi[⋆], José Carlos Alves[♯], and Ulf Schlichtmann[†]
{up201303442, jca}@fe.up.pt, davide.bertozzi@unife.it, {tsun-ming.tseng, ulf.schlichtmann}@tum.de
[♯] Faculdade de Engenharia, Universidade do Porto, rua Dr. Roberto Frias, 4200-465 Porto, Portugal
[†] Chair of Electronic Design Automation, Technical University of Munich, Arcisstraße 21, 80333 München, Germany
[⋆] University of Ferrara, Via Saragat 1, 44121 Ferrara, Italy

ABSTRACT

Optical Networks-on-Chip are a promising solution for high-performance multi-core integration with better latency and bandwidth than traditional Electrical NoCs. Wavelength-routed ONoCs offer yet additional performance guarantees. However, WRONoC design presents new EDA challenges which have not yet been fully addressed. So far, most topology analysis is abstract, i.e., overlooks layout concerns, while for layout the tools available perform Place & Route (P&R) but no topology optimization. Thus, a need arises for a novel optimization method combining both aspects of WRONoC design. In this paper such a method is laid out. When compared to the state-of-the-art design procedure, results show a remarkable 50% reduction in maximum insertion loss.

1. INTRODUCTION

Optical Networks-on-Chip (ONoCs) have been proposed as a solution for the ever-increasing integration requirements of large System-on-Chip designs. Compared to traditional Electrical Networks-on-Chip, ONoCs present not only lower dynamic power consumption but also extremely low signal delay and higher bandwidth [1].

The use of light as opposed to electrical signals to send information between network nodes requires the following four main components on the optical routing plane: 1) *modulators* to convert electrical signals into optical signals at every node (electrical-optical interface) of the optical network, 2) *demodulators* to do the opposite, 3) *waveguides* acting as optical wires and 4) *optical routing elements* to transfer optical signals between waveguides [2].

ONoCs can be organized into two main categories: 1) *active networks* [3–5] and 2) *passive networks*. Active networks require a control layer for routing. Passive networks use routing elements which resonate with different frequencies such that a message is passively routed according to the wavelength of the carrier light. Hence, a message's path is completely defined, at design time, by its origin and wavelength alone. This eliminates network delay resulting from path setup and dynamic power consumption required for the extra control layer. Thus, passive ONoCs are also termed Wavelength-Routed ONoCs (WRONoCs) [6].

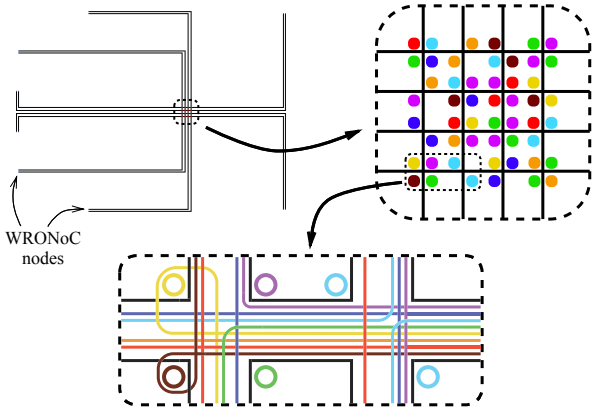


Figure 1: Final design of a WRONoC router for 8 nodes given by our method. A portion of some message paths is shown (color indicates wavelength).

Multiple light sources of different wavelengths can be used to transmit separate information streams on the same waveguide without interference (wavelength-division multiplexing). This enables conflict-free communications with increased bandwidth. The only requirement is to make sure at design time that no two messages with the same wavelength are allowed to share the same waveguides.

The optical switching element in ONoCs is the Micro-Ring Resonator (MRR). It has a circular silicon structure whose radius defines the resonance frequency. A light signal with a certain wavelength propagating on a waveguide close to a MRR with a matching resonance frequency will be coupled to the MRR and moved onto another waveguide also close to that MRR [7].

The design of a WRONoC router is an optimization process with *two aspects* to consider: the logical topology and the physical layout of the router. The former assigns a wavelength to each message and each MRR and also connects the nodes through waveguides and MRRs such that the communication matrix, which specifies the communication requirements between nodes, is fulfilled. The latter optimally places and routes those elements on the optical plane while considering the physical positions of the nodes and constraints related to the physical placement of the waveguides.

So far both aspects have only been considered separately or with restrictions. Various works have presented specific topologies with few concerns about their layout [2, 8, 9]. Ramini et al. [10] present a topology designed in tandem with placement constraints, yet it results from a manual op-

ACM ISBN 978-1-4503-2138-9.
DOI: 10.1145/1235

timization effort for one set of node positions. Ortín-Obón et al. [1] take into consideration physical constraints, but analyze only the ring topology. Few attempt to optimize for non-complete communication matrices [11]. P&R tools to optimize the second aspect have been developed [12,13], but all take a topology as input, forcing the designer to choose the topology beforehand.

However, neither aspect can be considered in isolation, as each influences the other [9,10,13]. During generation of the logical topology we are unable to accurately predict important physical characteristics, e.g. the number of waveguide crossings, of the final design after P&R. Furthermore, during P&R, if the logical topology has already been chosen and fixed, any subsequent optimization is being done only around a local minimum of the solution space.

Ideally, a design tool would take as inputs the communication matrix and the physical positions of the nodes and, by working on both aspects simultaneously, produce a fully-optimized fully-custom logical topology and matching physical layout. [9] In reality, the problem space of such an optimization is discouragingly vast for any but the simplest cases. Thus, in this paper we propose and solve a constrained version of the complete problem. In this version, a physical layout template is also given as an input to the optimization. The template mainly consists of MRR placeholders and waveguides already placed and routed on the optical plane, and connects all nodes.

A Mixed Integer Programming (MIP) model is presented to tackle the constrained optimization problem. Then, a 3-step algorithm to efficiently solve the model is proposed. Finally, three layout templates are presented and tested on test cases from the state-of-the-art P&R Proton+ tool [13]. One of the final results is shown in Figure 1.

2. WRONoC DESIGN PROBLEM

We formally define the optimization problem for the design of WRONoC routers as follows:

Input data:

- Communication matrix: a square binary matrix $CM_{i,j} \in \mathbb{R}^{N*N}$ with N equal to the number of nodes and where $CM_{i,j} = 1$ if node i sends a message to node j.
- Physical positions of the modulators and demodulators of each node on the optical plane.
- Technology parameters: power loss values.

Output data:

- Wavelength of each message and MRR.
- Placement of each MRR.
- Routing of each waveguide.

Minimization objectives: their choice depends on the technology and the needs of the designer. We consider **1)** number of wavelengths, **2)** message insertion loss and **3)** number of MRRs, as in previous publications [1, 7–10, 13]. In our method the weighting coefficient for each objective can be freely adjusted to meet different designer demands.

Message insertion loss is the sum of seven types of losses: **1)** crossing loss, **2)** dropping loss, **3)** through loss, **4)** bending loss, **5)** propagation loss, **6)** modulator loss and **7)** demodulator loss [13,14]. We consider all except the last two, which are constant and equal for all messages and thus can be ignored from an optimization perspective.

3. PHYSICAL LAYOUT TEMPLATE

We consider a constrained version of the complete problem, where an extra input is required. This input, called a **physical layout *template***, consists of a collection of WRONoC router elements (modulators, demodulators, waveguides and MRR placeholders) already placed and routed on the optical plane.

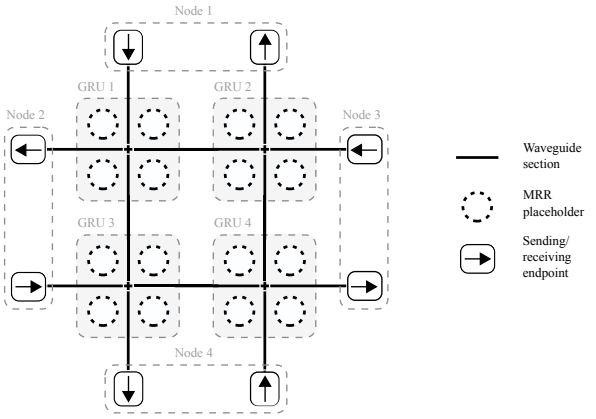


Figure 2: Generalizing the 4x4 GWOR topology [8] using endpoints, GRUs and waveguide sections.

The role of the solver with this new input is to optimally route the messages defined in the communication matrix through the template and to activate the necessary routing features for the chosen paths.

This way we significantly reduce the complexity of the complete problem while still improving upon the state-of-the-art solutions. Nevertheless, this template does not need to be intricate or sophisticated. In fact, the intuitive knowledge of the designer about the structure of the router to be created is more than enough to provide a good template.

3.1 Template elements

We model layout templates with three layout elements. Together they allow for the design of any WRONoC topology (an example is shown in Figure 2).

Endpoints represent modulators and demodulators. They are placed wherever the (de)modulators for each node are and connect to one waveguide section.

General Routing Units (GRUs) are elements that connect to multiple waveguide sections (the *edges* of the GRU) and contain MRR placeholders, to be populated by the solver as needed. They are the routing building blocks of the template and are described further in section 3.2.

Waveguide sections connect two GRUs or a GRU and an endpoint. Each section has two associated parameters: *length* and *extraloss*. The latter is used to describe templates where messages going through that section incur extra insertion loss (for example, from bends in the waveguide).

3.2 General Routing Unit

Photonic Switching Elements (PSEs) are commonly applied in WRONoC routers [2, 8–10]. For PSEs, MRR locations and wavelengths are explicitly specified and the waveguide structure is fixed.

GRUs are the routing building blocks for the proposed layout template and, in contrast to PSEs, GRUs are not inherently constrained to a specific internal structure. Instead, only MRR placeholders are predefined in a GRU. Thus, different MRR placement and wavelength configurations can happen for each GRU, as well as different edge connection arrangements. This provides more flexibility in the resulting WRONoC design.

3.2.1 Structure

Figure 3(a) shows the structure of a GRU: the four waveguide sections form a crossing where any of the four corners

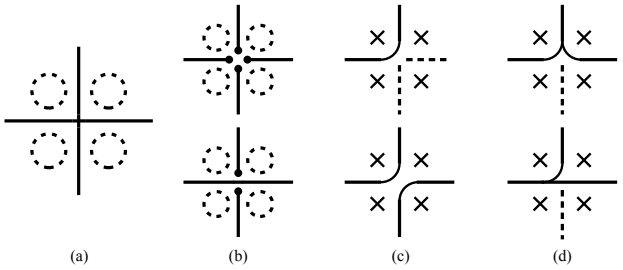


Figure 3: Internal structure of a GRU. (a) 4 MRR placeholders and a crossing. (b) Avoiding the crossing, when possible (c) Valid corner bending states. (d) Invalid corner bending states.

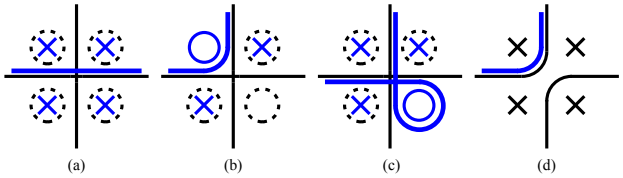


Figure 4: Routing possibilities on a GRU. (a) Direct path. (b)(c) Routing through a MRR. (d) Routing through a bend.

on that crossing can have a MRR. Sometimes the crossing can be avoided, leading to the variations in Figure 3(b).

We also consider an additional structure variation called *corner bending*. When active, the GRU contains no MRRs and some corners may be replaced by a bend between the two edges in that corner, as in Figure 3(c).

Note that two corners connected to the same edge of a GRU *cannot* be both bent. Therefore, if two edges are connected through a corner bend, the other two edges must be bent through the opposite corner if they have messages going through. Figure 3(d) shows two invalid configurations.

This extra variation proves useful for sparser templates (low ratio of the number of messages to the number of MRR positions), or in cases where multiple messages must be routed through the same corner.

3.2.2 *Routing*

Figure 4 shows the routing possibilities through a GRU. If no MRRs of the same wavelength as the message are present and corner bending is not activated, the message will have no direction change, as shown in Figure 4(a).

For wavelength routing, the message can be routed through a MRR with the same wavelength in the closest corner, as shown in Figure 4(b), or in the opposite corner, as shown in Figure 4(c).

With corner bending, since the two waveguides become connected, all messages in any of the two waveguides are routed through that corner, regardless of wavelength, as shown in Figure 4(d).

A message's path through a GRU is always independent of its direction, i.e., all routing features are bidirectional. Also, the four MRRs on a GRU can have different wavelengths (examples are shown in Figure 1). This allows for intricate multi-message routing capabilities per waveguide crossing which have not yet been optimized to full potential.

3.3 Communication Matrix

Given a layout template, the communication matrix can be translated to a set of messages (one for each nonzero entry), where each message is associated with two endpoints on that template, the sender and the receiver.

Table 1: Model constants & indices

Constants	
N_{gru}, N_{wg}, N_m, N_{ep}, N_λ	Total number of GRUs, waveguide sections, messages, endpoints and wavelengths
L^P, L^C, L^B, L^D, L^T	Values for propagation, crossing, bending, drop and through loss
L_{wg}, L^E_{wg}	Length and extra loss of waveguide section wg
Indices	
W^T_g, W^B_g, W^L_g, W^R_g	Waveguide section connected to GRU g to the top, bottom, left and right
W^E_{ep}	Waveguide section connected to endpoint ep
E^S_m, E^R_m	Sending and receiving endpoints for message m

Table 2: Model variables

Binary	
$mwg_{m,wg}$	Message m goes through waveguide section wg
$cl_{g,m}$, $bl_{g,m}$	Message m has crossing/bending loss on GRU g
$tl_{g,p,m}$	Message m has through loss due to MRR p in GRU g
$mwl_{m,\lambda}$	Message m uses wavelength λ
mwe_{m_1,m_2}	Messages m_1 and m_2 use the same wavelength
wlu_λ	At least one message uses wavelength λ
$rum_{g,p,m}$	MRR on GRU g, corner p, used by message m
$ru_{g,p}$	MRR on GRU g, corner p, used by a message
$cb_{g,p}$	Corner p on GRU g is bent
mch_g, mcv_g	GRU g has at least one message going through the center crossing horizontally/vertically
Integer	
nwl	Number of used wavelengths
Continuous	
mil_m	Insertion loss for message m
$maxil$	Maximum insertion loss over all messages

Index $p \in \mathbb{P}$, $\mathbb{P} = \{TL$: Top-Left, TR : Top-Right, BL : Bottom-Left, BR : Bottom-Right$\}$.

4. MATHEMATICAL MODEL

We solve the constrained problem using a Mixed Integer Programming model. Advantages of MIP models include:

1) A MIP model can give optimal solutions, or at least an upper/lower bound to the optimal value of the optimization function.

2) The same MIP can be used to optimize different objectives, therefore giving the designer more flexibility.

3) MIP models are flexible, so new GRU designs, routing features or other modifications can easily be added.

The model constants and indices are outlined in Table 1. Constants L_{wg}, L^E_{wg} and indices W^*_i collectively describe the physical layout template and indices E^*_m define the communication matrix. Table 2 lists all model variables.

We now specify the constraints and the optimization function (similar constraints for multiple directions or corners and the linearization techniques applied are omitted due to space limitations). Finally, we present some model reduction techniques.

4.1 Constraints

Message routing. A path with the correct beginning and end must be guaranteed for each message. For that we apply the following three sets of constraints:

1) A message must be on the waveguide of the endpoints it is sent from and received by.

$$mwg_{m,W^E_{E^S_m}} = 1 \quad mwg_{m,W^E_{E^R_m}} = 1 \qquad \forall m = 1...N_m$$

2) If an endpoint does *not* send or receive a message, that

message *cannot* be present on its waveguide section.

$$mwg_{m,W^E_{ep}} = 0 \quad \forall ep = 1...N_{ep} \setminus \{E^S_m, E^R_m\}$$
$$\forall m = 1...N_m$$

3) A message is exactly on 0 or 2 edges of a GRU.

$$mwg_{m,W^T_g} + mwg_{m,W^R_g} + mwg_{m,W^B_g} + mwg_{m,W^L_g} \in \{0, 2\}$$
$$\forall m = 1...N_m, g = 1...N_{gru}$$

It is possible for a message to be on all four edges of a GRU, but this was neglected because it appearing on an optimized solution is highly unlikely, and not including it simplifies the model and the problem space. The reason is that a message routing through all 4 edges (enter through edge 1, leave through 2, enter through 3, leave through 4) can also route through 2 edges (enter through 1, leave through 4) with half the loss on that GRU and a shorter path.

Wavelength exclusion. Each waveguide section has at most one message going through it for each wavelength. First, each message must use exactly one wavelength:

$$\sum_{\lambda=1}^{N_\lambda} mwl_{m,\lambda} = 1 \qquad \forall m = 1...N_m$$

Then the value of mwe_{m_1,m_2} is set accordingly:

$$mwl_{m_1,\lambda} \wedge mwl_{m_2,\lambda} \Rightarrow mwe_{m_1,m_2}$$
$$\forall \lambda = 1...N_\lambda$$
$$\forall m_1, m_2 = 1...N_m : m_2 \neq m_1$$

Now enforce exclusivity of wavelengths on all waveguides:

$$mwe_{m_1,m_2} \Rightarrow (mwg_{m_1,wg} + mwg_{m_2,wg} \leqslant 1)$$
$$\forall m_1, m_2 = 1...N_m : m_1 \neq m_2$$
$$\forall wg = 1...N_{wg}$$

Activation of routing features. A path is chosen for each message but, to make that path take effect, constraints are needed to enforce the activation of the routing features responsible for it.

If a message takes the direct path through a GRU, no features need to be turned on. However, if a message is present on adjacent edges of a GRU, then one of the three options from Figure 4(b-d) must be active:

$$mwg_{m,W^T_g} \wedge mwg_{m,W^L_g} \Rightarrow rum_{g,TL,m} \vee rum_{g,BR,m} \vee cb_{g,TL}$$
$$\forall \text{ 4 corners}, m = 1...N_m, g = 1...N_{gru}$$

Each MRR can only be used for one message. The following constraints both set the value of $ru_{g,p}$ and enforce that restriction:

$$ru_{g,p} = \sum_{m=1}^{N_m} rum_{g,p,m} \qquad \forall g = 1...N_{gru}, p \in \mathbb{P}$$

Corner bending.[1] The following three sets of constraints are needed:

1) A GRU cannot have corners bent and MRRs active.

$$cb_{g,p_1} + ru_{g,p_2} \leqslant 1 \qquad \forall p_1, p_2 \in \mathbb{P}, g = 1...N_{gru}$$

2) Corners for the same edge cannot be bent at the same time for the same GRU.

$$cb_{g,TL} + cb_{g,TR} \leqslant 1 \qquad cb_{g,TR} + cb_{g,BR} \leqslant 1$$
$$cb_{g,TL} + cb_{g,BL} \leqslant 1 \qquad cb_{g,BL} + cb_{g,BR} \leqslant 1$$
$$\forall g = 1...N_{gru}$$

[1]This feature can be turned off, if needed, by adding constraints to set all $cb_{g,p}$ variables to zero.

3) If a corner is bent then messages present on one of the edges of that corner must be present on the other.

$$cb_{g,TL} \Rightarrow mwg_{m,W^T_g} = mwg_{m,W^L_g}$$
$$\forall \text{ 4 corners}, m = 1...N_m, g = 1...N_{gru}$$

Crossing loss. A message suffers crossing loss when going through a crossing with a perpendicular waveguide. Two things must happen for a message to have crossing loss on a GRU: **1)** the message must take a direct path through the GRU and **2a)** the perpendicular direct path must be taken by at least one other message *or* **2b)** there must be at least one message taking the path on Figure 4(c). On any other case the crossing on the GRU can be avoided, as exemplified in Figure 3(b), and no crossing loss exists.

First set the values of the variables mch_g and mcv_g:

$$mwg_{m,W^L_g} \wedge mwg_{m,W^R_g} \Rightarrow mch_g$$
$$\forall \text{ 2 directions}, m = 1...N_m, g = 1...N_{gru}$$
$$mwg_{m,W^T_g} \wedge mwg_{m,W^L_g} \wedge rum_{g,BR,m} \Rightarrow mch_g \wedge mcv_g$$
$$\forall \text{ 4 corners}, m = 1...N_m, g = 1...N_{gru}$$

The value of $cl_{g,m}$ follows:

$$mwg_{m,W^T_g} \wedge mwg_{m,W^B_g} \wedge mch_g \Rightarrow cl_{g,m}$$
$$\forall \text{ 2 directions}, m = 1...N_m, g = 1...N_{gru}$$

Through loss: if a message is going through the direct path on a GRU, then it has through loss for each MRR present on that GRU.

$$mwg_{m,W^L_g} \wedge mwg_{m,W^R_g} \wedge ru_{g,p} \Rightarrow tl_{g,p,m}$$
$$\forall \text{ 2 directions}, m = 1...N_m, p \in \mathbb{P}, g = 1...N_{gru}$$

Bending loss: a message has bending loss on a GRU if it routes through a corner that is bent.

$$mwg_{m,W^T_g} \wedge mwg_{m,W^L_g} \wedge cb_{g,TL} \Rightarrow bl_{g,m}$$
$$\forall \text{ 4 corners}, m = 1...N_m, g = 1...N_{gru}$$

Drop loss: proportional to the number of MRRs used by each message.

Propagation loss: proportional to the length of the waveguides the message goes through.

Message insertion loss: the total insertion loss of a message over all waveguides and GRUs is a weighted sum.

$$mil_m = \sum_{i=1}^{N_{wg}} (L^P * L_i + L^E_i) * mwg_{m,i} + L^T * \sum_{g=1}^{N_{gru}} \sum_{p \in \mathbb{P}} tl_{g,p,m}$$
$$+ \sum_{g=1}^{N_{gru}} (L^C * cl_{g,m} + L^B * bl_{g,m} + L^D * \sum_{p \in \mathbb{P}} rum_{g,p,m})$$
$$\forall m = 1...N_m$$

4.2 Objective function

Calculating the number of wavelengths is done with the following constraints:

$$wlu_\lambda \geqslant mwl_{m,\lambda} \qquad \forall m = 1...N_m, \lambda = 1...N_\lambda$$
$$nwl = \sum_{\lambda=1}^{N_\lambda} wlu_\lambda$$

Calculating the maximum insertion loss over all messages is done with the following constraints:

$$maxil \geqslant mil_m \qquad \forall m = 1...N_m$$

Finally, the following objective function is minimized:

$$\alpha_1 * nwl + \alpha_2 * maxil + \alpha_3 * \sum_{m=1}^{N_m} mil_m + \alpha_4 * \sum_{g=1}^{N_{gru}} \sum_{p \in \mathbb{P}} ru_{g,p}$$

where α_i are optimization weights chosen by the designer.

Since the value for the insertion loss of each message is available through the mil_m variables, functions other than the maximum or the sum of the insertion loss can also be added to the model and used for optimization.

4.3 Model reduction techniques

4.3.1 Restrictions on usage of wavelengths

The following set of constraints can be added:

$$mwl_{m,\lambda} = 0 \qquad \forall \lambda = (m+1)...N_\lambda \qquad \forall m = 1...N_m$$

They restrict the possible wavelengths for each message: message 1 uses wavelength 1, message 2 uses wavelengths 1 or 2, etc. This way, some meaningless variations around the same effective solution are removed. The optimal solution, however, is not removed from the solution space.

4.3.2 Restrictions on usage of MRRs

Empirically we find that minimizing the insertion loss favors optimal solutions where messages rarely route through GRU corners using MRRs, i.e., each message uses a low number of MRRs in total. Following this reasoning, constraints can be added to the model that force a maximum number of MRRs per message (R^{max}):

$$\sum_{g=1}^{N_{gru}} \sum_{p \in \mathbb{P}} rum_{g,p} \leqslant R^{max} \qquad \forall m = 1...N_m$$

This reduces the set of paths considered by the solver by removing poor, convoluted paths while keeping the more direct paths between endpoints.

5. PROOF OF FEASIBILITY

It is possible that the chosen layout template cannot satisfy the entire communication matrix (for example, if the template is too small). For those cases, the model above will be unfeasible. Verifying the existence of a solution can be done much faster using a simplified version of the model. For that we consider $N_\lambda = N_m$ and uniquely assign a wavelength to each message by adding these constraints:

$$mwl_{m,\lambda} = 1 \qquad \forall m = 1...N_m, \lambda = m$$
$$mwl_{m,\lambda} = 0 \qquad \forall m = 1...N_m, \lambda \neq m$$

The resulting model can be solved much faster but, if the solver is unable to find a feasible solution for this simplified model, the complete model is also unfeasible.

PROOF. Assume a feasible solution exists. It will have $nwl \leqslant N_m$. From that solution build another where each message uses its own wavelength (thus either maintaining or increasing nwl). Any message that changes its wavelength must also change the wavelength of the MRRs it uses. This is always possible because each MRR routes only one message. Furthermore, the wavelength exclusion rule is always satisfied. Hence, the feasibility of the complete model implies the existence of a solution for the simplified version. □

6. 3-STEP OPTIMIZATION

Section 4 introduced a MIP model that is capable of solving the constrained problem for *any* layout template. Therefore, programming the model as presented on any MIP solver and solving it directly for the chosen minimization objective is enough to obtain the optimal solution. However, due to the nature of the problem, it is possible to slightly alter the optimization process yielding more control and faster results. This leads to our proposed 3-step optimization process, where each step optimizes a slightly different version of the model and produces a solution used at the start of the next step.

In the first step we consider $N_\lambda = N_m$ and apply the feasibility proof from section 5. In this way we can generate the first feasible solution much faster if one exists. It can then be used as a warm start, which decreases optimization times substantially. This has the added bonus of stopping the process as quickly as possible if unfeasible.

In the second step we only minimize the number of wavelengths, for two reasons. Firstly, the designer will most likely want to use less wavelengths than the number of messages. Secondly, because, after completing this step, a feasible solution for a smaller number of wavelengths is then available, so the model can again be simplified by eliminating from it the $N_m - nwl$ unused wavelengths.

The designer might be willing to use more wavelengths than the minimum needed. In that case it is up to the designer to know the maximum acceptable number of wavelengths. The second step can be stopped earlier once a solution is found within that acceptable range.

In the third step we consider the complete model (with the needed amount of wavelengths only) and further optimize the last solution using the chosen function ($maxil$, for example). We have now reached the final solution.

Using this process we can notably simplify the problem space during the optimization. However, because the model reductions are always done within the designer's needs, the optimal solution is never missed.

7. RESULTS

The MIP model and 3-step optimization algorithm are programmed in C++ and make use of Gurobi [15], a MIP solver, on a 2.6 GHz CPU.

We tested our model and optimization procedure against the state-of-the-art Proton+ P&R tool. Most of its result analysis is dedicated to an 8 node test case with 44 messages. We solved the same test case considering the same communication matrix, node placement, die size, crossing size and loss parameters (Proton+ does not consider through loss).

Proton+ compares results originating from P&R of three logical topologies (8x8 λ-Router, 8x8 GWOR and 8x8 Standard-Crossbar), five different sets of node positions and various permutations of solver parameters. We used the node positions that produced the best result over all presented, shown in Figure 5(a). We manually designed three simple layout templates, presented in Figure 5(b-d), that connect to these node positions. On the last step of the optimization we optimized for the maximum insertion loss ($maxil$), just like Proton+.

7.1 Physical templates

All templates share some common features:

1) Each node has two endpoints: a modulator and a demodulator.

2) The power distribution network, not represented in these templates, can always be routed from the outside. Hence no other crossings in the router exist besides those considered by the template.

The **centralized grid** template is a $w \times h$ grid of GRUs where $w + h$ equals the number of nodes. Each node is connected with waveguides to two ports on the grid (one for sending, other for receiving), which are next to each other.

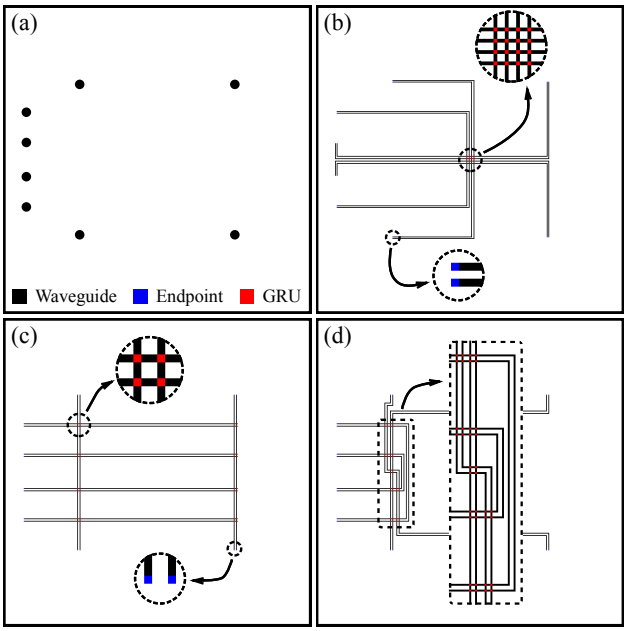


Figure 5: (a) Location of the eight nodes that produces the best result in Proton+. (b) A centralized grid template connecting those nodes. (c) A distributed grid template. (d) A custom template.

Table 3: Results for 8 nodes, 44 messages

	#WLs	Max IL	#MRRs	Time	
Proton+					T_{total}
λ-Router	**8**	**6.6 - 9.0**	56		134
GWOR	**7**	**8.1 - 11.3**	48		79
Std. crossbar	**8**	**10.5 - 13.0**	64		601.6
Ours				T_{opt}	T_{total}
Centralized	**8**	**3.126**	52	178	271
Distributed	**8**	**3.565**	48	37	376
Custom	**7**	**4.076**	40	-	6

T_{opt} is time to find the optimal solution, T_{total} is total execution time (for our method: $T_{total} = T_{opt}$ + time to prove optimality; for Proton+: the time that produces the best result).
Time in seconds, insertion loss in dB.

This router can be thought of as a different generalization of the 4x4 GWOR router, in Figure 2.

The grid itself was placed on the center of the die, the ports used by each node were chosen as to remove any crossings external to the grid and the waveguides connecting the nodes to the grid were manually routed to minimize bends.

The **distributed grid** template was built by placing horizontal or vertical pairs of waveguides starting at each node, with a GRU on each crossing.

The **custom** template was built specifically for this test case (i.e., these node positions and communication matrix). In particular, no message needs to use more than one MRR.

For the first two templates, the maximum number of MRRs per message in our tests was set to 2, but for the third it was set to 1, since it was designed with this in mind.

7.2 Comparison to the state of the art

Table 3 presents the various comparisons. Most important are the number of wavelengths and maximum insertion loss, but #MRRs and execution time are also given. Results from our method are the optimal solutions.

Number of wavelengths. Each node has only one modulator and some send 7 messages. Thus, 7 wavelengths is the minimum. The custom template achieves this value, but the grid templates require 8. Our method can reduce this number if given a smaller communication matrix, in contrast to the presented logical topologies.

Max. insertion loss. Our method produces results that are twice to three times better. This shows the substantial benefits of developing a combined logical topology and physical layout optimization algorithm.

MRR usage was not an optimization objective in these tests. Nevertheless, the comparison to Proton+ remains favourable.

Time. Grid templates have a total execution time comparable with Proton+. The custom template is much faster, mostly because of the model reduction technique from Section 4.3.2. Furthermore, the optimal solution is consistently reached in half or less than the total execution time. Thus, a designer that does not require proof of optimality can end the optimization once a satisfactory solution is found which, based on these results, is likely to appear quickly and be close to optimal.

8. CONCLUSION

In this work we defined the WRONoC design problem and presented a novel method for solving it. This method uses a physical layout template to combine logical topology and physical layout optimization. We also presented a new, flexible, routing element, the GRU. We used a MIP model and a 3-step optimization procedure to solve for the optimal solution. These combined efforts produce results vastly superior to the state of the art. In future work the proposed method can be extended to include optimization of the power distribution network and other GRU designs.

9. REFERENCES

[1] M. Ortín-Obón, L. Ramini, V. Viñals Yúfera, and D. Bertozzi, "A tool for synthesizing power-efficient and custom-tailored wavelength-routed optical rings," in *2017 22nd Asia and South Pacific Design Automation Conference (ASP-DAC)*, Jan 2017, pp. 300–305.

[2] I. O'Connor, M. Brière, E. Drouard, A. Kazmierczak, F. Tissafi-Drissi, D. Navarro, F. Mieyeville, J. Dambre, D. Stroobandt, J.-M. Fedeli, Z. Lisik, and F. Gaffiot, "Towards reconfigurable optical networks on chip," *ReCoSoC'05*, pp. 121–128, 2005.

[3] H. Gu, K. H. Mo, J. Xu, and W. Zhang, "A low-power low-cost optical router for optical networks-on-chip in multiprocessor systems-on-chip," in *2009 IEEE Computer Society Annual Symposium on VLSI*, May 2009, pp. 19–24.

[4] Y. Xie, M. Nikdast, J. Xu, W. Zhang, Q. Li, X. Wu, Y. Ye, X. Wang, and W. Liu, "Crosstalk noise and bit error rate analysis for optical network-on-chip," in *Proceedings of the 47th Design Automation Conference*, ser. DAC '10. New York, NY, USA: ACM, 2010, pp. 657–660.

[5] M. A. Seyedi, A. Descos, C.-H. Chen, M. Fiorentino, D. Penkler, F. Vincent, B. Szelag, and R. G. Beausoleil, "Crosstalk analysis of ring resonator switches for all-optical routing," *Opt. Express*, vol. 24, no. 11, pp. 11 668–11 676, May 2016.

[6] M. Ortín-Obón, L. Ramini, H. Tatenguem Fankem, V. Viñals, and D. Bertozzi, "A complete electronic network interface architecture for global contention-free communication over emerging optical networks-on-chip," in *Proceedings of the 24th Edition of the Great Lakes Symposium on VLSI*, ser. GLSVLSI '14. New York, NY, USA: ACM, 2014, pp. 267–272.

[7] A. Peano, L. Ramini, M. Gavanelli, M. Nonato, and D. Bertozzi, "Design technology for fault-free and maximally-parallel wavelength-routed optical networks-on-chip," in *2016 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, Nov 2016, pp. 1–8.

[8] X. Tan, M. Yang, L. Zhang, Y. Jiang, and J. Yang, "On a scalable, non-blocking optical router for photonic networks-on-chip designs," in *2011 Symposium on Photonics and Optoelectronics (SOPO)*, May 2011, pp. 1–4.

[9] M. Tala, M. Castellari, M. Balboni, and D. Bertozzi, "Populating and exploring the design space of wavelength-routed optical network-on-chip topologies by leveraging the add-drop filtering primitive," in *2016 Tenth IEEE/ACM International Symposium on Networks-on-Chip (NOCS)*, Aug 2016, pp. 1–8.

[10] L. Ramini, P. Grani, S. Bartolini, and D. Bertozzi, "Contrasting wavelength-routed optical noc topologies for power-efficient 3d-stacked multicore processors using physical-layer analysis," in *2013 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, March 2013, pp. 1589–1594.

[11] S. L. Beux, I. O'Connor, G. Nicolescu, G. Bois, and P. Paulin, "Reduction methods for adapting optical network on chip topologies to 3d architectures," *Microprocessors and Microsystems*, vol. 37, no. 1, pp. 87 – 98, 2013.

[12] A. von Beuningen and U. Schlichtmann, "Platon: A force-directed placement algorithm for 3d optical networks-on-chip," in *Proceedings of the 2016 on International Symposium on Physical Design*, ser. ISPD '16. New York, NY, USA: ACM, 2016, pp. 27–34.

[13] A. von Beuningen, L. Ramini, D. Bertozzi, and U. Schlichtmann, "Proton+: A placement and routing tool for 3d optical networks-on-chip with a single optical layer," *J. Emerg. Technol. Comput. Syst.*, vol. 12, no. 4, pp. 44:1–44:28, Dec. 2015.

[14] M. Nikdast, J. Xu, L. H. K. Duong, X. Wu, X. Wang, Z. Wang, Z. Wang, P. Yang, Y. Ye, and Q. Hao, "Crosstalk noise in wdm-based optical networks-on-chip: A formal study and comparison," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 23, no. 11, pp. 2552–2565, Nov 2015.

[15] Gurobi Optimization, Inc., *Gurobi Optimizer Reference Manual*. http://www.gurobi.com.

References

[1] M. Nikdast, J. Xu, L. H. K. Duong, X. Wu, X. Wang, Z. Wang, Z. Wang, P. Yang, Y. Ye, and Q. Hao. Crosstalk noise in wdm-based optical networks-on-chip: A formal study and comparison. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 23(11):2552–2565, Nov 2015.

[2] I. O'Connor, M. Brière, E. Drouard, A. Kazmierczak, F. Tissafi-Drissi, D. Navarro, F. Mieyeville, J. Dambre, D. Stroobandt, J.-M. Fedeli, Z. Lisik, and F. Gaffiot. Towards reconfigurable optical networks on chip. *ReCoSoC'05*, pages 121–128, 2005.

[3] Anja Von Beuningen, Luca Ramini, Davide Bertozzi, and Ulf Schlichtmann. Proton+: A placement and routing tool for 3d optical networks-on-chip with a single optical layer. *J. Emerg. Technol. Comput. Syst.*, 12(4):44:1–44:28, December 2015.

[4] Assaf Shacham, Keren Bergman, and Luca P. Carloni. Photonic networks-on-chip for future generations of chip multiprocessors. *IEEE Transactions on Computers*, 57(9):1246–1260, 2008.

[5] M. Tala, M. Castellari, M. Balboni, and D. Bertozzi. Populating and exploring the design space of wavelength-routed optical network-on-chip topologies by leveraging the add-drop filtering primitive. In *2016 Tenth IEEE/ACM International Symposium on Networks-on-Chip (NOCS)*, pages 1–8, Aug 2016.

[6] C. Chen, T. Zhang, P. Contu, J. Klamkin, A. K. Coskun, and A. Joshi. Sharing and placement of on-chip laser sources in silicon-photonic nocs. In *2014 Eighth IEEE/ACM International Symposium on Networks-on-Chip (NoCS)*, pages 88–95, Sept 2014.

[7] S. Koohi and S. Hessabi. All-optical wavelength-routed architecture for a power-efficient network on chip. *IEEE Transactions on Computers*, 63(3):777–792, 2014.

[8] H. Gu, K. H. Mo, J. Xu, and W. Zhang. A low-power low-cost optical router for optical networks-on-chip in multiprocessor systems-on-chip. In *2009 IEEE Computer Society Annual Symposium on VLSI*, pages 19–24, May 2009.

[9] Yiyuan Xie, Mahdi Nikdast, Jiang Xu, Wei Zhang, Qi Li, Xiaowen Wu, Yaoyao Ye, Xuan Wang, and Weichen Liu. Crosstalk noise and bit error rate analysis for optical network-on-chip. In *Proceedings of the 47th Design Automation Conference*, DAC '10, pages 657–660, New York, NY, USA, 2010. ACM.

[10] M. J. R. Heck and J. E. Bowers. Energy efficient and energy proportional optical interconnects for multi-core processors: Driving the need for on-chip sources. *IEEE Journal of Selected Topics in Quantum Electronics*, 20(4):332–343, July 2014.

[11] C. Batten, A. Joshi, J. Orcutt, C. Holzwarth, M. Popovic, J. Hoyt, F. Kartner, R. Ram, V. Stojanovic, and K. Asanovic. Building manycore processor-to-dram networks with monolithic cmos silicon photonics. *IEEE Micro*, pages 1–1, 2016.

[12] C. Gunn. Cmos photonics for high-speed interconnects. *IEEE Micro*, 26(2):58–66, March 2006.

[13] M. Georgas, J. Orcutt, R. J. Ram, and V. Stojanović. A monolithically-integrated optical receiver in standard 45-nm soi. In *2011 Proceedings of the ESSCIRC (ESSCIRC)*, pages 407–410, Sept 2011.

[14] A. Peano, L. Ramini, M. Gavanelli, M. Nonato, and D. Bertozzi. Design technology for fault-free and maximally-parallel wavelength-routed optical networks-on-chip. In *2016 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pages 1–8, Nov 2016.

[15] J. F. Buckwalter, X. Zheng, G. Li, K. Raj, and A. V. Krishnamoorthy. A monolithic 25-gb/s transceiver with photonic ring modulators and ge detectors in a 130-nm cmos soi process. *IEEE Journal of Solid-State Circuits*, 47(6):1309–1322, June 2012.

[16] Marta Ortín-Obón, Luca Ramini, Herve Tatenguem Fankem, Víctor Viñals, and Davide Bertozzi. A complete electronic network interface architecture for global contention-free communication over emerging optical networks-on-chip. In *Proceedings of the 24th Edition of the Great Lakes Symposium on VLSI*, GLSVLSI '14, pages 267–272, New York, NY, USA, 2014. ACM.

[17] M. Ortín-Obón, L. Ramini, V. Viñals Yúfera, and D. Bertozzi. A tool for synthesizing power-efficient and custom-tailored wavelength-routed optical rings. In *2017 22nd Asia and South Pacific Design Automation Conference (ASP-DAC)*, pages 300–305, Jan 2017.

[18] Kai Feng, Yaoyao Ye, and Jiang Xu. A formal study on topology and floorplan characteristics of mesh and torus-based optical networks-on-chip. *Microprocessors and Microsystems*, 37(8, Part B):941 – 952, 2013. Embedded Multicore Systems: Architecture, Performance and Application.

[19] Mahdi Nikdast, Xu Jiang, Luan H. K. Duong, Wu Xiaowen, Wang Zhehui, Wang Xuan, and Wang Zhe. Fat-tree-based optical interconnection networks under crosstalk noise constraint. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 23(1):156–169, 2015.

[20] X. Tan, M. Yang, L. Zhang, Y. Jiang, and J. Yang. On a scalable, non-blocking optical router for photonic networks-on-chip designs. In *2011 Symposium on Photonics and Optoelectronics (SOPO)*, pages 1–4, May 2011.

[21] L. Ramini, P. Grani, S. Bartolini, and D. Bertozzi. Contrasting wavelength-routed optical noc topologies for power-efficient 3d-stacked multicore processors using physical-layer analysis. In *2013 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pages 1589–1594, March 2013.

[22] M. Ortín-Obón, M. Tala, L. Ramini, V. Viñals-Yufera, and D. Bertozzi. Contrasting laser power requirements of wavelength-routed optical noc topologies subject to the floorplanning, placement, and routing constraints of a 3-d-stacked system. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 25(7):2081–2094, July 2017.

[23] S. V. R. Chittamuru and S. Pasricha. Spectra: A framework for thermal reliability management in silicon-photonic networks-on-chip. In *2016 29th International Conference on VLSI Design and 2016 15th International Conference on Embedded Systems (VLSID)*, pages 86–91, Jan 2016.

[24] F. Jiao, S. Dong, B. Yu, B. Li, and U. Schlichtmann. Thermal-aware placement and routing for 3d optical networks-on-chips. In *2018 IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 1–4, May 2018.

[25] M. Meyer, Y. Okuyama, and A. B. Abdallah. A power estimation method for mesh-based photonic noc routing algorithms. In *2016 Fourth International Symposium on Computing and Networking (CANDAR)*, pages 451–453, 2016.

[26] C. J. Nitta, M. K. Farrens, and V. Akella. Resilient microring resonator based photonic networks. In *2011 44th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*, pages 95–104, Dec 2011.

[27] Po Dong, Wei Qian, Hong Liang, Roshanak Shafiiha, Ning-Ning Feng, Dazeng Feng, Xuezhe Zheng, Ashok V. Krishnamoorthy, and Mehdi Asghari. Low power and compact reconfigurable multiplexing devices based on silicon microring resonators. *Opt. Express*, 18(10):9852–9858, May 2010.

[28] Rong Cao, Kun Wang, Huaxi Gu, Bowen Zhang, and Xiaoshan Yu. A crosstalk-aware wavelength assignment method for optical network-on-chip. *IEICE Electronics Express*, 13(18):20160821–20160821, 2016.

[29] M. Ashkan Seyedi, Antoine Descos, Chin-Hui Chen, Marco Fiorentino, David Penkler, François Vincent, Bertrand Szelag, and Raymond G. Beausoleil. Crosstalk analysis of ring resonator switches for all-optical routing. *Opt. Express*, 24(11):11668–11676, May 2016.

[30] Le Beux Sébastien, Li Hui, Nicolescu Gabriela, Trajkovic Jelena, and O'Connor Ian. Optical crossbars on chip, a comparative study based on worst-case losses. *Concurrency and Computation: Practice and Experience*, 26(15):2492–2503, 2014.

[31] K. Preston, N. Sherwood-Droz, J. S. Levy, and M. Lipson. Performance guidelines for wdm interconnects based on silicon microring resonators. In *CLEO: 2011 - Laser Science to Photonic Applications*, pages 1–2, May 2011.

[32] Sébastien Le Beux, Ian O'Connor, Gabriela Nicolescu, Guy Bois, and Pierre Paulin. Reduction methods for adapting optical network on chip topologies to 3d architectures. *Microprocessors and Microsystems*, 37(1):87 – 98, 2013.

[33] A. Jalabert, S. Murali, L. Benini, and G. De Micheli. ×pipesCompiler: a tool for instantiating application specific networks on chip. In *Proceedings Design, Automation and Test in Europe Conference and Exhibition*, volume 2, pages 884–889 Vol.2, Feb 2004.

[34] A. Boos, L. Ramini, U. Schlichtmann, and D. Bertozzi. Proton: An automatic place-and-route tool for optical networks-on-chip. In *2013 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pages 138–145, Nov 2013.

[35] Anja von Beuningen and Ulf Schlichtmann. Platon: A force-directed placement algorithm for 3d optical networks-on-chip. In *Proceedings of the 2016 on International Symposium on Physical Design*, ISPD '16, pages 27–34, New York, NY, USA, 2016. ACM.

[36] S. Le Beux, J. Trajkovic, I. O' Connor, G. Nicolescu, G. Bois, and P. Paulin. Optical ring network-on-chip (ornoc): Architecture and design methodology. In *2011 Design, Automation & Test in Europe*, pages 1–6, 2011.

[37] Ali Naimi Sadigh, Marzieh Mozafari, and Ali Husseinzadeh Kashan. A mixed integer linear program and tabu search approach for the complementary edge covering problem. *Advances in Engineering Software*, 41(5):762 – 768, 2010.

[38] João Pedro Pedroso. *Tabu Search for Mixed Integer Programming*, pages 247–261. Springer US, Boston, MA, 2005.

[39] Gurobi Optimization, Inc. *Gurobi Optimizer Reference Manual*. http://www.gurobi.com, 2018.

[40] International Business Machines Corp. *IBM ILOG CPLEX Optimization Studio CPLEX User's Manual*. https://www.ibm.com/analytics/data-science/prescriptive-analytics/cplex-optimizer, 2016.

[41] Antti Alhonen, Erno Salminen, Lasse Lehtonen, and Timo D. Hämäläinen. A scalable, non-interfering, synthesizable network-on-chip monitor – extended version. *Microprocessors and Microsystems*, 37(4):446 – 459, 2013.