CustomTopo: A Topology Generation Method for Application-Specific Wavelength-Routed Optical NoCs

Mengchu Li[†▽], Tsun-Ming Tseng[†], Davide Bertozzi[⋆], Mahdi Tala[⋆], and Ulf Schlichtmann[†]
mengchu.li@campus.lmu.de, {tsun-ming.tseng, ulf.schlichtmann}@tum.de, {davide.bertozzi, mahdi.tala}@unife.it
[†]Chair of Electronic Design Automation, Technical University of Munich, Arcisstraße 21, 80333 München, Germany
[▽]Ludwig-Maximilians-Universität München, Geschwister-Scholl-Platz 1, 80539 München, Germany
[⋆]Dipartimento di Ingegneria, University of Ferrara, Via Saragat, 1, 44122 Ferrara, Italy

ABSTRACT

Optical network-on-chip (NoC) is a promising platform beyond electronic NoCs. In particular, wavelength-routed optical network-on-chip (WRONoC) is renowned for its high bandwidth and ultra-low signal delay. Current WRONoC topology generation approaches focus on full-connectivity, i.e. all masters are connected to all slaves. This assumption leads to wasted resources for application-specific designs. In this work, we propose CustomTopo: a general solution to the topology generation problem on WRONoCs that supports customized connectivity. CustomTopo models the topology structure and its communication behavior as an integer-linear-programming (ILP) problem, with an adjustable optimization target considering the number of add-drop filters (ADFs), the number of wavelengths, and insertion loss. The time for solving the ILP problem in general positively correlates with the network communication densities. Experimental results show that CustomTopo is applicable for various communication requirements, and the resulting customized topology enables a remarkable reduction in both resource usage and insertion loss.

1 Introduction

With the rapid growth of on-chip communication in multi-processor systems-on-chips (MPSoCs), optical networks-on-chips (ONoCs) have emerged as an appealing next-generation platform, thanks to their advantages in high bandwidth and ultra-low signal delay. The key components of ONoCs include silicon waveguides for signal transmission and optical switching blocks for signal routing. Based on the routing mechanisms, ONoCs can be classified into two categories: 1) *active networks* that apply a real-time switching mechanism to operate the routing process, and 2) *passive networks* that apply add-drop filters (ADFs) [1] tuned to fixed wavelengths. The latter are also known as wavelength-routed optical networks-on-chips (WRONoCs).

WRONoCs are renowned for supporting congestion- and reconfiguration-free communication [2] [3]. In contrast to active networks where path setup/reservation for each signal is performed during the communication process [4–6], WRONoCs statically reserve signal paths in the design phase,

ICCAD 2018, DOI: 10.1145/3240765.3240789
https://dl.acm.org/citation.cfm?id=3240789

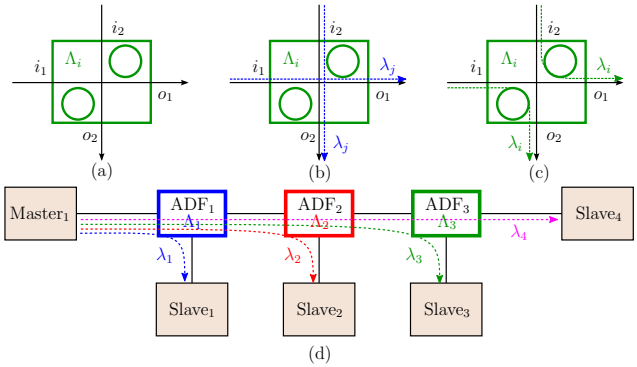

Figure 1: (a) A 2×2 ADF structure. (b) Non-resonant signals pass through. (c) On-resonant signals change their direction. (d) A simple WRONoC topology.

and thus do not suffer from signal delay caused by path setup and conflicts resolution. Signals modulated to different wavelengths can travel along the same waveguides (which is known as wavelength-division multiplexing (WDM)), until they are demultiplexed at different ADFs. Figure 1(a) illustrates a 2-input×2-output ADF structure, which includes a pair of crossing waveguides and two microring resonators (MRRs) configured to be on-resonant with a specific wavelength denoted as Λ_i. When signals of wavelength λ_j other than Λ_i enter the ADF, they will pass through the ADF keeping their original direction as shown in Figure 1(b); but when signals of wavelength λ_i equal to Λ_i enter the ADF, they will resonate with the MMRs and thus change their directions as shown in Figure 1(c). Figure 1(d) illustrates a simple WRONoC topology, where *master* and *slaves* can be regarded as the output and the input ports of communication nodes, respectively. With three ADFs configured to Λ_1, Λ_2, and Λ_3, the master communicates with its four slaves with signals modulated on four different wavelengths: signals modulated to λ_1, λ_2, and λ_3 get demultiplexed at ADF$_1$, ADF$_2$, and ADF$_3$ to the first three slaves; since the signal modulated to λ_4 does not resonate with any ADF, it travels straight through all ADFs and arrives at Slave$_4$.

Considering that WRONoCs support *full bandwidth*, i.e., all signals can travel along all waveguides simultaneously, most related work makes the natural assumption that a WRONoC topology should provide *full connectivity* (or the communication graph should be *complete*), i.e all masters require connection to all slaves [2, 3, 7, 8]. However, this assumption is often over-conservative. Representative examples are processor-memory networks. Though a processor may communicate with all other processors and memories, a memory typically never communicates with another memory [9, 10]. Besides, for a 3D setting where an optical NoC is

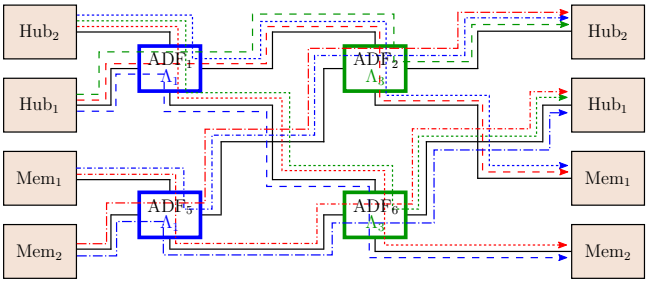

Figure 2: Logic topology for a 2-hub×2-memory design without redundancy (signals from different masters differ in form, and signals of different wavelengths differ in color).

vertically stacked on top of an electronic NoC. If the latency- and/or bandwidth-critical communication flows are known in advance, the ONoC can be used to "accelerate" the critical flows, while the ENoC delivers baseline global connectivity [11, 12].

For applications that do not require complete communication, providing full connectivity may lead to significant *waste of resources*. Two important metrics [8] for resource usage in a WRONoC topology are *the number of wavelengths* and *the number of ADFs*, which are limited by both manufacturing technologies and performance factors: constrained by available WDM bandwidth, many works point out that the maximal number of wavelengths cannot surpass 64 [13–15], and some others indicate that only 16 distinguishable and stable wavelengths can be achieved [11]. Moreover, wavelength usage correlates with ADF usage. As each ADF introduces two MRRs and a pair of (crossed) waveguides, more ADFs result in a higher power loss and crosstalk noise power, and thus also a lower signal-to-noise ratio (SNR) [16]. However, under the full-connectivity context, the required number of wavelengths is linearly proportional to the number of communication nodes, and the required number of ADFs increases even quadratically [17], which raises severe *scalability concerns*.

Take a 2-hub×2-memory design[1] proposed in [18] as an example: if we implement the design under the complete communication assumption, a reasonable choice is to apply the widely-acknowledged 4×4 λ-router [2], which consists of 6 ADFs configured to 4 different wavelengths. But if we target the actual network function: 1) there is no communication between the two memories, and 2) neither hub nor memory transmits a signal to itself. Therefore, the wavelengths and the ADFs reserved for memory-to-memory communication and for self-communication are indeed redundant. Figure 2 illustrates an exemplary topology that removes the redundancy, where colored dashed lines represent the routing paths of signals. As we can see, 4 ADFs configured to 2 different wavelengths are already enough for all necessary communications, i.e. a resource reduction of −33% ADFs and −50% wavelengths is achieved, which gives an indication of the potential benefits that we can expect from a *customized topology* fitting to the *required connectivity*.

Unlike full-connectivity designs, for which there are generally applicable routers such as λ-router and GWOR [3], customized designs require individual treatment based on a well-formulated topology generation model. Previous work has studied the WRONoC topology from some specific aspects. [19] lays the groundwork for a topology synthesis discipline, but it is restricted to symmetric $n\times n$ topologies. [11] proposes a reduction method on λ-router for application-specific designs. But since λ-router is not intended for incomplete communication, its performance is limited and heavily depends on the symmetry of the communication graph. As the scope of WRONoC applications continuously extends, the symmetric subclass becomes insufficient [12]. Thus, there is a pressing need for a general WRONoC topology generation method that is not limited to any initial topology, communication graph, or router-structure.

In this paper, we propose *CustomTopo*: a general solution to the topology generation problem on WRONoCs, without the assumption that the network communication must be complete or symmetric. Based on a communication matrix derived from an input communication graph, we model the topology generation problem as an integer linear programming problem, the optimization objective of which focuses on *reducing the ADF and the wavelength usage*, while *keeping the insertion loss small*.

2 General Model for WRONoCs

The general model for WRONoCs is formulated as follows:

Input: a *communication graph* specifying all master-slave pairs that require communication.

Output: a full-bandwidth *topology* specifying the logic connection between network components, the wavelength usage, and the signal path between each master-slave pair.

Minimization Objective [8, 19]:
1) the number of ADFs,
2) the number of wavelengths assigned to ADFs.
3) the worst-case insertion loss of all signal paths[2].

2.1 From Communication Graph to Communication Matrix

For a given communication graph, we build a set $\mathcal{M}$ for masters and a set $\mathcal{S}$ for slaves. In case the communication graph is not complete, we introduce a function $\varphi: \mathcal{M}\times\mathcal{S}\to\{1,0\}$ to represent whether a master $m\in\mathcal{M}$ communicates with a slave $s\in\mathcal{S}$:

$$\varphi(m,s)=\begin{cases}1 & \text{if } m \text{ communicates with } s,\\ 0 & \text{otherwise.}\end{cases}$$

If we denote the number of masters as $n_m := |\mathcal{M}|$ and the number of slaves as $n_s := |\mathcal{S}|$, the entire communication behavior can be modeled as an $n_s\times n_m$ *communication matrix*, in which each entry represents a wavelength:

$$\begin{array}{c} \\ s_1\\ s_2\\ \vdots\\ s_{n_s}\end{array}\begin{array}{c}\begin{array}{cccc} m_1 & m_2 & \cdots & m_{n_m}\end{array}\\ \begin{pmatrix} \lambda_{m_1,s_1} & \lambda_{m_2,s_1} & \cdots & \lambda_{m_{n_m},s_1}\\ \lambda_{m_1,s_2} & \lambda_{m_2,s_2} & \cdots & \lambda_{m_{n_m},s_2}\\ \vdots & \vdots & \ddots & \vdots\\ \lambda_{m_1,s_{n_s}} & \lambda_{m_2,s_{n_s}} & \cdots & \lambda_{m_{n_m},s_{n_s}}\end{pmatrix}\end{array}.$$

For all $1\le i\le n_m$ and $1\le j\le n_s$, if $\varphi(m_i,s_j)=0$, i.e. there is no communication between m_i and s_j, the corresponding entry λ_{m_i,s_j} will be set as NA; otherwise if $\varphi(m_i,s_j)=1$, the entry λ_{m_i,s_j} will be set as the wavelength that m_i uses for communication with s_j.

On WRONoCs, a wavelength can be shared among different master-slave pairs, but the corresponding signal paths must not overlap [19]. This puts two constraints on wavelength usage at end nodes: 1) *wavelengths for communication between the same master and different slaves must be different*; and 2) *wavelengths for communication between different masters and the same slave must be different*. With

[1] *Hubs* indicate optical interfaces connected to processor clusters and *memories* indicate memory controllers.

[2] We assume that optical power is provided by an array of continuous-wave off-chip laser sources, which are multiplexed onto a single input power waveguide. On the chip, the optical power is then distributed to the transmitters for modulation through a power distribution network. In this context, controlling the worst-case insertion loss turns out to be a simple yet effective way to limit the output power requirements for the laser sources [10, 20, 21].

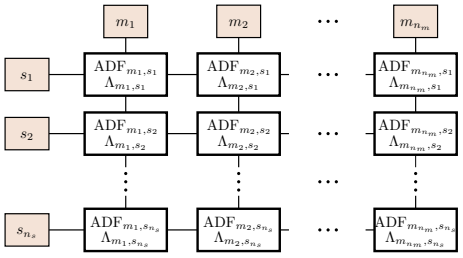

Figure 3: An initial topology with n_m masters and n_s slaves.

our communication matrix, these constraints can be formulated as follows:

$$\forall 1\le i\le n_m\ \forall 1\le j\le n_s\ \forall 1\le k_1,k_2\le n_s\ \forall 1\le k_1',k_2'\le n_m:$$
$$(\varphi(m_i,s_{k_1})=\varphi(m_i,s_{k_2})=1)\ \Rightarrow\ \lambda_{m_i,s_{k_1}}\neq\lambda_{m_i,s_{k_2}},\quad (1)$$
$$(\varphi(m_{k_1'},s_j)=\varphi(m_{k_2'},s_j)=1)\ \Rightarrow\ \lambda_{m_{k_1'},s_j}\neq\lambda_{m_{k_2'},s_j},\quad (2)$$

which means that except for the entries set as NA, all entries in the same column and in the same row of the matrix must be pairwise distinct. We call the communication matrices that satisfy these constraints *valid communication matrices*.

2.2 From Communication Matrix to Topology

2.2.1 Initial Topology

A valid communication matrix can be transformed into a full-bandwidth topology straightforwardly by implementing an ADF for each non-NA entry, as shown in Figure 3. An ADF at the i-th column and the j-th row of the topology is indexed as ADF_{m_i,s_j} with a label Λ_{m_i,s_j}, indicating that it only resonates with wavelength λ_{m_i,s_j}. By sequentially connecting the ADFs in the same columns and in the same rows, signals can be delivered to their destinations simultaneously without conflict. We refer to this topology as *initial topology* in the rest of this section.

2.2.2 Topology Optimization: Three Targets

As proposed in [8], a WRONoC topology is specified by two design parameters: the number of ADFs and the number of different wavelengths assigned to the ADFs. However, when considering the physical layout, insertion loss becomes another key metric. Though the explicit insertion loss is dependent on the physical features and thus cannot be derived from a logic topology [20], we still try to approximate the insertion loss for comprehensive optimization.

In general, we optimize the initial topology based on *three criteria*: wavelength usage, ADF usage, and insertion loss[3].

2.2.3 Expectation: Wavelength Usage

The wavelength usage for full-bandwidth communication can be read from the communication matrix, where the number of wavelengths equals the number of the distinct entries.

For a complete communication graph, the theoretical minimum wavelength usage can be derived by transforming the matrix into a colored *bipartite graph*: $\mathcal{M}$ and $\mathcal{S}$ can be regarded as two sets of vertices, and an entry $\lambda_{m_i,s_j}\neq$NA can be regarded as the color denotation of an edge from m_i to s_j. Thus, the entries in the same column/row of the communication matrix can be regarded as the colors of adjacent edges sharing the same master/slave vertex. In this manner, the design constraints (1)(2) can be transformed into an *edge coloring problem*, and the minimum number of distinct entries is thus equal to $\omega:=\max\{n_m,n_s\}$, according to Vizing's theorem [22]. Since complete communication is not a necessity in our topology generation model, we can expect the minimum wavelength usage in a customized topology to be smaller than or equal to ω.

[3] Another layout-dependent performance factor of ONoCs is signal-to-noise ratio (SNR). As proposed in [16], SNR usually negatively correlates with the number of ADFs and wavelengths. In this work, we approximate the maximization of SNR by existing optimization targets.

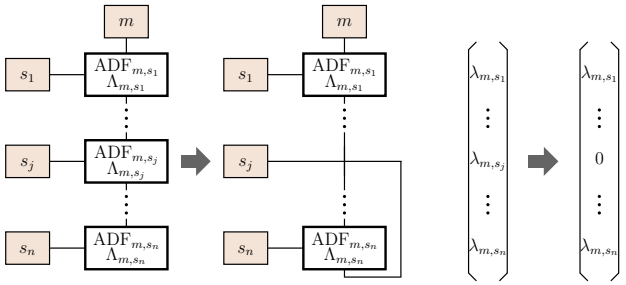

Figure 4: ADF reduction for a one-column matrix.

2.2.4 Minimization: ADF Usage

The ADF usage in an initial topology can be directly read from the communication matrix, where the number of ADFs is equal to the number of the non-NA entries. However, considering a 2×2 ADF structure as shown in Figure 1, some of the ADFs can be removed by utilizing wavelengths shared among different master-slave pairs.

We first consider a simple one-column communication matrix for one master and its n slaves. The initial topology derived from this matrix consists of n ADFs. However, $n-1$ ADFs are already sufficient to carry out the communication, as shown in Figure 4: if we arbitrarily choose an ADF_{m,s_j} to be replaced by a normal waveguide, and connect s_j to the vacant output port of the bottom ADF, signals modulated to λ_{m,s_j} will travel across all the sequentially connected ADFs and finally arrive at s_j without being affected, since $\lambda_{m,s_j}\neq\Lambda_{m,s_i}$ for all $1\le i\le n, i\neq j$. In this case, we define s_j as the *default slave* of m and we define the signal path from m to s_j as the *default path* of m. Besides, we denote λ_{m,s_j} as 0, indicating that the communication between m and s_j does not rely on any ADF for demultiplexing.

We now consider an $n_s\times n_m$ communication matrix with a 0 in each column. An example is shown in Figure 5(a), where each master has its own default slave. Specifically, the default slave of m_1 is s_1, and the default slave of m_2 is s_2. In this case, if both (m_1,s_2) and (m_2,s_1) apply λ_k for communication, the resulting topology shown in Figure 5(b) can be safely reduced by one ADF, either as shown in Figure 5(c) or as shown in Figure 5(d), with corresponding changes to the signal communication paths. In particular, the removal of ADF_{m_2,s_1} results in a detour of the signal path from m_2 to s_1 through the default path of m_2, to share the same ADF with m_1 for demultiplexing.

In general, suppose that m_{i_1}, m_{i_2}, s_{j_1}, and s_{j_2} are masters and slaves that fulfill the following constraint:

$$\lambda_{m_{i_1},s_{j_2}}=\lambda_{m_{i_2},s_{j_1}}=0\ \wedge\ \lambda_{m_{i_1},s_{j_1}}=\lambda_{m_{i_2},s_{j_2}}:=\lambda_k.\quad (3)$$

Then m_{i_1} and m_{i_2} can share the same ADF, and the corresponding communication matrix will be in the form:

$$\begin{array}{c} \\ \vdots \\ s_{j_1} \\ \vdots \\ s_{j_2} \\ \vdots \end{array}\!\!\begin{array}{c} \begin{array}{ccccc} \cdots & m_{i_1} & \cdots & m_{i_2} & \cdots \end{array} \\ \begin{pmatrix} \vdots & \vdots & \vdots & \vdots & \vdots \\ \cdots & \lambda_k & \cdots & 0 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \cdots & 0 & \cdots & \lambda_k & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \end{pmatrix} \end{array} \text{ or } \begin{array}{c} \\ \vdots \\ s_{j_2} \\ \vdots \\ s_{j_1} \\ \vdots \end{array}\!\!\begin{array}{c} \begin{array}{ccccc} \cdots & m_{i_1} & \cdots & m_{i_2} & \cdots \end{array} \\ \begin{pmatrix} \vdots & \vdots & \vdots & \vdots & \vdots \\ \cdots & 0 & \cdots & \lambda_k & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \cdots & \lambda_k & \cdots & 0 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \end{pmatrix} \end{array}.$$

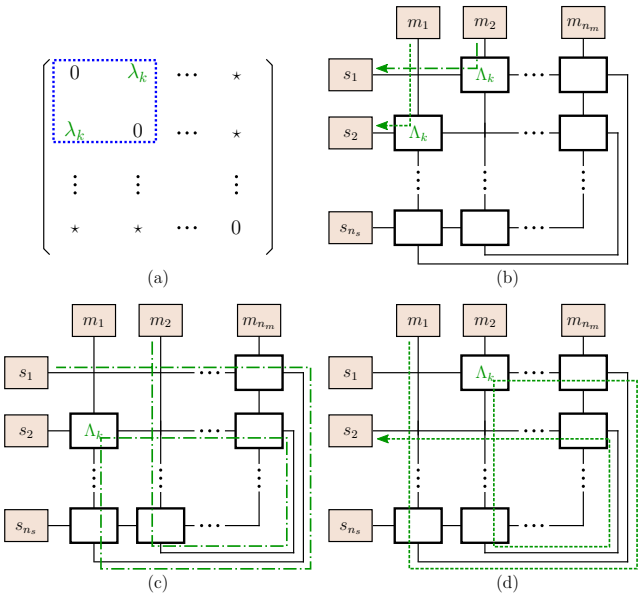

Figure 5: ADF reduction with ADF-sharing structure. (a) A communication matrix containing an ADF-sharing structure. (b) Reduced topology with a default path for each master. (c)(d) Further ADF reduction options. The resulting changed signal paths are indicated by green dash lines.

We refer to this situation as $\{\lambda_{m_{i_1},s_{j_1}},\lambda_{m_{i_1},s_{j_2}},\lambda_{m_{i_2},s_{j_1}},\lambda_{m_{i_2},s_{j_2}}\}$ *form an ADF-sharing structure*. To minimize the ADF usage, we aim to maximize the number of ADF-sharing structures in the communication matrix.

2.2.5 *Minimization: Insertion Loss in Signal Paths*

On WRONoCs, insertion loss results from waveguide length (propagation loss), waveguide bending (bending loss), waveguide crossings (crossing loss), and ADFs (through loss and drop loss) [23]. Among them, propagation loss, bending loss, and crossing loss are dependent on the physical layout of the chip. In this work, we do not want to make restrictive assumptions on the physical location of the masters and the slaves, which will be determined by placement and routing tools at a later design stage. Therefore, we do not directly target the optimization of the layout-aware insertion loss, but rather a more abstract yet indirect metric with tight correlation with the final insertion loss figures. We identify this metric with the insertion loss contributions from ADFs in the logic topology, namely the through loss and the drop loss of the constituting MRRs, and the internal crossing loss in the ADF. For all master-slave pairs that communicate, we derive their corresponding signal paths, and minimize the worst-case insertion loss.

In order to reduce the accuracy gap in insertion loss analysis between logic topology and physical one, we try to reduce inevitable waveguide crossings implied by the logic topology. By taking advantage of the reduction approach introduced in Section 2.2.4, some ADFs can be removed from the initial topology, but the waveguide crossings contained in these ADFs will be kept by default to retain the signal paths. Among these crossings, some are removable by rearranging the location of network components, but some are inevitable for the given logic connection, as shown in Figure 6(a). However, by optimizing the communication matrix, we can reduce the formation of inevitable crossings. For example, Figure 6(b) shows a communication matrix and its resulting topology that supports the same connectivity as Figure 6(a).

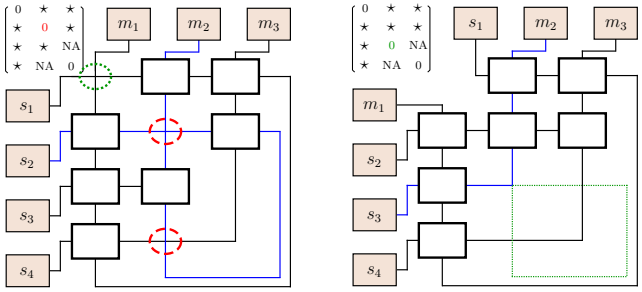

Figure 6: Two topologies supporting the same connectivity with different default paths (blue line). (a) contains a removable crossing (green circle), and two inevitable crossings (red circles); (b) removes the removable crossing and transforms the inevitable crossings into a removable waveguide loop (green dash line).

The only difference is that the default slave of m_2 is changed from s_2 to s_3, which results in a new default path containing a removable waveguide loop without ADF, and thus all waveguide crossings in this loop become removable as well. To reduce external crossings in the final layout, we maximize the number of removable crossings implied by the topology.

3 Mathematical Model of CustomTopo

We implement the topology optimization approach introduced in Section 2 as an integer-linear-programming model.

3.1 Matrix Initialization

The wavelength assignment in a topology is specified by entries in the communication matrix. For an $n_s \times n_m$ communication matrix, we denote the maximum number of the non-NA entries in each column and in each row as n'_s and n'_m, respectively. As introduced in Section 2.2.3, we can expect the upper bound of wavelength usage to be $\omega := \max\{n'_s, n'_m\}$. Thus, the range of the entries in the communication matrix can be denoted as a set of integers: $\{0,1,\cdots,\omega\}$. In particular, we add 0 to this set to indicate the default slaves, as introduced in Section 2.2.4.

For each individual entry λ_{m_i,s_j}, we introduce a sequence of binary variables $(b_k^{\lambda_{i,j}})_{0\le k\le\omega}$, where $b_k^{\lambda_{i,j}} = 1$ implies that λ_{m_i,s_j} is assigned with value k, i.e. (m_i,s_j) uses wavelength k for communication. We introduce the following constraint to ensure that each entry is assigned exactly once:

$$\sum_{0\le k\le\omega} b_k^{\lambda_{i,j}} = 1. \tag{4}$$

3.2 Conflict Resolution

To avoid signal conflicts, all entries in the same column and in the same row of the communication matrix must be pairwise distinct, as described in constraints (1)(2) in Section 2.1. We linearize these constraints as follows:

$$\forall 1\le i\le n_m\ \forall 0\le k\le\omega\colon \sum_{0\le j\le n_s} b_k^{\lambda_{i,j}} \le 1, \tag{5}$$

$$\forall 1\le j\le n_s\ \forall 0\le k\le\omega\colon \sum_{0\le i\le n_m} b_k^{\lambda_{i,j}} \le 1, \tag{6}$$

which ensure that a value appears at most once in each column and in each row of the matrix.

3.3 ADF-Sharing Structure

To optimize the ADF usage, for every two masters m_{i_1}, m_{i_2} and every two slaves s_{j_1}, s_{j_2}, we introduce a binary variable

$b^{\text{share}}_{i_1,j_1,i_2,j_2}$ to indicate whether they form an ADF-sharing structure, as introduced in Section 2.2.4. We then introduce the following constraints:

$$b_0^{\lambda_{i_1,j_2}} \geq b^{\text{share}}_{i_1,j_1,i_2,j_2}, \tag{7}$$

$$b_0^{\lambda_{i_2,j_1}} \geq b^{\text{share}}_{i_1,j_1,i_2,j_2}, \tag{8}$$

$$\sum_{1\leq k\leq\omega} b_k^{\lambda_{i_1,j_1}}\cdot k - \sum_{1\leq k\leq\omega} b_k^{\lambda_{i_2,j_2}}\cdot k \leq M\cdot(1-b^{\text{share}}_{i_1,j_1,i_2,j_2}), \tag{9}$$

$$\sum_{1\leq k\leq\omega} b_k^{\lambda_{i_2,j_2}}\cdot k - \sum_{1\leq k\leq\omega} b_k^{\lambda_{i_1,j_1}}\cdot k \leq M\cdot(1-b^{\text{share}}_{i_1,j_1,i_2,j_2}), \tag{10}$$

where M is an extremely large auxiliary constant. If $b^{\text{share}}_{i_1,j_1,i_2,j_2}=1$, the above constraints can be considered as the linear form of constraint (3) in Section 2.2.4. Specifically, constraints (7)(8) imply that $\lambda_{m_{i_1},s_{j_2}}$ and $\lambda_{m_{i_2},s_{j_1}}$ will be assigned with 0, and constraints (9)(10) imply that $\lambda_{m_{i_1},s_{j_1}}$ and $\lambda_{m_{i_2},s_{j_2}}$ will be assigned with the same positive value k. Otherwise if $b^{\text{share}}_{i_1,j_1,i_2,j_2}=0$, the above constraints trivially hold and thus do not influence the wavelength assignment.

3.4 Topology Construction

We initialize the topology by introducing a binary variable $b^{\Lambda_{i,j}}$ for each entry in the communication matrix. $b^{\Lambda_{i,j}}=1$ indicates that there is an ADF at the i-th column and the j-th row of the topology, as introduced in Section 2.2.1. By default, there is an ADF for each positive entry, except for the entries that form an ADF-sharing structure, where the two positive entries can share one single ADF. This can be formulated as the following constraint:

$$b^{\Lambda_{i,j}} \geq \sum_{1\leq k\leq\omega} b_k^{\lambda_{i,j}} - \sum_{m_a\in\mathcal{M},s_b\in\mathcal{S}} b^{\text{share}}_{i,j,a,b}, \tag{11}$$

which means that for a positive entry λ_{m_i,s_j}, if it is not part of an ADF-sharing structure, $b^{\Lambda_{i,j}}$ will be set to 1; otherwise $b^{\Lambda_{i,j}}$ can be either 1 or 0.

We then introduce the following constraints to model the ADF usage for every two masters m_{i_1}, m_{i_2}, and every two slaves s_{j_1}, s_{j_2}:

$$b^{\Lambda_{i_1,j_1}}+b^{\Lambda_{i_2,j_2}} \leq 2-b^{\text{share}}_{i_1,j_1,i_2,j_2}, \tag{12}$$

$$b^{\Lambda_{i_1,j_1}}+b^{\Lambda_{i_2,j_2}} \geq b^{\text{share}}_{i_1,j_1,i_2,j_2}. \tag{13}$$

Thus, if $b^{\text{share}}_{i_1,j_1,i_2,j_2}=1$, either $b^{\Lambda_{i_1,j_1}}$ or $b^{\Lambda_{i_2,j_2}}$ will be set to 1 while the other will be set to 0. Otherwise if $b^{\text{share}}_{i_1,j_1,i_2,j_2}=0$, the above constraints become trivial and thus do not influence the ADF usage.

3.5 Signal Paths and Insertion Loss

For each master-slave pair (m_i,s_j), we approximate the insertion loss in its signal path as the summation of through loss, drop loss, and crossing loss caused by ADFs, as introduced in Section 2.2.5.

Specifically, if a signal passes through an ADF without direction change, we denote the insertion loss at the ADF as $\mathfrak{I}_{\text{ADF}}$, and calculate it as:

$$\mathfrak{I}_{\text{ADF}}=2\cdot\mathfrak{I}_{\text{through}}+\mathfrak{I}_{\text{cross}},$$

where $\mathfrak{I}_{\text{through}}$ indicates the insertion loss resulting from a non-resonant MRR, and $\mathfrak{I}_{\text{cross}}$ indicates the insertion loss resulted from a waveguide crossing. $\mathfrak{I}_{\text{through}}$ is multiplied with 2 since an ADF consists of two MRRs. On the other hand, if a signal is *dropped* by an ADF, i.e. the signal resonates with the ADF, we denote the drop loss at the ADF as $\mathfrak{I}_{\text{drop}}$.

Thus, the insertion loss of a signal path can be calculated by the number of non-resonant and on-resonant ADFs in the path. For each master-slave pair (m_i,s_j), we introduce an integer variable $v^{\mathfrak{I}}_{i,j}$ to indicate the insertion loss in its signal path, and we introduce an integer variable $v^{\Lambda_{i,j}}$ to indicate the number of ADFs in the path. The calculation is formulated as follows:

$$v^{\mathfrak{I}}_{i,j}=v^{\Lambda_{i,j}}\cdot\mathfrak{I}_{\text{ADF}}+(1-b_0^{\lambda_{i,j}})\cdot(\mathfrak{I}_{\text{drop}}-\mathfrak{I}_{\text{ADF}}). \tag{14}$$

If $b_0^{\lambda_{i,j}}=1$, i.e. (m_i,s_j) does not rely on any ADF for demultiplexing, the insertion loss is calculated as the product of $v^{\Lambda_{i,j}}$ and $\mathfrak{I}_{\text{ADF}}$. Otherwise if $b_0^{\lambda_{i,j}}=0$, i.e. the signal resonates with an ADF in its path, one $\mathfrak{I}_{\text{ADF}}$ will be replaced by $\mathfrak{I}_{\text{drop}}$ in the calculation.

To model the number of ADFs in each signal path, we distinguish three cases for the signal paths:

3.5.1 Case 1: Default Path

A master communicates with its default slave through its default path, where the signal does not resonate with any ADF, as introduced in Section 2.2.4. In this case, the entry λ_{m_i,s_j} in the communication matrix will be set to zero, and the signal path consists of all ADFs in the i-th column and in the j-th row of the topology. This can be formulated as the following constraint:

$$v^{\Lambda_{i,j}} \geq \sum_{1\leq k\leq n_s} b^{\Lambda_{i,k}} + \sum_{1\leq k\leq n_m} b^{\Lambda_{k,j}} - M\cdot(1-b_0^{\lambda_{i,j}}), \tag{15}$$

where M is an extremely large auxiliary constant. If $b_0^{\lambda_{i,j}}=1$, i.e. s_j is the default slave of m_i, the above constraint counts all ADFs in the default path and assigns the value to $v^{\Lambda_{i,j}}$[4]. Otherwise if $b_0^{\lambda_{i,j}}=0$, the above constraint trivially holds, which means that this path model will not be applied to (m_i,s_j).

3.5.2 Case 2: Direct Demultiplexing

A master m_i communicates with its slave s_j via direct demultiplexing, if there is an ADF at the i-th column and the j-th row of the topology, as shown in Figure 5(b). In this case, the signal path consists of ADF_{m_i,s_j} and all ADFs that are above it (in the i-th column) or left to it (in the j-th row). This can be formulated as the following constraint:

$$v^{\Lambda_{i,j}} \geq \sum_{1\leq k\leq j} b^{\Lambda_{i,k}} + \sum_{1\leq k<i} b^{\Lambda_{k,j}} - M\cdot(1-b^{\Lambda_{i,j}}), \tag{16}$$

where M is an extremely large auxiliary constant. If $b^{\Lambda_{i,j}}=1$, i.e. there is an ADF for λ_{m_i,s_j}, the above constraint counts all ADFs in the signal path and assigns the value to $v^{\Lambda_{i,j}}$. Otherwise if $b^{\Lambda_{i,j}}=0$, the above constraint trivially holds, which means that this path model will not be applied to (m_i,s_j).

3.5.3 Case 3: Demultiplexing with Detour

A master m_i communicates with its slave s_j via demultiplexing with a detour, if the ADF for λ_{m_i,s_j} is removed by an ADF-sharing structure, as shown in Figure 5(c)(d). In this case, by denoting the shared ADF as $\text{ADF}_{m'_i,s'_j}$, the signal path can be considered as two parts: 1) from m_i to $\text{ADF}_{m'_i,s'_j}$, along the default path of m_i; and 2) from $\text{ADF}_{m'_i,s'_j}$ to s_j, along the default path of m'_i. This can be

[4] The '$\geq$' in this constraint implies '=', because $v^{\Lambda_{i,j}}$ will be assigned to its minimum allowable value by the minimization target.

formulated as the following constraint:

$$v^{\Lambda_{i,j}} \geq \sum_{1\leq k\leq n_s} b^{\Lambda_{i,k}} + \sum_{i'\leq k\leq n_m} b^{\Lambda_{k,j'}} + \sum_{j'<k\leq n_s} b^{\Lambda_{j',i'}} + \sum_{1\leq k\leq n_m} b^{\Lambda_{k,s_j}} - M\cdot b^{\Lambda_{i,j}} - M\cdot(1-b^{\text{share}}_{i,j,i',j'}), \quad (17)$$

where M is an extremely large auxiliary constant. This constraint is applied to all $\lambda_{m'_i,s'_j}$ that can potentially form an ADF-sharing structure with λ_{m_i,s_j}. If $b^{\Lambda_{i,j}}=0$ and $b^{\text{share}}_{i,j,i',j'}=1$, i.e. the ADF for λ_{m_i,s_j} does not exist and (m_i,s_j) shares an ADF with (m'_i,s'_j), the above constraint counts all ADFs in the signal path and assigns the value to $v^{\Lambda_{i,j}}$. Otherwise if either $b^{\Lambda_{i,j}}=1$ or $b^{\text{share}}_{i,j,i',j'}=0$, the above constraint trivially holds, which means that this path model will not be applied to (m_i,s_j) and $\text{ADF}_{m'_i,s'_j}$.

3.6 Removable Waveguide Loop

To reduce inevitable waveguide crossings outside ADFs, we tend to form removable waveguide loops in the topology, as introduced in Section 2.2.5. Suppose s_j is the default slave of m_i, a removable waveguide loop is formed when there is no ADF beneath the j-th row in the i-th column and there is no ADF right to the i-th column in the j-th row, as shown in Figure 6.

For each master-slave pair (m_i,s_j), we introduce a binary variable $b^{\text{loop}}_{i,j}$ to indicate whether its signal path contains a removable waveguide loop. The corresponding constraints are formulated as follows:

$$b^{\text{loop}}_{i,j} \leq b_0^{\lambda_{i,j}}, \quad (18)$$

$$M\cdot(1-b^{\text{loop}}_{i,j}) \geq \sum_{j+1\leq k\leq n_s} b^{\Lambda_{i,k}} + \sum_{i+1\leq k\leq n_m} b^{\Lambda_{k,j}}, \quad (19)$$

where M is an extremely large auxiliary constant. Constraint (18) ensures that a removable waveguide loop can only be formed if s_j is the default slave of m_i, and constraint (19) ensures that a removable waveguide loop will not be formed if there is any ADF in the loop.

We then introduce a binary variable $b^{\text{remove}}_{i,j}$ to indicate whether the waveguide crossing in the i-th column and in the j-th row of the topology is removable. A waveguide crossing is removable, when it is in a removable waveguide loop. This can be formulated by the following constraint:

$$b^{\text{remove}}_{i,j} \leq \sum_{1\leq k\leq j-1} b^{\text{loop}}_{i,k} + \sum_{1\leq k\leq i-1} b^{\text{loop}}_{k,j}. \quad (20)$$

3.7 Optimization Target

The optimization target of the model is described at the beginning of Section 2. Specifically, we introduce three more integer variables that should be minimized:

$v^{\Lambda_{\text{total}}}$ represents the number of ADFs in the topology, which is modelled as follows:

$$v^{\Lambda_{\text{total}}} = \sum_{1\leq i\leq n_m, 1\leq j\leq n_s} b^{\Lambda_{i,j}}. \quad (21)$$

$v^{\lambda_{\text{total}}}$ represents the number of wavelengths assigned to the ADFs, which is modelled as follows:

$$\forall 1\leq i\leq n_m \forall 1\leq j\leq n_s: \quad v^{\lambda_{\text{total}}} \geq \sum_{1\leq k\leq \omega} b_k^{\lambda_{i,j}}\cdot k. \quad (22)$$

And $v^{\mathfrak{I}_{\text{worst}}}$ represents the worst-case insertion loss among all signal paths, which is modelled as follows:

$$\forall 1\leq i\leq n_m \forall 1\leq j\leq n_s: \quad v^{\mathfrak{I}_{\text{worst}}} \geq v^{\mathfrak{I}}_{i,j}. \quad (23)$$

Besides, we introduce another integer variable v^{remove} to represent the number of removable waveguide crossings, which is modelled as follows:

$$v^{\text{remove}} = \sum_{1\leq i\leq n_m, 1\leq j\leq n_s} b^{\text{remove}}_{i,j}. \quad (24)$$

To reduce inevitable waveguide crossings in the final layout, we add maximizing v^{remove} to the optimization target.

Thus, the complete optimization objective can be formulated as follows:

$$\textbf{Minimize: } \alpha\cdot v^{\Lambda_{\text{total}}} + \beta\cdot v^{\lambda_{\text{total}}} + \gamma\cdot v^{\mathfrak{I}_{\text{worst}}} - \delta\cdot v^{\text{remove}},$$

where α, β, γ, and δ are constant weight coefficients that can be adjusted by the user to control optimization preference. v^{remove} is multiplied with -1 for maximization.

4 Experimental Results

We implement CustomTopo in C++, and solve the optimization model using Gurobi [24], a mixed integer linear programming (MILP) solver. The program is run on a computer with 2 Xeon processors under 2.67 GHz base frequency. The weight coefficients α, β, γ, and δ are assigned with 10, 10, 100, and 1, respectively[5]. The insertion loss parameters are from [16] and shown as follows:

Propagation loss	0.274dB/cm
Crossing loss ($\mathfrak{I}_{\text{cross}}$)	0.04dB
Through loss per MRR ($\mathfrak{I}_{\text{through}}$)	0.005dB
Drop loss ($\mathfrak{I}_{\text{drop}}$)	0.5dB

4.1 General Comparison

We tested CustomTopo on 7 test cases including: one small case (case 1) with high communication density; two medium-sized cases (case 2, 3) with medium communication density; one large case (case 4) with low communication density; and 3 cases with (semi-)symmetric communication graphs (case 5, 6, 7).

We compare against a state-of-the-art reduction method for topology customization, namely the λ-router reduction method proposed in [11]. We compare the topologies generated by CustomTopo with λ-router topologies before and after the reduction to demonstrate the benefits that we can expect from our customized topologies. Detailed test case information and comparison results are shown in Table 1[6].

In general, CustomTopo reduces the resource usage significantly, which also contributes to a remarkable reduction of the worst-case insertion loss:

- Compared with original λ-router topologies, for test cases that have high communication density (case 1 and case 5), the average ADF and wavelength usage are reduced by about 20% and 30%, respectively. The reduction becomes more significant as the communication density decreases: e.g. for the largest test case (case 4) that has sparse network communication (22 communicating master-slave pairs), the ADF usage is reduced from 120 to 10 and the wavelength usage is reduced from 16 to 6.
- The λ-router reduction approach [11] depends on the symmetry of the test cases. Case 5 and case 6 are semi-symmetric, and our customized topologies enable

[5] This assignment balances the optimization preferences among $v^{\Lambda_{\text{total}}}$, $v^{\lambda_{\text{total}}}$, and $v^{\mathfrak{I}_{\text{worst}}}$, since $v^{\mathfrak{I}_{\text{worst}}}$ differs from the others by one magnitude. v^{remove} is assigned with a smaller value since it is not the major optimization target.

[6] The reduction approach proposed in [11] focused on ADF and wavelength usage. The worst-case insertion loss $v^{\mathfrak{I}_{\text{worst}}}$ cannot easily be derived from our implementation of this method. Thus, $v^{\mathfrak{I}_{\text{worst}}}$ is only calculated for the last three test cases, which were proposed in the paper with explicit signal paths.

Table 1: Comparison among topology generation approaches

Index	Ref.	#N	#p	Method	$v^{\Lambda_{\text{total}}}$	$v^{\lambda_{\text{total}}}$	$v^{\Im_{\text{worst}}}$	Time
1	[20]	8	44	λ-router	28	8	0.85	–
				λ-reduction	24	8	–	
				CustomTopo	24	6	0.85	53s
2	[25]	12	26	λ-router	66	12	1.05	–
				λ-reduction	24	10	–	
				CustomTopo	13	7	0.8	184s
3	[26]	12	20	λ-router	66	12	1.05	–
				λ-reduction	15	9	–	
				CustomTopo	9	4	0.6	14s
4	[27]	16	22	λ-router	120	16	1.25	–
				λ-reduction	20	14	–	
				CustomTopo	10	6	0.7	13s
5 (sym.)	[11]	8	48	λ-router	28	8	0.85	–
				λ-reduction	20	7	0.75	
				CustomTopo	20	5	0.9	138s
6 (sym.)		8	24	λ-router	28	8	0.85	–
				λ-reduction	18	7	0.75	
				CustomTopo	12	6	0.8	3s
7 (full sym.)		8	24	λ-router	28	8	0.85	–
				λ-reduction	12	6	0.75	
				CustomTopo	12	6	0.8	63s

Ref.: reference of the test cases.
#N: the number of nodes.
#p: the number of communicating master-slave pairs.
$v^{\Lambda_{\text{total}}}$: the number of ADFs.
$v^{\lambda_{\text{total}}}$: the number of different wavelengths assigned to ADFs.
$v^{\Im_{\text{worst}}}$: the worst-case insertion loss resulted from ADFs.
Time: the program runtime denoted in seconds.

more wavelength reduction for both cases, and more ADF reduction for case 6. Case 7 is fully symmetric, and our customized topology shows the same reduction results as [11]. For other test cases (case 1-4), both the ADF usage and the wavelength usage is further reduced by about 40%.

- For small test cases (case 1, 5, 6, 7), the worst-case insertion loss in our customized topologies is similar or slightly higher than λ-router topologies. But for larger test cases, as the ADF usage is significantly reduced, we achieve a remarkable reduction in the worst-case insertion loss. A more detailed analysis of layout-aware insertion loss follows in Section 4.3, proving that the customized logic topology lends itself to better physical implementation as well.
- The program runtime varies among individual cases. But in general it shows a trend of positive correlation with the communication density. For most cases, the optimization terminates within 3 minutes.

4.2 Results Illustration

We illustrate the experimental results of test case 1[7] in Figure 7 to give an overview of the whole working process of CustomTopo.

Figure 7(a) shows the communication graph of case 1, which is given as the input to CustomTopo. Based on this input, CustomTopo constructs an optimized communication matrix where each entry indicates a wavelength, as shown in Figure 7(b). For each ADF-sharing structure in the communication matrix, CustomTopo removes one ADF from the initial topology and outputs an optimized logic topology, as shown in Figure 7(c). The logic topology can then be used for physical design with a state-of-the-art placement and routing tool, in this case we use PROTON+ [20], to determine the final layout of the ONoC, as shown in Figure 7(d).

[7] This test case consists of 4 hubs and 4 memory controllers with their location explicitly specified.

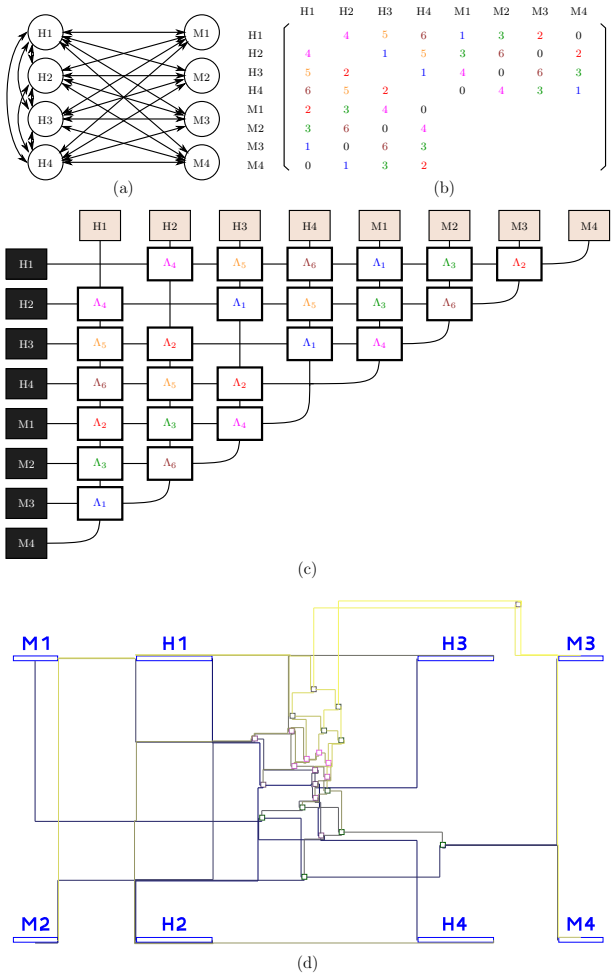

Figure 7: Illustration of test case 1. (a) Input communication graph. (b) Optimized communication matrix. (c) Optimized logic topology. (d) Final physical layout.

4.3 Discussion: Physical Layout

CustomTopo minimizes the worst-case insertion loss based on the logic topology. Since we do not make an assumption on the location of masters and slaves, the propagation loss and the crossing loss outside ADFs are not included in the optimization model. To investigate the insertion loss in the final layout, we feed CustomTopo topology for test case 1 to PROTON+ [20], a state-of-the-art physical design tool for WRONoCs, to compare it with the λ-router topology shown in Figure 8.

Comparing Figure 7(d) with Figure 8, we can see that ADFs in the λ-router layout are centralized in the middle, and ADFs in the CustomTopo layout are distributed in a larger area. A possible reason is that the signal paths in λ-router topology have similar lengths, and the signal paths in our customized topology have much different lengths. To minimize the worst-case insertion loss, PROTON+ optimizes the longest signal path as the first priority, which changes the centralized layout.

As shown in Table 2, CustomTopo layout saves 8 waveguides compared with λ-router layout, contributed by the ADF reduction. The waveguide reduction and the distributed layout feature then contribute to the remarkable 40+% re-

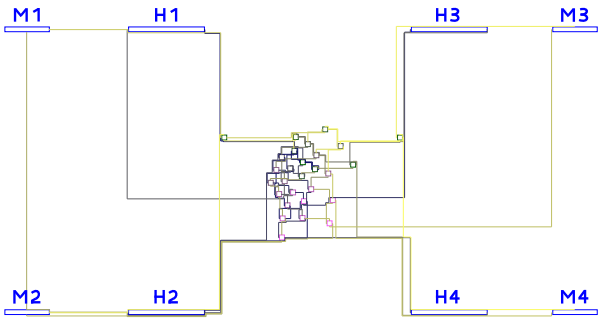

Figure 8: Physical layout generated from 8×8 λ-router topology by PROTON+.

Table 2: Physical features of final layouts

Physical feature	λ-router	CustomTopo
# waveguides (total)	64	56
# crossings (total)	90	51
# crossings (P_{worst})	40	23
Length of waveguides (total)	9.7344cm	10.7073cm
Length of waveguides (P_{worst})	1.0521cm	1.5732cm
Insertion loss (total)	7.67dB	5.98dB
Insertion loss (P_{worst})	2.79dB	2.08dB

P_{worst}: the signal path with the maximum insertion loss.

duction of total waveguide crossings and the maximum single-path waveguide crossings. A trade-off is that the waveguide length is larger in CustomTopo layout than in λ-router layout. But thanks to fewer waveguide crossings, both the total insertion and the maximum single-path insertion loss are reduced by more than 20%.

5 Conclusion

In this work, we propose CustomTopo, a general approach for WRONoC topology generation that optimizes the resource usage for application-specific designs. We analyze the topology generation problem from three aspects: wavelength usage, ADF usage, and insertion loss, and we propose reduction methods regarding each aspect. We implement our methods as an integer-linear-programming model, the program runtime for solving which positively correlates with the network communication density. CustomTopo is applicable to both symmetric and asymmetric networks. Specifically, for asymmetric networks, CustomTopo significantly outperforms the state-of-the-art method by 40% further reduction in wavelength and ADF usage.

6 References

[1] C. Manolatou and H. A. Haus, *Passive Components for Dense Optical Integration*. Springer, 2002.

[2] M. Brière, B. Girodias, Y. Bouchebaba, G. Nicolescu, F. Mieyeville, F. Gaffiot, and I. O'Connor, "System level assessment of an optical noc in an mpsoc platform," in *Proc. Design, Automation, and Test Europe Conf.*, 2007, pp. 1084–1089.

[3] 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 *Symp. Photonics and Optoelectronics (SOPO)*, 2011.

[4] E. Fusella, J. Flich, and A. Cilardo, "Path setup for hybrid noc architectures exploiting flooding and standby," *IEEE Trans. Parallel Distrib. Syst.*, vol. 28, no. 5, pp. 1403–1416, 2017.

[5] 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 *VLSI, 2009. ISVLSI '09. IEEE Computer Society Annual Symposium on*, 2009, pp. 19–24.

[6] 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 *Proc. Design Autom. Conf.*, 2010, pp. 657–660.

[7] S. L. Beux, J. Trajkovic, I. OâĂŹConnor, G. Nicolescu, G. Bois, and P. Paulin, "Optical ring network-on-chip (ornoc): Architecture and design methodology," in *Proc. Design, Automation, and Test Europe Conf.*, 2011, pp. 788–793.

[8] 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 *Proc. Int. Conf. Comput.-Aided Des.*, 2016, pp. 3:1–3:8.

[9] Z. Wang, Z. Pang, P. Yang, J. Xu, X. Chen, R. K. V. Maeda, Z. Wang, L. H. Duong, H. Li, and Z. Wang, "Moca: an inter/intra-chip optical network for memory," in *Proc. Design Autom. Conf.*, 2017, pp. 1–6.

[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 *Proc. Design, Automation, and Test Europe Conf.*, 2013, pp. 1589–1594.

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

[12] H. Omar and K. Hamwi, "Mhynesys ii: Multi-stage hybrid network on chip synthesis for next generation 3d ic manycore," in *IEEE International Symposium on Circuits and Systems*, 2013, pp. 325–328.

[13] K. Preston, N. Scherwood-Droz, J. S. Levy, and M. Lipson, "Performance guidelines for wdm interconnects based on silicon microring resonators," *CLEO: Science and Innovations*, 2011.

[14] P. Grani, R. Proietti, V. Akella, and S. J. B. Yoo, "Design and evaluation of awgr-based photonic noc architectures for 2.5d integrated high performance computing systems," in *High Performance Computer Architecture (HPCA)*, 2017, pp. 289–300.

[15] D. Vantrease, R. Schreiber, M. Monchiero, M. McLaren, N. P. Jouppi, M. Fiorentino, A. Davis, N. Binkert, R. G. Beausoleil, and J. H. Ahn, "Corona: System implications of emerging nanophotonic technology," *ACM SIGARCH Computer Architecture News*, vol. 36, no. 3, pp. 153–164, 2008.

[16] 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-chips: A formal study and comparison," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 23, no. 11, pp. 2552–2565, 2015.

[17] F. Liu, H. Zhang, Y. Chen, Z. Huang, and H. Gu, "Wavelength-reused hierarchical optical network on chip architecture for manycore processors," *IEEE Transactions on Sustainable Computing*, 2017.

[18] L. Ramini, D. Bertozzi, and L. P. Carloni, "Engineering a bandwidth-scalable optical layer for a 3d multi-core processor with awareness of layout constraints," in *IEEE/ACM International Symposium on Networks-on-Chip (NoCS)*, 2012, pp. 185–192.

[19] 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 *IEEE/ACM International Symposium on Networks-on-Chip (NoCS)*, 2016, pp. 1–8.

[20] A. V. 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, 2016.

[21] A. V. Beuningen and U. Schlichtmann, "PLATON: A force-directed placement algorithm for 3d optical networks-on-chip," in *Proc. Int. Symp. Phy. Des.*, 2016, pp. 27–34.

[22] D. König, "Über Graphen und ihre Anwendung auf Determinantentheorie und Mengenlehre," *Mathematische Annalen*, vol. 77, no. 4, pp. 453–465, 1916.

[23] S. L. Beux, H. Li, G. Nicolescu, J. Trajkovic, and I. O'Connor, "Optical crossbars on chip, a comparative study based on worst-case losses," *Concurrency and Computation: Practice and Experience*, 2014.

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

[25] D. Bertozzi, A. Jalabert, S. Murali, R. Tamhankar, S. Stergiou, L. Benini, and G. D. Micheli, "Noc synthesis flow for customized domain specific multiprocessor systems-on-chip," *IEEE Trans. Parallel Distrib. Syst.*, vol. 16, no. 2, pp. 113–129, 2005.

[26] A. Moonen, M. Bekooij, R. van den Berg, and J. van Meerbergen, "Practical and accurate throughput analysis with the cyclo static dataflow model," in *15th International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS)*, 2007, pp. 238–245.

[27] A. Alhonen, E. Salminen, L. Lehtonen, and T. D. Hämäläinen, "A scalable, non-interfering, synthesizable network-on-chip monitor – extended version," *Microprocessors and Microsystems*, vol. 37, pp. 446–459, 2013.