Physical and Logical Design
# of Flexible and Scalable
Wavelength-Routed Networks

Yukinobu Fukushima

Department of Information Networking
Graduate School of Information Science and Technology
Osaka University

January 2006

Preface

The increase in the number of Internet users and the appearance of multimedia services such as video streaming have led to the rapid growth in Internet traffic. The answer to meet the bandwidth demand is centered on a new emerging technology, WDM (Wavelength Division Multiplexing). WDM allows transmitting multiple optical signals on different wavelengths on a single fiber, which dramatically increases the link bandwidth. The rapid advances in optical technologies make it possible to perform all-optical wavelength-routing. A wavelength-routed network, where optical signals on wavelengths are switched without electronic conversion, is a promising candidate for the next generation transport network.

Design of wavelength-routed networks has been one of hot topics. In conventional design methods for wavelength-routed networks, minimizing network equipment cost needed to accommodate a given traffic or maximizing the network throughput is main objective. In a real world, however, the advent of popular World Wide Web servers or data centers drastically affect traffic pattern. More significantly, there are various types of data traffic such as video streams and P2P traffic with different traffic characteristics, which introduce several traffic patterns. Therefore *flexibility* against those changing traffic patterns is an important property of wavelength-routed networks.

Lightpath provisioning across multiple domains has recently been discussed. The standardization of routing protocols for inter-domain wavelength-routed networks will accelerate the increase in the scale of wavelength-routed networks. As a result, the numbers of nodes and multiplexed wavelengths increase. These cause scalability problems such as the increase in routing table size and the increase in the number of optical fiber amplifiers. In addition, the diversity in the number of multiplexed wavelengths occurs as the scale of those networks increases because the number of wavelengths multiplexed within each domain is individually determined. Therefore edge nodes between links with different numbers of wavelengths multiplexed need to cover the difference.

In this thesis, we first propose a design method of flexible WDM physical network. Many conventional designing scheme of WDM networks assume that the future traffic demand is known beforehand. However, it is difficult to predict the future traffic demand accurately. Therefore we develop a design method of a WDM network that accommodates as much traffic as possible against a variety of traffic patterns, that is, a flexible WDM network. Our basic idea is to select a node-pair that is expected to be a bottleneck in the future, and then to deploy network equipments so that the volume of traffic accommodated by the node-pair increases. The results in our simulation show that the WDM network designed by our method can accommodate more traffic demand than those designed by the existing methods with the same cost.

We next focus on a logical topology design problem in large-scale wavelength-routed networks where hundreds or thousands of wavelengths are multiplexed. In conventional researches, it is assumed that a constant number of wavelengths be available on each fiber. But it is not necessary to utilize all wavelengths on each fiber in building a logical topology. Instead, several wavebands, which include a set of wavelengths amplified by an optical amplifier, may be considered for introduction while deploying additional wavebands and their corresponding optical amplifiers when additional wavelengths are actually required. In this case, the number of wavelengths available on the respective fibers depends on the number of optical fiber amplifiers deployed on each fiber. Therefore we propose a heuristic algorithm for the design of a logical topology with as few optical fiber amplifiers as possible. Our results indicate that our algorithm reduces the number of optical fiber amplifiers with a slight increase of average packet delays.

We further discuss the design of edge nodes, to which links with different numbers of wavelengths are connected, in large-scale wavelength-routed networks. We need to cope with the difference in the numbers of wavelengths by wavelength conversion on edge nodes. In previous researches for wavelength converter placement problem, the main purpose is eliminating fragmentation of wavelength resources between adjacent links that have the same number of wavelengths multiplexed. In large-scale wavelength-routed networks, however, we also need to utilize wavelength converters to cover the difference in the numbers of multiplexed wavelengths. In this part, we propose an edge node architecture that has fixed wavelength converter to solve the above-mentioned difference. This architecture offers total cost reduction at the edge nodes.

At last, we propose a node-clustering method for hierarchical routing. Hierarchical routing scales well by yielding enormous reductions in routing table length, but it can also increase blocking probability because longer paths in hierarchical routing tend to have less free wavelength channels. However, if the routes assigned to longer paths have greater wavelength resources, we can expect that the blocking probability will not increase. Therefore we propose a distributed node-clustering method that maximizes the number of lightpaths between nodes. The key idea behind our method is to construct node-clusters that have much greater wavelength resources from the ingress border nodes to the egress border nodes, which increases the wavelength resources on the routes of lightpaths. We evaluate the blocking probability for lightpath requests and the maximum table length in simulation experiments. We find that the method we propose significantly reduces the table length, while the blocking probability is almost the same as, or even lower than that without clustering.

Acknowledgements

First and foremost, I would like to express my sincere appreciation to Prof. Masayuki Murata of Osaka University for introducing me to the area of optical networking. His creative suggestions, insightful comments, and patient encouragement have been essential for my research activity. I also thank him for providing me with the opportunity to research with a talented team of researchers.

I am heartily grateful to the members of my thesis committee, Prof. Koso Murakami, Prof. Makoto Imase of Osaka University, and Prof. Tokumi Yokohira of Okayama University, for reading my dissertation and providing many valuable comments.

I would like to thank Dr. Hideo Miyahara, President of Osaka University, for getting me interested in the field of computer networks.

I am also deeply grateful to Dr. Shin'ichi Arakawa of Osaka University for his much appreciated comments and support. His kindness on my behalf were invaluable, and I am forever in debt. My thanks also go to Dr. Hiroaki Harai of NICT (National Institution of Information and Communications Technology). His expertise and insightful comments have been invaluable.

I would like to thank Prof. Naoki Wakamiya and Prof. Go Hasegawa of Osaka University for enlightening discussions. I am thankful to my friends in the department for their inciting discussions and fellowship.

Last, but not least, I thank my parents for their invaluable support and constant encouragement during my undergraduate and doctoral studies.

List of Papers

Journal Papers

1. Y. Fukushima, H. Harai, S. Arakawa, and M. Murata, “Design of wavelength-convertible edge nodes in wavelength-routed networks,” to appear in *OSA Journal of Optical Networking*, 2006.

2. Y. Fukushima, S. Arakawa, and M. Murata, “Design of logical topology with effective waveband usage in IP over WDM networks,” to appear in *Photonic Network Communications*, 2006.

3. Y. Fukushima, H. Harai, S. Arakawa, and M. Murata, “A distributed clustering method for hierarchical routing in large-scaled wavelength routed networks,” *IEICE Transactions on Communications*, vol. E88-B, pp. 3904–3913, Oct. 2005.

4. Y. Fukushima, H. Harai, S. Arakawa, and M. Murata, “On the robustness of planning methods for traffic changes in WDM networks,” *OSA Journal of Optical Networking*, vol. 4, pp. 11–25, Jan. 2005.

Refereed Conference Papers

1. Y. Fukushima, H. Harai, S. Arakawa, and M. Murata, “Distributed clustering method for large-scaled wavelength routed networks,” in *Proceedings of 2005 IEEE Workshop on High Performance Switching and Routing (HPSR)*, May 2005.

2. Y. Fukushima, H. Harai, S. Arakawa, and M. Murata, “Planning method of robust WDM networks against traffic changes,” in *Proceedings of Optical Network Design*

and Modeling 2004 (ONDM2004), pp. 695–714, Feb. 2004.

3. Y. Fukushima, H. Harai, S. Arakawa, M. Murata, and H. Miyahara, “A minimum interference routing algorithm for multi-period planning of WDM lightpath networks without traffic prediction,” in *Proceedings of 28th European Conference on Optical Communication (ECOC 2002)*, Sept. 2002. P4.9.

4. Y. Fukushima, S. Arakawa, M. Murata, and H. Miyahara, “A design method for logical topologies with consideration of wavebands,” in *Proceedings of Optical Network Design and Modeling 2002 (ONDM2002)*, Feb. 2002.

Non-Refereed Technical Papers

1. Y. Fukushima, H. Harai, S. Arakawa, and M. Murata, “Deployment of wavelength converters in wavelength-routed overlay networks,” *Technical Report of IEICE,* (IN2005-90) *(in Japanese)*, pp. 13–18, Oct. 2005.

2. Y. Fukushima, H. Harai, S. Arakawa, and M. Murata, “Node clustering method for hierarchical routing in WDM lightpath networks,” *Technical Report of IEICE,* (NS2004-172) *(in Japanese)*, pp. 1–6, Dec. 2004.

3. Y. Fukushima, H. Harai, S. Arakawa, M. Murata, and H. Miyahara, “Planning and design methods for WDM networks robust against traffic changes,” *Technical Report of IEICE,* (PS2003-3) *(in Japanese)*, pp. 11–16, Apr. 2003.

4. Y. Fukushima, H. Harai, S. Arakawa, M. Murata, and H. Miyahara, “An enhanced minimum interference routing algorithm for multi-period planning of WDM lightpath networks without traffic prediction,” *Technical Report of IEICE,* (IN2002-34) *(in Japanese)*, pp. 7–12, July 2002.

5. Y. Fukushima, S. Arakawa, M. Murata, and H. Miyahara, “Design of logical topologies in consideration of available wavebands,” *Technical Report of IEICE,* (NS2001-67) *(in Japanese)*, pp. 33–38, July 2001.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

The increase in the number of Internet users and the appearance of multimedia services such as video streaming have led to the rapid growth in Internet traffic. Until now, Internet traffic doubles each year [1]. New services such as grid computing, P2P will drive the traffic growth and bandwidth demands in the future. The answer to meet the bandwidth demand is centered on a new emerging technology, WDM (Wavelength Division Multiplexing). WDM allows transmitting multiple optical signals on different wavelengths on a single fiber, which dramatically increases the link bandwidth. For example, a commercially available WDM system offers 1.76 Tbps of link transmission capacity by multiplexing 176 wavelengths, each of which carries 10 Gbps [2].

WDM technology has only been used for increasing link capacities in point-to-point WDM transmission systems. In those systems, each wavelength in a fiber is treated as a physical link between network components such as electronic routers. Upon transmitting data from an electronic router, electronic signals are converted to optical signals having different wavelengths. The MUX multiplexes several optical signals into an optical fiber, and then optical signals arrive at the next electronic router. In WDM networks that deploy the point-to-point transmission systems, each electronic router on WDM nodes performs packet routing in the electronic region as shown in Fig. 1.1. The point-to-point WDM network certainly increases link capacities, however, since packet routing is still required

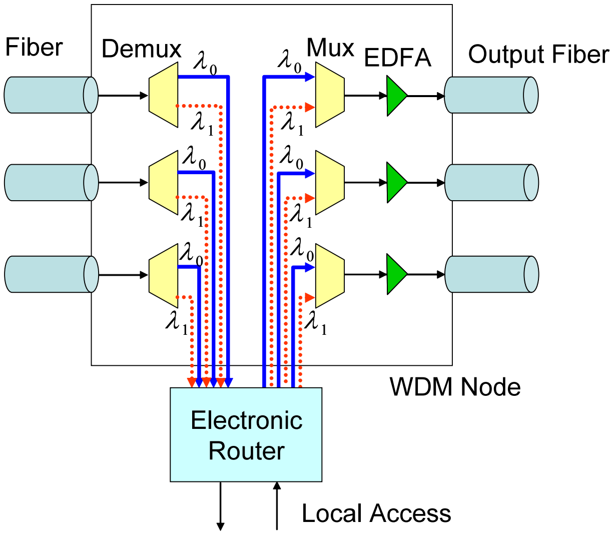


Figure 1.1: WDM node in point-to-point WDM network

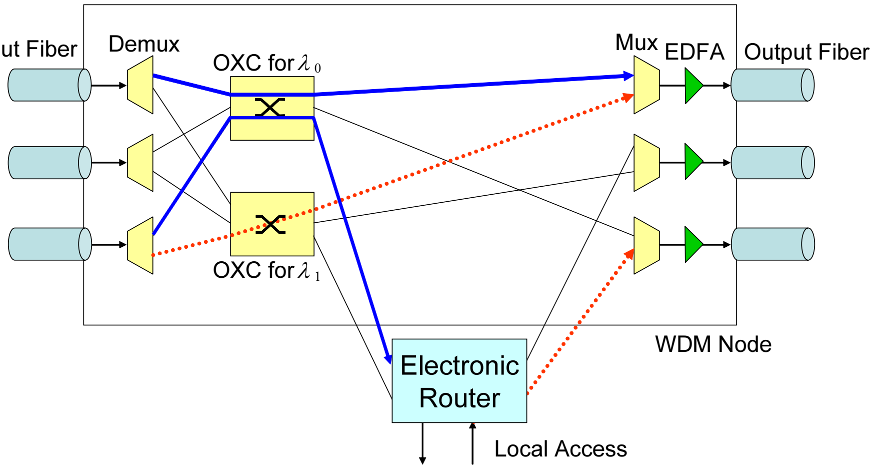


Figure 1.2: WDM node in wavelength-routed network

at all the nodes, the bottleneck of network just shifts from the transmission capacity of link to the processing capacity of electronic routers.

The rapid advances in optical technologies make it possible to perform all-optical wavelength-routing. Figure 1.2 shows a WDM node in wavelength-routed networks. Each WDM node is equipped with OXCs (Optical Cross-Connects) that switch an input wavelength to an appropriate output port without electronic conversion and electronic processing. Configuring OXCs on each intermediate node along the path enables to set up an optical channel (called *lightpath* [3]) that is an all-optical connection from source node to destination node (Fig. 1.3). In wavelength-routed networks, since optical signals are not converted into electronic signals on intermediate nodes, the load of electronic routes on intermediate nodes are relieved.

Design of wavelength-routed networks has been one of hot topics. The design problem consists of (1) the design of physical WDM networks (i.e., determine the number of OXCs, fibers, wavelength converters, amplifiers and other equipments) and (2) the design of *logical topology* (i.e., determine the routing and wavelength assignment for the lightpaths). In conventional design methods for wavelength-routed networks, minimizing network equipment cost needed to accommodate a given traffic or maximizing the network throughput is main objective. In a real world, however, the advent of popular World Wide Web servers or data centers drastically affect traffic pattern. More significantly, there are various types of data traffic such as video streams and P2P traffic with different traffic characteristics, which introduce several traffic patterns. Thus *flexibility* is an important property of wavelength-routed networks.

Lightpath provisioning across multiple domains has recently been discussed. Routing protocols that target for inter-domain wavelength-routed networks such as OBGP [4] based on BGP have recently been investigated [4, 5, 6]. The standardization of the routing protocol for inter-domain wavelength-routed networks will accelerate the increase in the scale of wavelength-routed networks. As a result, the numbers of nodes and multiplexed wavelengths increase. These cause scalability problems such as the increase in routing table size and the increase in the number of optical fiber amplifiers. In addition, the diversity in the number of multiplexed wavelengths occurs as the scale of those networks increases because the number of wavelengths multiplexed within each domain is individually determined. Therefore edge nodes between links with different numbers of wavelengths multiplexed need to cover the difference.

From above discussions, we need to cope with the following issues in designing wavelength-routed networks.

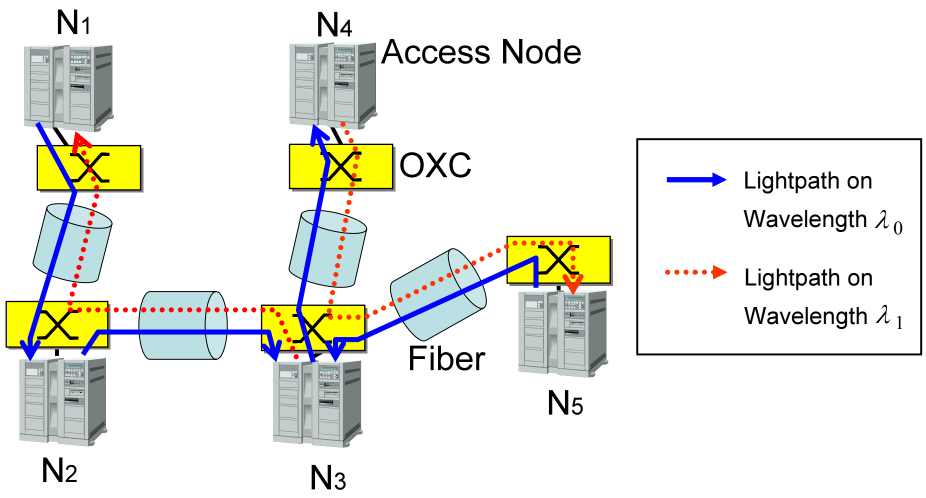


Figure 1.3: Wavelength-routed network

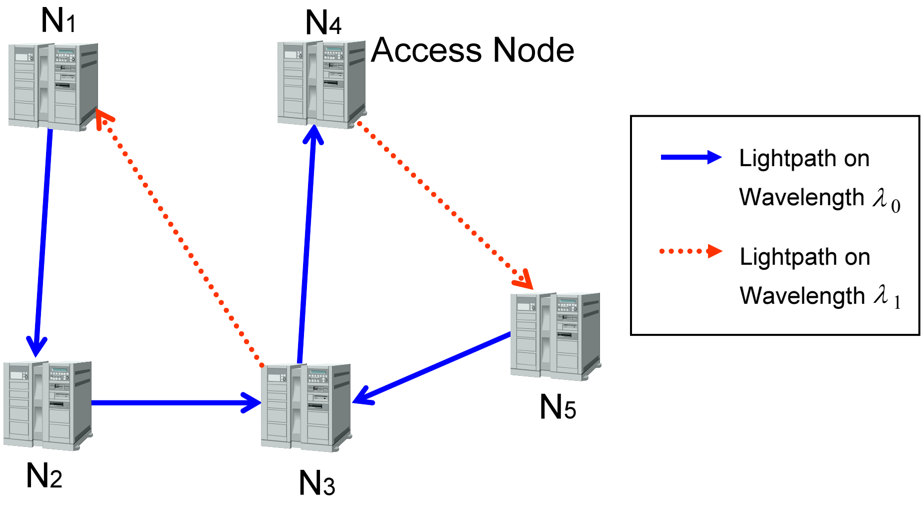


Figure 1.4: Logical topology

1.2 Design Issues for Wavelength-Routed Networks

1.2.1 Logical Topology Design

Logical topology is a graph consisting of lightpaths. An example of a logical topology is shown in Fig. 1.4. N_1 and N_3 can be logically connected in the logical topology while they are not in the physical topology. The logical topology is viewed as an underlying network by upper layer protocols. For example, when the upper layer protocol is IP, IP routing is performed on the logical topology. Because the number of lightpaths that are simultaneously going through an optical fiber is limited by the number of available wavelengths, the construction of the logical topology must be paid much attention for optimizing an objective performance. The design method of logical topology has been studied extensively in this research area [7, 8, 9]. They investigate routing and wavelength assignment algorithms for lightpaths, which we call RWA (Routing and Wavelength Assignment) problem. In the RWA problem, given (1) a physical WDM network, (2) a traffic matrix that expresses the static traffic demand for the network, and (3) constraints such as the number of wavelengths multiplexed on a fiber, we determine the route and the wavelength to be assigned to the lightpath of each traffic demand such that an objective function is optimized (e.g., throughput [9] or the number of wavelengths used in wavelength-routed networks [10]). For example, the authors in [9] formulated the logical topology design problem as an integer linear programing (ILP) and showed that the problem is NP-hard. A heuristic algorithm combining simulated annealing and flow deviation was proposed to minimize the average packet delay or to maximize the throughput.

1.2.2 Physical Topology Design

In the design of wavelength-routed networks, we determine how many equipments such as OXCs to deploy in order to optimize an objective function such as the cost of network equipment, given (1) a location candidate for network equipments, (2) a traffic matrix, and (3) constraints. The design of physical WDM network is solved in combined with the logical topology design; we first constructs the logical topology (i.e., determine the routes and wavelengths for the given traffic), then deploy network equipments needed to set up those lightpaths. Various methods for physical WDM network design have been studied [11, 12, 13]. For example, the authors in [12] formulated the physical WDM network design problem as an ILP and solved the problem with the branch and bound method.

1.2.3 Design of Node-Clusters for Hierarchical Routing

Introducing hierarchical routing [14] is indispensable to improve the scalability in terms of the number of nodes in wavelength-routed networks. The basic idea behind hierarchical routing is to form a set of nodes into a *cluster* to aggregate route information about nodes far from a source node. Each node has complete route information about nodes in the same

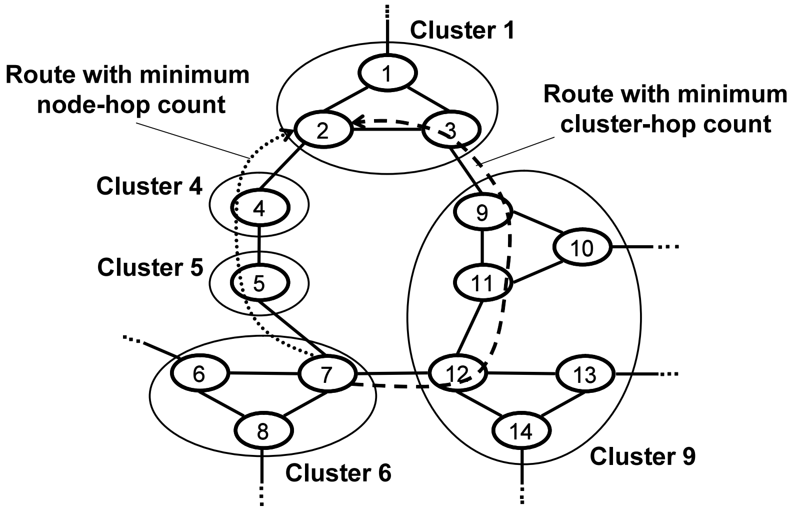

Cluster 1
Route with minimum node-hop count
Route with minimum cluster-hop count
1
2
3
Cluster 4
4
9
10
Cluster 5
5
11
6
7
12
13
8
14
Cluster 9
Cluster 6

Figure 1.5: Routes with/without hierarchical routing

cluster (i.e., intra-cluster route) and also has aggregated route information about nodes in the other clusters (i.e., inter-cluster route). Therefore, the routing table size is reduced.

Although hierarchical routing reduces the size of the routing table, it generally increases the path length. The main reason is that inter-cluster routes cannot always be the same routes as those in a non-clustered environment. That is, path length is increased when an inter-cluster route with a minimum cluster-hop count differs from the shortest path with a minimum node-hop count. In Fig. 1.5, the path length of the minimum node-hop route from node 7 to node 2 is 3 while the path length of the minimum cluster-hop route is 5. This increased path length is likely to increase the blocking probability for lightpath requests because the probability of finding wavelengths idle on the path decreases as the path length increases. Therefore, it is important to construct clusters to minimize the blocking probability.

1.3 Outline of Thesis

As we pointed out in Section 1.1, we need to consider how we can design a wavelength-routed network that flexibly accommodates the changing traffic. Other important considerations in large-scale wavelength-routed networks are (1) the scalability in terms of the number of nodes and the number of wavelengths multiplexed, (2) the diversity in the number of wavelengths multiplexed. In the rest of this section, we summarize the objectives of this thesis and refer to other related works in the literature.

Physical Design of Flexible Wavelength-Routed Networks [15, 16, 17, 18, 19]

Various design methods for minimizing the cost of the physical network have been studied [11, 12, 13]. In those studies, they design the WDM networks based on an explicit knowledge of the traffic demand (e.g., a typical traffic demand occurring during the period). While we may be able to estimate total traffic demand in the near future (e.g., Internet traffic doubles each year [1]), in practice, it is difficult to predict traffic pattern of each source-destination pair, because the advent of popular World Wide Web servers or data centers has drastically affected traffic demand and traffic pattern. More significantly, there are various types of data traffic such as video streams and P2P traffic with different traffic characteristics, which introduce several traffic patterns during the period. Therefore, conventional design methods using a single traffic pattern are inadequate to deal with the unpredictable traffic.

In Chapter 2, we propose a scheme for designing flexible WDM networks. Here, we consider a network is flexible when the network can accommodate a variety of traffic patterns. Our basic idea is to design a network accommodating several predicted traffic patterns that follow a certain distribution such as normal or exponential distribution. For each traffic pattern, we select node-pairs that become the bottleneck in accommodating the traffic and deploy network equipments for the node-pairs. By examining various traffic patterns, we expect that nodes that are likely to be the bottleneck get more equipments while nodes that are less likely to be the bottleneck get less equipments, which leads to constructing flexible and cost-effective networks.

Our design method incrementally extends the size of OXCs and increases the number of fibers until the designed network has the ability to accommodate a variety of traffic patterns. We handle the incremental operations based on an algorithm to which we modify the ADD algorithm [20]. The modified algorithm addresses two design problems of flexible networks; the *OXC-deployment problem* and the *fiber-deployment problem*. The *OXC-deployment problem* involves determining how many ports of each OXC are needed to design a flexible WDM network. In this problem, we first identify the node-pair with bottleneck, which is determined by obtaining the *maximum flow value* of each node-pair. Then, we upgrade an OXC on a node so that upgrading it leads to maximizing the maximum flow value of the node-pair with bottleneck. We also try to design a flexible WDM network based on the maximum flow value in the *fiber-deployment problem*, in which a number of dark fibers are leased. We determine where to set up lightpaths and where to lease optical fibers. There are various routing algorithms that determine the route of lightpaths. For instance, we may be able to accommodate as much traffic demand as possible without a priori knowledge of future traffic demand by utilizing MIRA (Minimum Interference Routing Algorithm) [21] and MOCA (Maximum Open Capacity Routing Algorithm) [22]. However, these two algorithms need physical topology as an input parameter and we cannot directly utilize them in our fiber-deployment problem, because the physical topology is not input information but output information in our problem. Thus, we also propose a routing and fiber/wavelength assignment algorithm that we call EMIRA (Enhanced Minimum Interference Routing Algorithm).

Design of Logical Topology with Effective Usage of Wavebands [23, 24, 25]

A lot of works have dealt with methods for the design of the logical topology [7, 8]. Most of these works have been based on the assumption that a constant number of wavelengths is available on each fiber, and then minimize the congestion of the network [8]. In the design of a cost-effective network, however, it is preferable to provide only the wavelengths that are actually needed on the fibers. Utilizing a constant number of wavelengths requires installing all kinds of amplifiers for the entire spectral range. On the other hand, we can minimize the number of optical amplifiers by deploying them only on fibers that are short of wavelengths. For this purpose, we need a new way of designing the logical topology such that it minimizes the number of optical amplifiers while meeting the demands imposed by traffic.

In Chapter 3, we propose an algorithm called MALDA (Minimum number of fiber Amplifiers Logical topology Design Algorithm) for designing a logical topology. This algorithm is in contrast to earlier approaches in that it minimizes the deployment of optical fiber amplifiers on the fiber under the constraint that the load of all the nodes should be kept under their processing capacity.

Design of Edge-Nodes with Effective Wavelength Conversion [26, 27]

Wavelength conversion improves the blocking performance of wavelength-routed networks. Wavelength converters change an input wavelength to another output one, thus eliminate the fragmentation of wavelength resource. Because wavelength converters remain expensive in the near future, we need to minimize the number of wavelength converters deployed for achieving an objective performance. In order to cost-effectively utilize wavelength converters, methods for deployment of wavelength converters are developed. In [28, 29], deploying wavelength converters only on a few nodes leads to the cost reduction. In [30], introducing about 1–5% of all wavelength converters to a part of ports on a few nodes achieves the blocking performance close to full-complete wavelength conversion where all ports on all nodes are equipped with wavelength converters.

In conventional researches [28, 29, 30], they focus on networks where each link has an identical number of wavelengths multiplexed. In those networks, wavelength converters are used for eliminating fragmentation of wavelength resources between adjacent links that have the same number of wavelengths multiplexed. In a wavelength-routed overlay network where there is diversity in the numbers of wavelengths multiplexed on links, however, we also need to utilize wavelength converters to cover the difference between the numbers of wavelengths on links. A wavelength-routed overlay network consists of a carrier network, end hosts, and access links. An intra-carrier network is maintained by a carrier. End users prepare access links with a few wavelengths multiplexed for cost reduction. On the other hand, A carrier prepares a core link with tens or hundreds of wavelengths multiplexed for accommodating traffic from access links. As a result, a difference in the numbers of wavelengths multiplexed on access and core links occurs. We need to cope with this difference with wavelength converters.

In Chapter 4, we first show that edge nodes, to which both access and core links are attached, need much more wavelength converters than core nodes, to which core links are only attached. Then, we propose an ingress edge node architecture with fixed wavelength converters that convert a predetermined input wavelength to another predetermined output wavelength. In our node architecture, fixed wavelength converters evenly distributes wavelengths from input access links to wavelengths on an output core link. Adopting fixed wavelength converters for distribution of input wavelengths leads to lower costs than nodes with full wavelength converters that convert any input wavelengths to another output one.

Design of Node-Clusters for Scalable Wavelength Routing [31, 32, 33]

Many researchers have investigated the routing and wavelength reservation protocols for establishing lightpaths in intra-domain networks. Routing and wavelength reservation protocols that target for the inter-domain network have recently been investigated [5, 6, 4, 34]. Bernstein *et al.* [5] specified key requirements for inter-domain routing protocols for optical networks. One of these is the “independence of the internal domain control plane mechanism”. Routing and wavelength reservation protocols in the inter-domain network are independent of protocols in the intra-domain network. BGP is the only existing protocol that conforms to these requirements and is widely deployed in the current Internet. We can use a BGP that is extended to wavelength-routed networks (e.g., Optical BGP [4]) as the inter-domain routing and wavelength reservation protocol.

Li *et al.* [35] pointed out that BGP lacks scalability of number of routes, which results from the increased number of nodes. This is because the BGP router’s memory size limits the routing table size and therefore BGP will not work with a large number of routes. One promising approach to keeping the routing table size scalable is to introduce *hierarchical routing* [14]. The basic idea behind hierarchical routing is to form a set of nodes into a *cluster* to aggregate route information about nodes far from a source node. Each node has complete route information about nodes in the same cluster (i.e., intra-cluster route) and also has aggregated route information about nodes in the other clusters (i.e., inter-cluster route). Therefore, the routing table size is reduced.

Although hierarchical routing reduces the size of the routing table, it generally increases the path length. The main reason is that inter-cluster routes cannot always be the same routes as those in a non-clustered environment. That is, path length is increased when an inter-cluster route with a minimum cluster-hop count differs from the shortest path with a minimum node-hop count (Fig. 1.5). This increased path length is likely to increase the blocking probability for lightpath requests because the probability of finding wavelengths idle on the path decreases as the path length increases. Therefore, it is important to construct clusters to minimize the blocking probability.

In Chapter 5, we propose a method of clustering in a distributed manner to minimize the blocking probability for lightpath requests. To achieve this, we maximize the number of lightpaths between nodes. The key idea behind our method is to construct the node-clusters that have many wavelength resources from ingress border nodes to egress border nodes, which increases wavelength resources on the routes of lightpaths. We expect the increased number of available lightpaths would lead to decreased blocking probability. Our method is a distributed clustering algorithm that is suited to large-scale wavelength-routed networks.

Finally, in Chapter 6, we conclude this thesis.

Chapter 2

Physical Design of Flexible Wavelength-Routed Networks

In conventional researches for the design of WDM networks, they assume that the future traffic demand is known beforehand. However, it is difficult to predict the future traffic demand accurately. Therefore, in this chapter, we develop a design method of a WDM network that accommodates as much traffic as possible against a variety of traffic patterns, that is, a flexible WDM network. Our basic idea is to select a node-pair that is expected to be a bottleneck in the future, and then to deploy network equipments so that the volume of traffic accommodated by the node-pair increases. The results in our simulation show that the WDM network designed by our method can accommodate more traffic demand than those designed by the existing methods with the same cost.

2.1 Network Model

Our WDM network model consists of both physical and logical topologies. The WDM physical topology is the actual network which consists of WDM nodes, WDM transmission links, and electronic routers. Each WDM node equips with MUXs/DEMUXs (multiplexers and demultiplexers) and OXCs as depicted in Fig. 2.1. The incoming multiplexed signals are divided into each wavelength at a DEMUX. Then, each wavelength is routed to an OXC. The OXC switches the incoming wavelength to the corresponding output port.

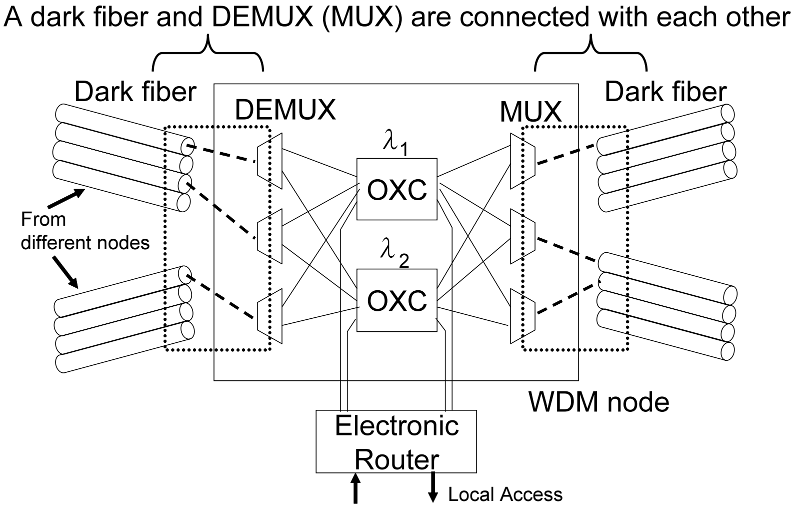


Figure 2.1: WDM node architecture

Finally, wavelengths routed to a MUX are multiplexed and transmitted to the next node. An OXC also switches wavelength from/to electronic routers to provide add/drop functions. We do not consider waveband switching [36] which decreases the number of OXCs. Wavelength conversion is not allowed at WDM nodes. As illustrated in Fig. 2.1, the number of optical fibers between two WDM nodes (optical fibers connected to MUXs/DEMUXs) may not be identical.

As we mentioned, we will design a WDM network robust against traffic changes. Our design scheme can be utilized by network designers (e.g. service providers) who deploy WDM nodes by themselves and lease dark fibers from carriers. Since the network designers are likely to decrease equipment cost, we use minimum size (in terms of the number of ports) of OXCs at WDM nodes and a minimum number of optical fibers in links to design a robust WDM network. In doing so, we develop incremental approach, which will be described in Section 2.3. Initially, we prepare small size OXCs and candidate fiber locations (i.e., links) for our design. The dark fibers are connected to available DEMUXs/MUXs as long as there are available ports at the OXC. The connection of dark fibers are allowed on the candidate fiber locations but the number of dark fibers to be leased is not limited. As for OXCs, we use OXCs with the discrete number of ports (e.g., 4×4, 8×8, and 16×16 OXCs). We assume that the number of multiplexed wavelengths is identical among all optical fibers.

We introduce the following restrictions on how to deploy OXCs to simplify maintenance for the network operator.

- We deploy one non-blocking OXC for each wavelength on each WDM node. For instance, when we require OXC with 8 ports to establish 8 lightpaths for each wavelength, we deploy an 8×8 OXC instead of two 4×4 OXCs. As a result, we can decrease the number of OXCs which the operators should maintain.
- We make the number of OXC ports for each wavelength on a WDM node identical. When we increase the number of OXC ports for a wavelength, we also add the same number of ports for the other wavelengths on the node.

2.2 Flexibility of Physical Topology

2.2.1 Measure of Flexibility: Robustness against Traffic Changes

Conventional design methods for WDM networks assume that traffic demand is predictable. However, in practice, because it is very difficult to precisely predict what this will be in the future, we should design a network that can accommodate this expected demand without getting involved with precise predictions. One promising way to design such a network is to deploy redundant resources to all links and nodes, that is, to introduce excess resources $X\%$ more than the required quantity. However, this approach tends to result in high-cost networks since overall traffic demand seldom exceeds the predicted demand.

Instead of preparing redundant resources, we try to design a network accommodating several predicted traffic patterns that follow a certain distribution, such as normal or exponential distribution. A real problem is that we have no ways of knowing which distribution the traffic will follow. We assume that the discrepancy between the volume of traffic actually occurring and the predicted volume will follow a normal distribution. Then we design a robust network based on this assumption by ensuring that the designed network will accommodate the traffic change that follows this distribution. Here, we define the traffic change as the error between predicted traffic volume and the volume of the traffic actually occurring. Note that "traffic change" does not refer to the change of traffic demand in a short time; for example, the difference between the volume of traffic in day-time and the volume of traffic at night.

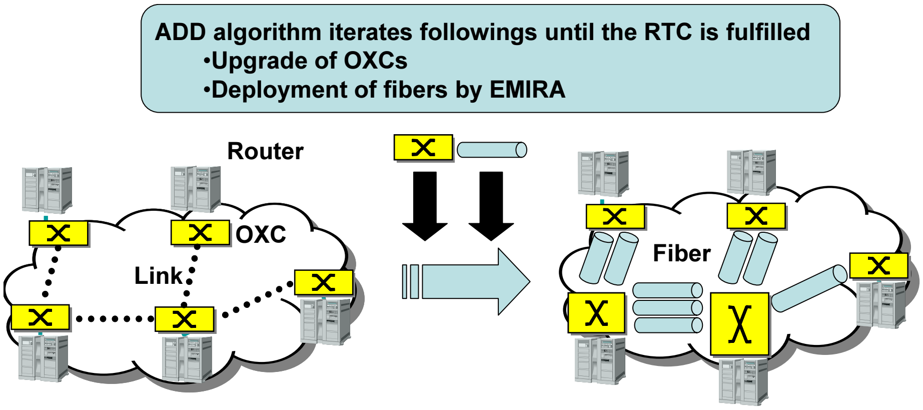
Our scheme generates a set of traffic demand based on a predicted traffic with prediction error assumed to follow a normal distribution and utilizes it as an input parameter of the WDM network design problem. Each traffic demand is expressed as a traffic matrix. The traffic matrix consists of the volume of traffic demand each node-pair requests ($\boldsymbol{T} = \{t_{ij}\}$). Given μ_{ij}, the average volume of traffic that node-pair (i, j) in a predicted traffic matrix requests, and σ_{ij}, the standard deviation which determines how much the traffic changes, our method generates $(K - 1)$ traffic matrices ($\boldsymbol{T}_k = \{t^k_{ij}\},\ k = 1, 2, \ldots, K - 1$). t^k_{ij} is a value of the random variable that follows a normal distribution $N(\mu_{ij}, (\sigma_{ij})^2)$. $\boldsymbol{T}_0 = \{\mu_{ij}\}$ and $\boldsymbol{\Sigma} = \{\sigma_{ij}\}$ are input parameters of the network design problem. $\boldsymbol{T}_0$ expresses the predicted traffic demand. $\boldsymbol{\Sigma}$ is a matrix consisting of σ_{ij}. The values for σ_{ij} will be selected based on the statistical measurement of the traffic change in the past and the network designer's judgment. However, how we should select those values is out of scope of this thesis.

Our method defines the condition robust WDM networks need to fulfill to individually accommodate all the K traffic matrices, which consists of $(K-1)$ generated traffic matrices and the predicted traffic matrix. This condition is called RTC (Robustness against Traffic Changes). Networks with RTC can accommodate traffic matrices changing within the range specified by $\boldsymbol{\Sigma}$ and K. When the traffic change does not actually follow a normal distribution, we believe that our method can accommodate the traffic demand by utilizing the obtained distribution as input information instead of a normal distribution.

2.2.2 Design Problems of Flexible WDM Network

In our design method, we deploy optical components (i.e., OXCs and fibers) until the designed network fulfills the RTC requirement. The design method includes the following two problems. We handle them repeatedly by using our ADD algorithm (See Fig. 2.2).

1. OXC-deployment problem: Given the expected traffic demand and a WDM physical topology, we determine how many ports of each OXC are needed to design a robust network. To achieve this, we first find the node-pair that limits the traffic volume



ADD algorithm iterates followings until the RTC is fulfilled

- Upgrade of OXCs
- Deployment of fibers by EMIRA

Figure 2.2: Physical WDM network design with ADD algorithm

accommodated by the network. We then determine the OXC port count needed on a node so that the traffic volume to be accommodated is maximized.

2. Fiber-deployment problem: Given the expected traffic demand and the WDM physical topology including the new OXCs in the OXC-deployment problem, we determine where and how many fibers to lease. To achieve this, we propose EMIRA algorithm. Its objective is to deploy optical fibers to maximize the volume of accommodated traffic. Note that our EMIRA adds a fiber only when there are sufficient OXC ports.

2.3 Design Algorithm of Physical Topology Robust against Traffic Changes

The traditional ADD algorithm was proposed to resolve the warehouse deployment problem [20]. In the traditional algorithm, the iteration of adding a warehouse is continued until the addition offers cost savings less than a given value. In our ADD algorithm, we find two main differences from the traditional one. The first is the condition to end the iteration. Iterations are stopped when the designed network can individually accommodate all the K traffic patterns. The other is a pointer to add resources during the iteration. We select the node to be upgraded on the basis that the maximum flow value of the bottleneck node-pair is increased to the highest possible level. The maximum flow value of a source-destination pair means an upper bound for the total amount of available bandwidth (the number of lightpaths in our case) that the node-pair will be able to accommodate by utilizing the remaining resources. The bottleneck node-pair is defined as the one whose ratio of the maximum flow value to the volume of traffic demand is lowest (See Section 2.3.1).

Our solution approach to the network design problem is as follows.

INPUT

$\boldsymbol{G}_{(x-1)}$: WDM physical topology designed during the previous period (the $(x-1)$ th period).

$\alpha^{(x)}$: Expected traffic growth rate from the previous design period.

$\boldsymbol{M}^{(x-1)}$: A matrix each element of which represents expected volume of traffic demand in the previous period, $\mu_{ij}^{(x-1)}$.

$\boldsymbol{\Sigma}^{(x)}$: A matrix each element of which represents a standard deviation, $\sigma_{ij}^{(x)}$. It determines how the traffic demands between nodes i and j change during period x. A different standard deviation for every node-pair can be inputted.

K : Number of traffic matrices used to design a robust WDM network.

p : Number of OXC ports initially placed on each node.

δ : Number of increased ports when a new OXC is upgraded.

OUTPUT

WDM physical topology that fulfills the RTC requirement during this period.

DESIGN METHOD

Step (1): Calculate K traffic matrices as follows.

Step (1-a): Generate a traffic matrix, $T_0 = \{\mu_{ij}^{(x)}\}$, based on a predicted traffic demand, where $\mu_{ij}^{(x)} = \alpha^{(x)} \times \mu_{ij}^{(x-1)}$.

Step (1-b): Based on T_0, generate $(K-1)$ traffic matrices $(T_1, \ldots, T_{K-1})$. Each element t_{ij}^k $(1 \leq k \leq K-1)$ follows a normal distribution $N(\mu_{ij}^{(x)}, (\sigma_{ij}^{(x)})^2)$.

Step (2): Install a $p \times p$ OXC for each wavelength at each node. We refer to the installed OXC as an upgradable OXC. They are added to a topology designed in $(x-1)$th period $\boldsymbol{G_{x-1}}$.

Step (3): Apply ADD algorithm. Namely, repeat following steps until RTC is satisfied.

Step (3-a): Increase the number of ports of upgradable OXCs by δ at node n that is a bottleneck of the traffic volume accommodated by the network. In Section 2.3.1, we describe how to select node n in detail.

Step (3-b): Lease fibers. Input K traffic patterns from T_0 through T_{K-1} and try to accommodate traffic demand that have not been accommodated in the previous iteration yet by using EMIRA (see Section 2.3.2). Set b_k to the number of lightpaths that cannot be accommodated when the traffic pattern is T_k.

Step (3-c): If the total number of blocked lightpaths ($\sum_{k=0}^{K-1} b_k$) is greater than 0, go back to Step (3-a) and try to upgrade OXCs. Otherwise finish the designing the network.

In Step (1), we roughly predict traffic pattern $\boldsymbol{T}_0$ assuming that the traffic increases at a regular rate [1]. Then we generate $(K-1)$ traffic patterns ($T_1, \ldots, T_{K-1}$). In Step (2), we install a $p \times p$ non-blocking OXC for each wavelength on nodes. On the node that is short of ports, increase the number of ports using the following steps. In Step (3), we apply our ADD algorithm. A WDM network can be designed by repeating Steps (3-a) through (3-c) until all the K traffic patterns are individually accommodated. In Step (3-a), we upgrade the OXCs on the target node. Since one OXC is prepared to each wavelength (see Fig. 2.1), we simultaneously upgrade all the OXCs on the node. As a result, we can keep the numbers of ports of the OXCs on the node identical regardless of wavelength. We regard the designed WDM network that accommodates all the traffic patterns generated in Step (1) as a robust one.

2.3.1 Design Algorithm for the OXC-Deployment Problem

The objective of the OXC-deployment problem is to determine that how many ports of each OXC are needed to design a robust WDM network. We increase the number of ports at WDM nodes so that the volume of traffic to be accommodated in the future can be maximized. To achieve this, we focus on the maximum flow value of each source-destination node-pair. Let $F_{ij}^{(n)}$ denote the maximum flow value of node-pair (i, j) when it is assumed that OXCs on node n are upgraded. Traffic demand to a node-pair, of which the maximum flow value is limited, tends to be blocked because of the lack of the resources. On the other hand, if the volume of the traffic demand is much smaller than the maximum flow value, the demand tends to be accepted. Therefore, we try to increase the maximum flow value of a node-pair in which the ratio of maximum flow value to the expected volume of traffic demand is the lowest. Our scheme for the OXC-deployment problem is described as follows.

Step (1): Select node n that satisfies $\max_{n} \min_{i,j} \dfrac{F_{ij}^{(n)}}{\mu_{ij}^{(x)}}$.

Step (2): Increase the numbers of OXCs ports on node n by δ.

2.3.2 Design Algorithm for the Fiber-Deployment Problem

We also try to design a robust WDM network based on the maximum flow value in the fiber-deployment problem. To do this, we propose EMIRA (Enhanced Minimum Interference Routing Algorithm), which is based on MIRA [21], summarized in the Appendix. Since a fixed physical topology is used in MIRA as input information we cannot apply it to our fiber deployment problem where the physical topology is *output* information. EMIRA uses the layered-graph described in [37] instead of the physical topology. The layered-graph has W layers as shown in Figs. 2.3 and 2.4, where W is the number of multiplexed wavelengths. In the graph of the wth layer, a vertex (i.e., node) corresponds to an OXC for wavelength w and an edge (expressed as $e_{(index\_of\_link),(index\_of\_wavelength)}$ in Fig. 2.4) corresponds to a set of wavelength w's available resources between two OXCs. The link cost of wavelength w on link s is given by Eq. (2.1). If no wavelength w is idle between an OXC-pair, the corresponding link cost is infinity. According to the shortest path routing

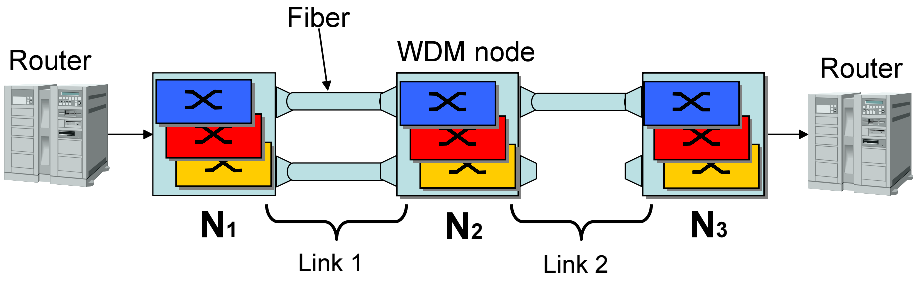


Figure 2.3: Original network of the layered graph

on the layered-graph, we determine where to route lightpaths that are to accommodate the traffic demand. We lease dark fibers on the basis of where lightpaths are to be set up. As a result, we can design the physical topology that can accommodate traffic demand.

The key idea behind EMIRA is to select a route such that sufficient equipment in addition to wavelength resources remains for potential traffic demand in the future. In EMIRA, we assign a link cost expressed by Eq. (2.1) to each link on the layered-graph. It takes into account the remaining resources as well as *critical link*s. Critical links are defined as links with properties that whenever traffic demand is routed over them the maximum flow values of one or more source-destination pairs decrease [21]. EMIRA gives priority to determining a path that has abundant remaining resources by utilizing the amount of remaining resources as the denominator of link cost.

$$Cost_{sw} = \begin{cases} \infty & \text{if } B_{sw} = 0 \text{ and } C_{sw} = 0, \\ 0 & \text{if } A_{sw} = 0,\ B_{sw} \neq 0 \text{ and } C_{sw} = 0, \\ \dfrac{A_{sw}}{B_{sw} \times \frac{A_{sw}}{Q(Q-1)} + C_{sw}} & \text{otherwise,} \end{cases} \qquad (2.1)$$

where

A_{sw}: Number of node-pairs that regard wavelength w on link s as a *critical link*. How to calculate A_{sw} is explained in the Appendix.

B_{sw}: The least number of remaining OXC ports for wavelength w at two nodes connected to link s.

C_{sw}: Number of idle wavelength w in multiple fibers on link s.

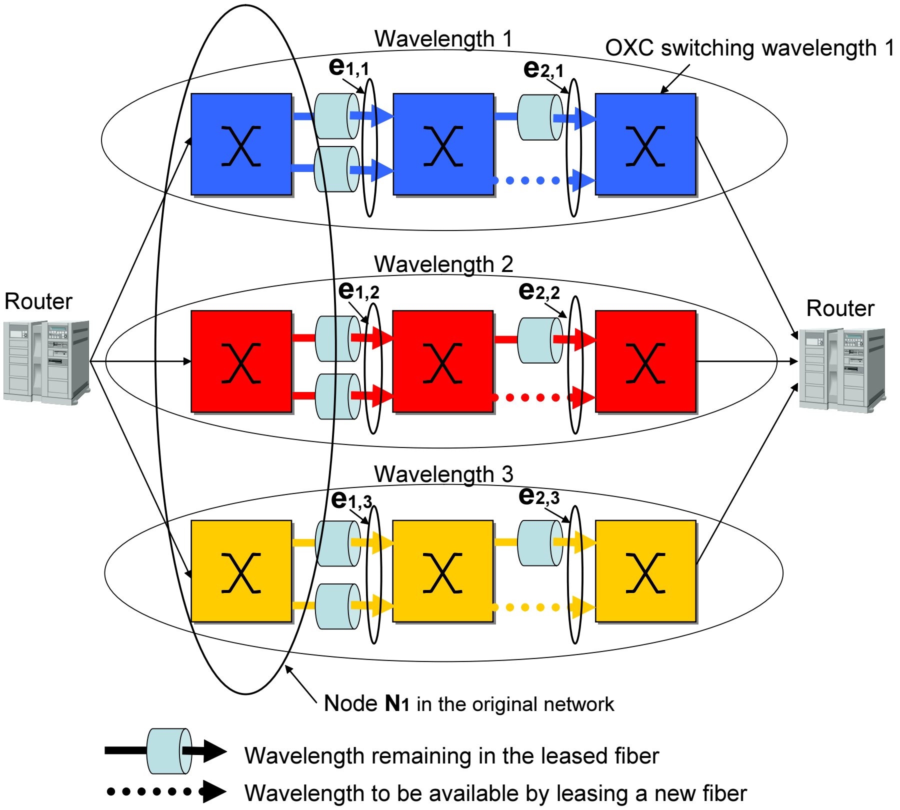

Wavelength 1
OXC switching wavelength 1
e1,1
e2,1
Wavelength 2
e1,2
e2,2
Router
Router
Wavelength 3
e1,3
e2,3
Node N1 in the original network
Wavelength remaining in the leased fiber
Wavelength to be available by leasing a new fiber

Figure 2.4: Example of layered graph: The number of wavelengths is 3

Q: Number of nodes in the physical topology. $Q \times (Q-1)$ is the total number of node-pairs, that is, the upper bound value of A_{sw}.

When $B_{sw} = 0$ and $C_{sw} = 0$, the link cost of wavelength w on link s is infinity because there is no wavelength to set up lightpaths. When $A_{sw} = 0$, $B_{sw} \neq 0$ and $C_{sw} = 0$, the link cost of wavelength w on link s is 0 because wavelengths remain by leasing new fibers and no node-pair regards it as a critical link.

By introducing B_{sw}, we place priority on selecting a route where more OXC ports remain. However, we do not simply use the number of remaining OXC ports as a link cost. Instead, we introduce a weight of B_{sw} that changes according to how congested wavelength w on link s is. This is based on the idea that we should use numerous remaining OXC ports in the congested link while keeping remaining OXC ports for the future traffic demand in links that are not congested. A congested link is defined as one that many node-pairs regard as a critical link. Therefore, we use the ratio of A_{sw} to the upper bound value of A_{sw} as the weight of B_{sw}. C_{sw} assigns a higher priority to selecting wavelengths remaining in leased fibers than to selecting wavelengths that will become available after a new fiber is leased. By doing this, the required number of fibers can be reduced.

The outputs of EMIRA are (1) the route and the wavelength of a lightpath to be set up, (2) the links where we need to lease new dark fibers. The layered-graph in EMIRA consists of wavelengths remaining on leased fibers, and potential wavelengths that will become available when new fibers are leased. Thus, when EMIRA finds the route for a lightpath, we can always set up the lightpath.

EMIRA is described as follows.

INPUT

- Layered-graph that consists of existing OXCs, remaining wavelengths and potential wavelengths that will become available when new fibers are leased.
- Traffic demand from node i to node j.

OUTPUT

- The route of a lightpath and its wavelength between nodes i and j.
- The links where we need to lease dark fibers between nodes i and j.

ALGORITHM

Step (1): Calculate the A_{sw} by following these steps.

Step (1-a): Calculate the maximum flow of each source-destination pair except (i, j) by using the Fold-Fulkerson algorithm [38] and obtain critical links for each source-destination pair.

Step (1-b): Calculate A_{sw} from Eq. (2.2), which is described in Appendix.

Step (2): Calculate B_{sw} and C_{sw} on the layered-graph.

Step (3): Calculate the link cost on each link by applying A_{sw}, B_{sw} and C_{sw} to Eq. (2.1).

Step (4): Select a path using Dijkstra's shortest path algorithm.

Step (5): Set a lightpath on the route obtained in Step (4). If no wavelength is available, lease a new fiber and connect it to the OXCs.

2.4 Numerical Evaluation and Discussions

2.4.1 Simulation Condition

We use the 15-node network model in Fig. 2.5. There are initially no fiber on each link and when we need them, we lease dark-fibers. We assume that the traffic demand is normalized into the wavelength capacity; that is, traffic demand is equivalent to the number of lightpaths that have been requested to be set up. The number of wavelengths multiplexed on a fiber, W, is set to 4. In our proposed algorithm, the number of OXC ports is initially set to 8 ($p = 8$), and increases by 2 ports ($\delta = 2$). We compare the network designed with our scheme with the one designed to minimize the OXC cost, which is designed by the heuristic optimization method [39]. This belongs to the class of "deterministic heuristics". In this class of methods, an initial topology, which accommodates the traffic demand, is designed by adopting a set of heuristic criteria (e.g., MIN-HOP (Minimum Hop routing) and LLR (Least Loaded Routing)). Then, the network is globally optimized by trying to reroute the traffic demand. The heuristic optimization method has proved to be a superior algorithm which obtains sub-optimal results with less computational effort than ILP (Integer Linear Programming). We use MIN-HOP in the heuristic optimization method. We call these two networks as follows.

PT_{ADD} : Network designed with our proposed scheme to be robust against the traffic changes.

PT_{hom} : Network designed with the heuristic optimization method [39] to minimize OXC costs.

When the traffic demand actually occurs, we must determine which route will accommodate it. Since actual traffic demand occurs dynamically, the route that is assumed to accommodate it during the design stage can differ from the route that actually accommodates it. As a routing algorithm, we use MIRA [21] for both PT_{ADD} and PT_{hom} because it can accommodate as much unpredicted traffic as possible.

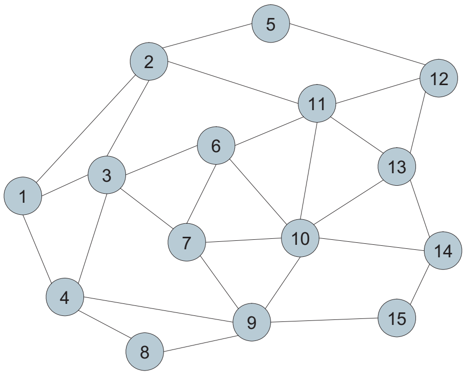


Figure 2.5: Network model (15 nodes, 28 links)

2.4.2 Evaluation Results

We evaluate the performance of PT_{ADD} and PT_{hom} when the traffic change occurs, that is, the value of σ in actual traffic demand changes. We express the predicted traffic as a traffic matrix, $\boldsymbol{T}_0 = \{\mu_{ij}\}$. μ_{ij} is the traffic volume requested by node-pair (i, j). We calculate the cost of $v \times v$ OXC as $\frac{v^2}{64} \times C_8$ (C_8 is the cost of a 8×8 OXC), assuming that the non-blocking OXCs are implemented as crossbar switches. In the PT_{hom}, the OXC cost is calculated based only on the number of ports actually used.

We now discuss the evaluation results when the traffic change occurs. The original heuristic optimization method does not incorporate cases where traffic demand that actually occurs varies, that is, it always regards σ as 0. We modify the original heuristic optimization method to accommodate traffic changes. When K different traffic matrices are inputted, the modified heuristic optimization method first generates a traffic matrix, $\boldsymbol{T}_{\max}$. Each element $t_{ij}^{\max}$ of $\boldsymbol{T}_{\max}$ equals the maximum traffic volume of node-pair (i, j) out of K traffic matrices ($t_{ij}^{max} = \max_k(t_{ij}^k), (k = 0, 1, 2, \ldots, K-1)$). The modified heuristic optimization method, then, can be used to design a network that accommodates $\boldsymbol{T}_{\max}$ with minimum OXC cost. We call the network designed with the modified heuristic optimization method $PT_{\mathrm{modified-hom}}$. Figure 2.6 shows the OXC costs of PT_{ADD} and $PT_{\mathrm{modified-hom}}$ when we use $\mu = 2$ and $\sigma = 1$. The OXC costs represent the relative values to the cost of an 8×8 OXC. The horizontal axis is the number of traffic matrices that are used by each design method. The OXC cost value at the kth index of the horizontal axis shows traffic matrices (from $\boldsymbol{T}_0$ to $\boldsymbol{T}_{k-1}$). t_{ij}^k (i.e., each element of $\boldsymbol{T}_k$) is a value of the random variable that follows a normal distribution, $N(\mu, (\sigma)^2)$. The cost of $PT_{\mathrm{modified-hom}}$

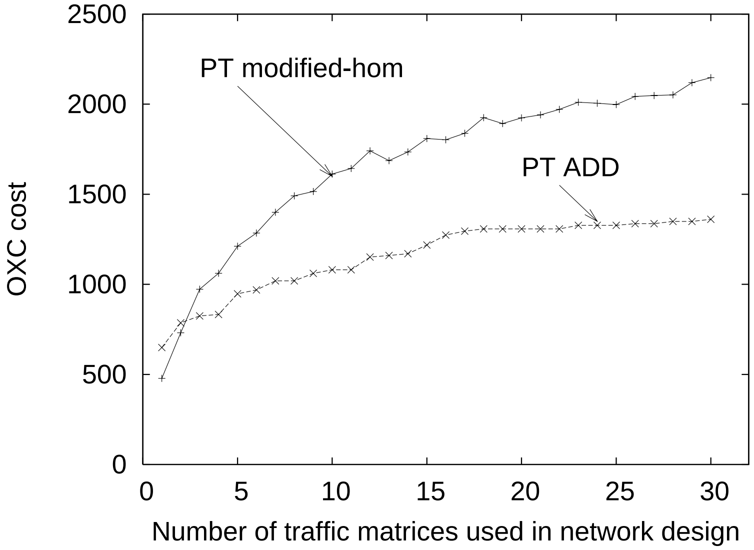


Figure 2.6: OXC costs of PT_{ADD} and $PT_{\mathrm{modified-hom}}$ (traffic $\mu = 2, \sigma = 1$)

does not keep increasing although $\boldsymbol{T}_{\max}$ keeps rising as the number of inputted traffic matrices increases. This is because the estimation-error between the optimal OXC cost and the sub-optimal OXC cost obtained by the modified heuristic optimization method can change as the inputted traffic matrices changes. Note that the cost of $PT_{\mathrm{modified-hom}}$ exceeds that of PT_{ADD} as the number of traffic matrices used in network design gets larger. We can say that it is pointless trying to accommodate the maximum traffic volume of predicted traffic matrices, $\boldsymbol{T}_{\max}$.

To evaluate how cost-effectively our method permits the network equipment to be used, we compare the blocking performance of PT_{ADD} with that of $PT_{\mathrm{modified-hom}}$, both of which are designed with almost the same OXC cost. For this purpose, we selected PT_{ADD} designed with $K = 14$, $\mu = 2$, and $\sigma = 1$ and $PT_{\mathrm{modified-hom}}$ designed with $K = 5$, $\mu = 2$, and $\sigma = 1$. The former costs 1169 and the latter 1211. These costs represents the OXC cost. The numbers of fibers needed by PT_{ADD} and $PT_{\mathrm{modified-hom}}$ are also almost the same; PT_{ADD} needs 381 fibers and $PT_{\mathrm{modified-hom}}$ does 422 fibers. Figure 2.7 shows the average ratio of blocked lightpaths with a 95% confidence interval in PT_{ADD} and $PT_{\mathrm{modified-hom}}$ when the traffic change of the actual traffic (σ) varies from 1 to 4. The horizontal axis is the value of σ in the actual traffic. When the traffic change is the same as predicted ($\sigma = 1$), PT_{ADD} shows about 0.00002 and does $PT_{\mathrm{modified-hom}}$ about 0.0038. Both networks can accommodate almost all the requested lightpaths when the traffic change is the same as

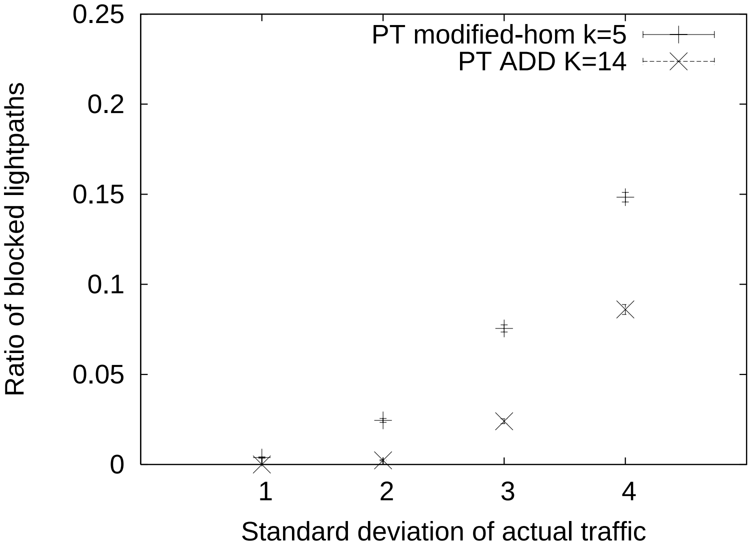


Figure 2.7: Ratios of blocked lightpaths in PT_{ADD} and $PT_{\mathrm{modified-hom}}$

predicted. When the traffic change is larger than predicted, the difference in the ratio of blocked lightpaths between PT_{ADD} and $PT_{\mathrm{modified-hom}}$ gets larger (0.022, 0.051 and 0.064 when σ is 2, 3 and 4, respectively).

We finally compare our design method with the over-provisioning approach. Over-provisioning is a simple way of designing a network, which can accommodate more traffic demand than that predicted. Now let us assume a situation where the the traffic change (σ) is predicted as 1 in designing the network. Here, our method can be used to design a network with traffic matrices that follow $N(2, 1^2)$ while the heuristic optimization method for over-provisioning can be used to design a network that can accommodate more traffic volume than 2 in each node-pair. Figure 2.8 shows the ratio of blocked lightpaths with a 95% confidence interval in PT_{ADD} with $K = 12$, $\mu = 2$, and $\sigma = 1$ and PT_{hom} with $K = 1$, $\mu = 3$, and $\sigma = 0$. In this case, the cost of $PT_{\mathrm{ADD}}$is almost same as that of PT_{hom} with over-provisioning, which tries to accommodate 1.5 times as much traffic demand as predicted. The former costs 1151 and the latter 1156. PT_{ADD} needs 365 fibers and PT_{hom} does 462 fibers. We assume that the traffic change of the actual traffic would vary from 1 to 4 in Fig. 2.8. The horizontal axis shows the value of the traffic change. PT_{ADD} has a lower ratio of blocked lightpaths than PT_{hom} over all σs. Similarly the difference in the ratio of blocked lightpaths between PT_{ADD} and PT_{hom} gets larger (0.0016, 0.018, 0.049 and 0.063 when σ is 1, 2, 3 and 4, respectively). Our method can design the cost-effective

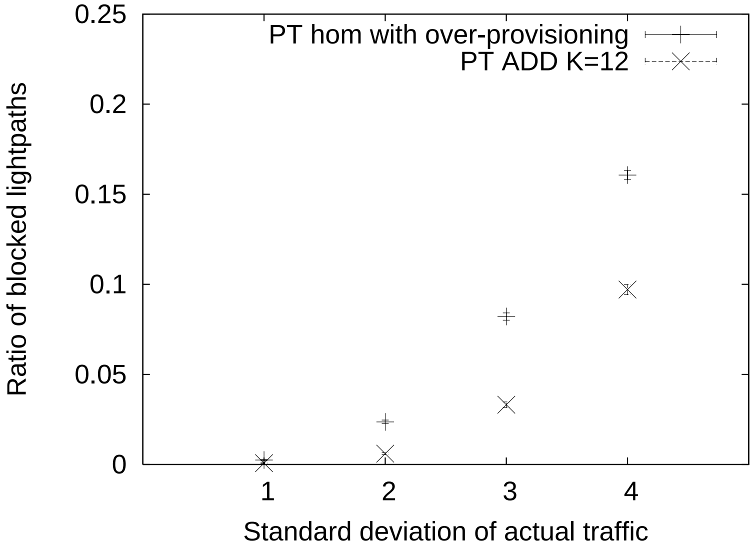


Figure 2.8: Ratios of blocked lightpaths in PT_{ADD} and PT_{hom} with over-provisioning

network by properly adjusting the number of OXC ports.

2.5 Conclusion

In this chapter, we have proposed a novel design method of WDM network that is robust against traffic changes. Through the simulation, we evaluated how cost-effectively we use the network equipment by comparing the network that our proposed method designs with those that the conventional methods design, both of which need almost the same OXC cost. As a result, we have shown the network that our proposed method designs achieves lower ratio of blocked lightpaths than the one obtained by the over-provisioning approach does. We conclude that our proposed method designs a robust WDM network in the cost-effective way.

Appendix: MIRA (Minimum Interference Routing Algorithm)

Here we briefly explain MIRA [21]. MIRA dynamically determines the routes needed to meet traffic demand one-by-one as they occur, without a priori knowledge of future traffic demand. The key idea behind MIRA is to select a path that minimizes interference with potential future traffic demands between other source-destination pairs. Figure 2.9

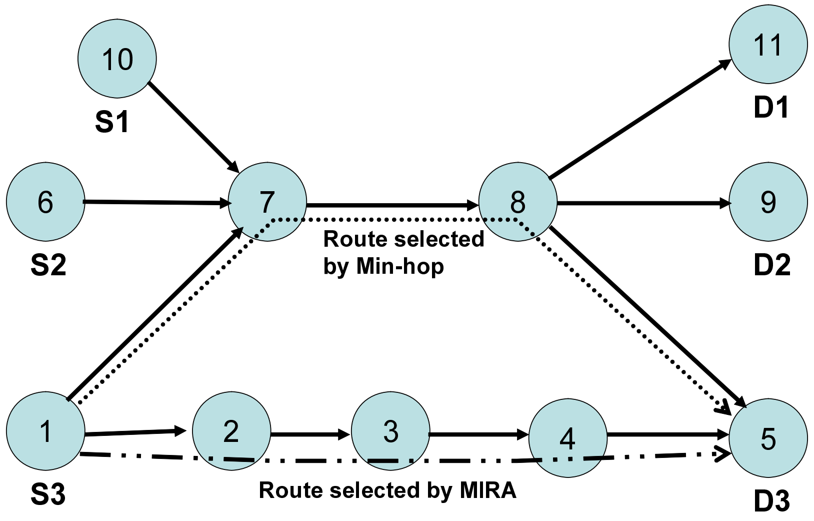


Figure 2.9: Routes selected by MIN-HOP and MIRA

illustrates how MIRA selects a route. There are three source-destination pairs, (S1,D1), (S2,D2), and (S3,D3) in the network. When (S3,D3) requires one lightpath, the existing MIN-HOP (minimum hop-count) routing algorithm selects a route $1 \rightarrow 7 \rightarrow 8 \rightarrow 5$. MIN-HOP is a routing algorithm that selects a route with minimum-hop counts. However, the link from node 7 to node 8 is also used for both (S1,D1) and (S2,D2). Setting up a lightpath on route $1 \rightarrow 7 \rightarrow 8 \rightarrow 5$ affects the potential use for (S1,D1), (S2,D2). MIRA avoids passing on a route that has the potential for a lot of traffic. It selects route $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5$, which minimizes the interruption to other node-pairs.

To move on from the concept of minimum interference links to a viable routing algorithm that uses maximum flow and shortest path algorithms, MIRA incorporates the notion of “critical links”. The “critical links” are defined as links with the property that whenever traffic demand is routed over them the maximum flow values of one or more source-destination pairs decrease. MIRA counts the number of node-pairs for each link, which regard the link as a “critical link”, and sets it to the link cost to cope with future traffic demand. MIRA assigns the link cost, $Cost_{sw}$, to wavelength w on link s and determines the route using Dijkstra’s shortest path algorithm. $Cost_{sw}$ is represented by A_{sw}, which is the number of source-destination pairs whose critical links include wavelength w

on link s. That is,

$$Cost_{sw} \quad = \quad A_{sw} = \sum_{i,j} x_{ij}^{sw} a_{ij}^{sw}, \tag{2.2}$$

where

x_{ij}^{sw}: If the maximum flow from node i to node j includes wavelength w on link s, then $x_{ij}^{sw} = 1$. Otherwise $x_{ij}^{sw} = 0$.

a_{ij}^{sw}: If wavelength w on link s is available after maximum flow has been carried from node i to node j, then $a_{ij}^{sw} = \ 0$. Otherwise $a_{ij}^{sw} = 1$.

Chapter 3

Design of Logical Topology with Effective Usage of Wavebands

In this chapter, we focus on a logical topology design problem in large-scale wavelength-routed networks where hundreds or thousands of wavelengths are multiplexed. In conventional researches, it is assumed that a constant number of wavelengths be available on each fiber. But it is not necessary to utilize all wavelengths on each fiber in building a logical topology. Instead, several wavebands, which include a set of wavelengths amplified by an optical amplifier, may be considered for introduction while deploying additional wavebands and their corresponding optical amplifiers when additional wavelengths are actually required. In this case, the number of wavelengths available on the respective fibers depends on the number of optical fiber amplifiers deployed on each fiber. Therefore we propose a heuristic algorithm for the design of a logical topology with as few optical fiber amplifiers as possible. Our results indicate that our algorithm reduces the number of optical fiber amplifiers with a slight increase of average packet delays.

3.1 Design of Logical Topology in Large-Scale Wavelength-Routed Network

WDM technology, in which multiplexed wavelength channels are carried on a single fiber, is expected to cope with the explosion of the traffic demand for the current and future

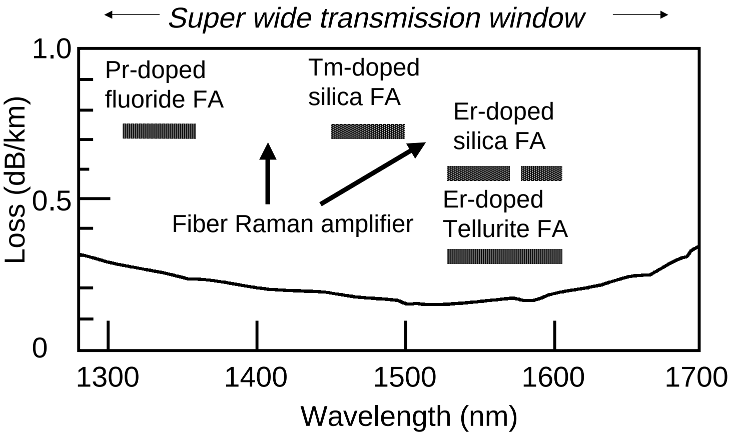


Figure 3.1: Loss spectrum of typical low-loss optical fiber

Internet. Since the majority of Internet traffic is IP-packets, much recent research has been devoted to an IP-over-WDM network, where IP packets are directly carried over the WDM network. Among several architectures for IP-over-WDM networks, one promising approach is to create a logical topology that is made up of lightpaths as an overlay upon the physical WDM network, each of which carries IP traffic between two edge nodes [40]. Such a lightpath is a wavelength-channel that does not require any electronic processing at intermediate nodes. This reduces the load of packet processing at the intermediate nodes.

Having more wavelengths multiplexed on each fiber allows the network to accommodate more lightpaths. Thus, the number of wavelengths available on a single fiber is an important parameter in the design of the logical topology. In the near future, multiplexing of 1,000 wavelengths on a fiber is possible by using a spectral range of 1290–1690 nm [41, 42]. Figure 3.1 shows the amplifiers required across the spectral range to realize 1,000 wavelengths [41]. As the figure shows, deploying additional optical fiber amplifiers makes a number of low loss regions (e.g., 1530–1610 nm) available. We require several kinds of optical fiber amplifiers to utilize more wavelengths on top of those considered in previous work [43].

A lot of works have dealt with methods for the design of the logical topology [8, 7]. Most of these works have been based on the assumption that a constant number of wavelengths is available on each fiber, and then minimize the congestion of the network [8]. In the design of a cost-effective network, however, it is preferable to provide only the wavelengths that are actually needed on the fibers. Utilizing a constant number of wavelengths requires installing all kinds of amplifiers for the entire spectral range. On the other hand, we can minimize the number of optical amplifiers by deploying them only on fibers that are short of wavelengths. For this purpose, we need a new way of designing the logical topology such that it minimizes the number of optical amplifiers while meeting the demands imposed by traffic. This is the main subject of this chapter.

Some approaches aim to minimize the number of wavelengths required within a Wavelength Routed Optical Network (WRON) for the given traffic demands [10]. In a WRON, each lightpath is directly set up from the source to the destination. It seems that minimizing the number of wavelengths leads to minimizing the number of wavebands (optical amplifiers). However, there exist wavelengths that remain unused on fibers because they do not satisfy the *wavelength continuity constraint*. The wavelength continuity constraint means that a lightpath must consist of the same wavelength across all fibers that it traverses. Thus, we need to deploy additional optical amplifiers even if there exist available wavelengths on the fibers. In IP-over-WDM networks, on the other hand, we do not need to directly set up lightpath from the source to the destination. Instead, we split the lightpath into two parts; a lightpath (denote L_A) from the source node to an intermediate node and a lightpath (denote L_B) from the intermediate node to the destination node. In this case, we can assign different wavelengths to L_A and L_B, which leads to relaxation of the wavelength continuity constraint. As a result, we expect to decrease the number of optical amplifiers. However, the processing capacity of the intermediate nodes should also be of concern because cutting a lightpath at an intermediate node increases the packet processing load of it.

In this chapter, we propose a new algorithm called MALDA (Minimum number of fiber Amplifiers Logical topology Design Algorithm) for IP-over-WDM networks. This algorithm is in contrast to earlier approaches in that it minimizes the deployment of optical fiber amplifiers on the fiber under the constraint that the load of all the nodes should be kept under their processing capacity. As far as we know, this is the first work that tries to minimize the number of fiber amplifiers.

The Chapter is organized as follows. In Section 3.2, we extend the conventional method for designing the logical topology to indirectly set lightpaths based on the actual traffic demand. We next propose a logical topology design method that has, as its objective function, the minimization of the number of fiber amplifiers. This is done in Section 3.3. Section 3.4 contains a comparative evaluation of our proposed algorithms and the conventional algorithm. We finally conclude this chapter in Section 3.5.

3.2 Design Algorithm for Logical Topology Based on Requested Traffic Volume

In this section, we extend MLDA (Minimum-delay Logical topology Design Algorithm), a conventional method for designing the logical topology proposed in [8]. We do this extension in order to propose a new logical topology design algorithm (1) that ensures the accommodation of the traffic demand and (2) that incorporates IP's route selection mechanism, i.e., the packet traverses on the shortest path. We call our new algorithm e-MLDA (extended MLDA).

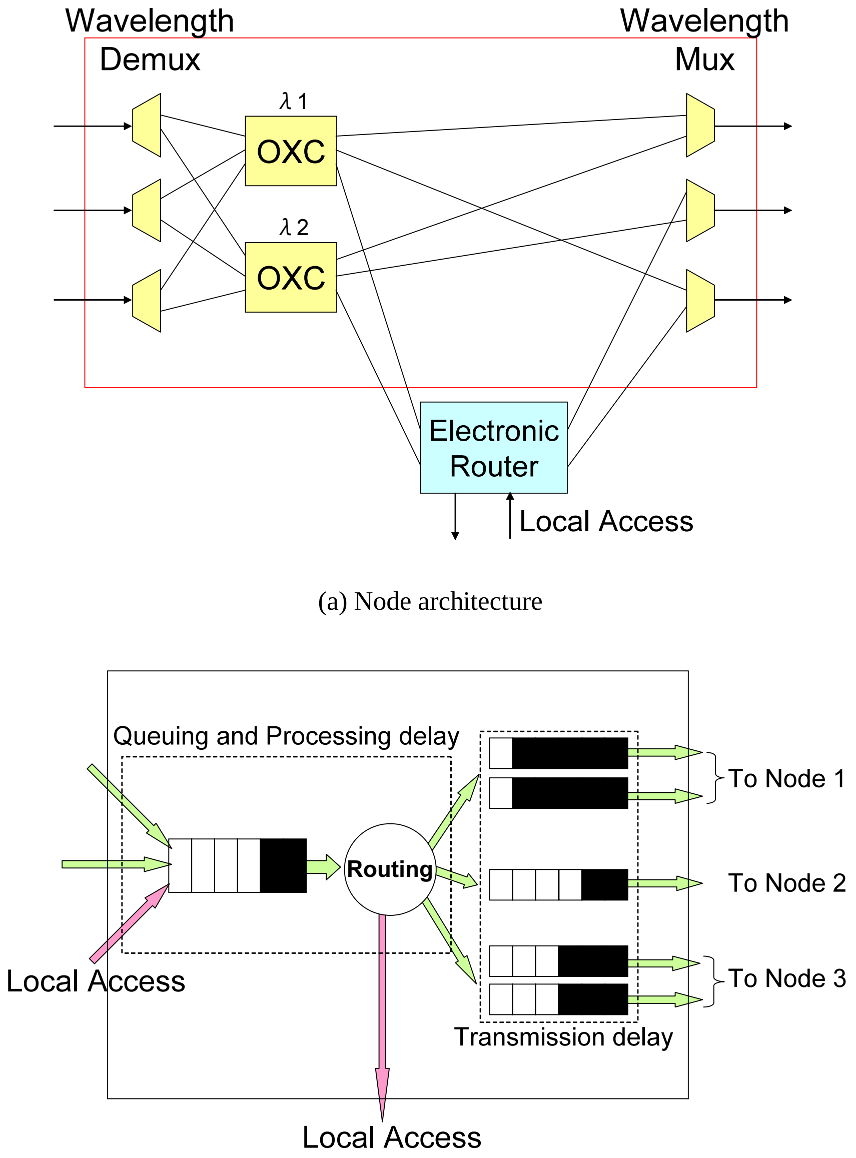
The design problem of a logical topology in WDM networks is traditionally called the RWA (Routing and Wavelength Assignment) problem. RWA solves the following problem. Given (1) a physical network, (2) a traffic matrix that expresses the static traffic demand in the physical network, and (3) constraints (e.g., the number of wavelengths multiplexed on a fiber), we must determine (1) the route and (2) the wavelength to be assigned to the lightpath of each traffic demand so that an objective function (e.g., throughput or the number of wavelengths utilized) is optimized. Note that the above mentioned traffic matrix is determined by long-term measurements. When the traffic matrix is different from the real one, we can cope with it by performing a reconfiguration of the logical topology with minimal disruption [44, 45].

Since MLDA heuristically sets up lightpaths without considering the traffic volume that a lightpath can accommodate, the logical topology designed by MLDA may not accommodate the traffic demand. On the other hand, we want to accommodate the given traffic demand, the unit of which has a particular value in, e.g., Gbps, on the network with a lot of wavelengths multiplexed. Then, our e-MLDA sets up enough lightpaths to accommodate the volume of the required traffic. For each lightpath, MLDA sets up a "one-hop" lightpath. Here, the term "one-hop" lightpath means that a lightpath is directly set up from the source node to the destination node without terminating on intermediate nodes. Setting up only one-hop lightpaths is not desirable because that needs more wavelengths to overcome the wavelength continuity constraint. Thus, our e-MLDA approach sets up "multi-hop" ligthpaths. The term "multi-hop" lightpath means that the lightpath is split at some intermediate nodes. At those intermediate nodes, the traffic on the lightpath is processed by an IP router and it can be assigned to the lightpaths that use another wavelength.

We need these extensions to deal with our main objective of minimizing the number of optical fiber amplifiers. This objective is covered in the next section. Note that in this section we extend the conventional approach assuming that the number of wavelengths on the fiber is fixed. In the next section, we will also cover the case where the number of wavelengths is a design variable that depends on some number of costly optical amplifiers.

Before describing our algorithm, we depict the node-architecture model in Fig. 3.2. Every node is equipped with an optical switch and an electronic router. The optical switch consists of three main blocks; input section, non-blocking switch, and output section. In the input section, the optical signals are demultiplexed into W fixed wavelengths, $\lambda_1, \cdots,$ λ_W. Each wavelength is then switched into an appropriate output port, without wavelength conversion, by a non-blocking switch. Finally, the wavelengths are again multiplexed on the fibers, that go to the respective next nodes. Note that a lightpath is configured by the non-blocking switches along the paths, so that the traffic on a particular wavelength is forwarded from the input port to the required output port without any electronic processing. At the terminal node of a lightpath, IP packets in the lightpath are converted to electronic signals and forwarded to the electronic router. The electronic router performs packet forwarding, in the same way as in a conventional router. If the packet requires further forwarding to other nodes, it is put on the appropriate lightpath. IP packets, whether they come through the optical switch or from local access, are first buffered for processing. The packets are then processed on a FIFO (first-in first-out) basis. Packets that are to be forwarded within the network are queued in the appropriate output port buffer.

Now we show our e-MLDA algorithm. We introduce the following notations to represent the physical network.



(a) Node architecture

(b) Model of electronic router

Figure 3.2: Node architecture model

N: Number of nodes in the WDM network.

P_{ij}: Matrix that represents the connectivity of the physical network. If there is a fiber that connects node i and node j, then entry $P_{ij} = 1$, otherwise $P_{ij} = 0$.

Q: Traffic distribution matrix. The value of an element (i, j) represents the traffic demand between nodes i and j.

C: Bandwidth of each wavelength.

W: Number of wavelengths multiplexed on a single fiber.

Given these parameters, e-MLDA designs the logical topology by setting up multi-hop lightpaths that are sufficient to accommodate the requested traffic volume between nodes. The reason we set up multi-hop lightpaths is to avoid the lack of wavelengths. If we set up one-hop lightpaths from source node to destination, we can set up fewer lightpaths because of the wavelength continuity constraint. Furthermore, we can decrease the number of wavebands by assigning traffic to the lightpaths that use the wavelengths in the same waveband at an intermediate node.

Our e-MLDA sets lightpaths on the shortest routes in terms of the propagation delay between nodes, which is the same route selection as MLDA does. In addition, we make the number of the intermediate nodes (i.e., hop count over the logical topology) for the same node-pair identical when more than one lightpaths are set up between a node-pair. As a result, we expect that IP packets, which flow on the shortest-path in terms of the propagation delay, can flow on any of the lightpaths. If we do not make their hop count identical, IP packets will flow only on the lightpaths whose hop counts are minimum.

The wavelengths chosen for the lightpaths is based on a First-Fit policy, that is, e-MLDA selects the wavelength with the lowest index of λ among those wavelengths that are not yet assigned to lightpaths. First-Fit is preferable in our case because it gives priority to selecting the wavelength available by already installed fiber amplifiers.

We use the following notations to explain our algorithm.

s, d: Source/destination nodes of a lightpath to be set up. Our algorithm recursively tries to set up multi-hop lightpaths; if a direct lightpath cannot be set up between node i

and j, $\{s, d\}$ is first set to $\{i, x\}$, then to $\{x, j\}$. The x is an intermediate node on the shortest path from node i to node j.

q_{ij}: Traffic volume that is requested for node-pair (i, j).

B_{ij}: Node connected to node j along the shortest path from node i to node j.

T_{ij}: Total available bandwidth in the existing lightpaths between nodes i and j.

Using these notations, we now explain our e-MLDA algorithm. This is followed by some additional comments on the algorithm.

Step 1 Among node-pairs that are directly connected by the fiber, select a pair of nodes (i', j') such that element $q_{i'j'}$ of the traffic-distribution matrix Q is the largest. If $q_{i'j'}$ is larger than 0, go to Step 2 and try to set up lightpaths for the connected node-pair $i'j'$. Otherwise, select (i', j') again such that $q_{i'j'}$ is the largest among node-pairs that are not directly connected. If $q_{i'j'} = 0$, then the lightpaths are selected between all the nodes. Thus, we terminate our algorithm in finite steps. Otherwise, go to Step 2.

Step 2 Initialize the variables as $s \leftarrow i'$, $d \leftarrow j'$. Then, go to Step 3 and try to set lightpaths of adequate capacity between nodes s and d.

Step 3 If $s = j'$, the lightpaths have enough capacity to accommodate the traffic from node i' to node j'. Then, set $q_{i'j'} \leftarrow 0$, and go back to Step 1. Otherwise, go to Step 4.

Step 4 Try to accommodate $q_{i'j'}$ on the existing lightpaths between nodes s and d according to the following two conditions.

1. If $T_{sd} \geq q_{i'j'}$, then we can accommodate $q_{i'j'}$ by using the existing lightpaths between nodes s and d. That is, set $s \leftarrow d$, $d \leftarrow j'$ and go back to Step 3.
2. If $T_{sd} < q_{i'j'}$, on the other hand, it is not possible to accommodate $q_{i'j'}$ on the existing lightpaths. Thus, go to Step 5 and try to set new lightpaths between nodes s and d.

Step 5 Try to set $\lfloor (q_{i'j'} - T_{sd})/C \rfloor$ lightpaths between nodes s and d. If it is possible to set the lightpaths along the shortest route, go to Step 5.1. Otherwise, go to Step 5.2.

Step 5.1 After setting up the lightpaths between nodes s and d, we split the lightpaths that originate at node s and pass through node d at node d. Then, we set $s \leftarrow d$, $d \leftarrow j'$ and go back to Step 3.

Step 5.2 If nodes s and d are directly connected via fiber, we are unable to set up lightpaths between nodes s and d because we have already checked that there exists no available wavelength between nodes s and d. In this case, it is not possible to accommodate the requested traffic between nodes i' and j', and we terminate our algorithm. If nodes s and d are not directly connected, on the other hand, we try to accommodate the traffic by creating lightpaths between node s and inter-node B_{sd}. Set $d \leftarrow B_{sd}$ and go back to Step 4.

Comments on e-MLDA

In Step 1, e-MLDA gives priority to setting up lightpaths between node-pairs that are directly connected by fiber. This operation is necessary to ensure the reachability between nodes. The e-MLDA approach selects a node-pair (i', j') in descending order of traffic volume, which is the same way of selecting the node-pair as MLDA does. Though there are other ways of selecting the node-pairs to be accommodated (e.g., longest first, random), the effect of the order of node-pairs to be accommodated on the performance is small (the difference among the various ways is bellow 10% [39]). Step 4 checks whether or not existing lightpaths are capable of accommodating the traffic $q_{i'j'}$. If the available bandwidth T_{sd} is insufficient to transport the IP traffic, new lightpaths are set up in Step 5. Since T_{sd} is already available by existing lightpaths, the number of lightpaths required to accommodate the requested traffic volume is $\lfloor (q_{i'j'} - T_{sd})/C \rfloor$.

Step 5.1 deals with the case where we are able to set up enough lightpaths to accommodate the requested traffic. However, in an IP-over-WDM network, we must consider the property of IP, that is, the shortest path is utilized by IP traffic, even if multi-hop lightpaths with larger hop count are available. To avoid the situation where multi-hop lightpaths with different hop counts are set up between any node-pair, any lightpaths originating at node s and passing through node d are split at node d. In Step 5.2, if we are unable to set up the required lightpaths because there are too few available wavelengths, we set $d \leftarrow B_{sd}$ and go back to Step 4 in order to accommodate q_{ij} between nodes s and B_{sd}. Note that, after $q_{i'j'}$ has been accommodated between s and B_{sd}, Step 5.1 sets s to B_{sd} and d to j'. We then try to set up a lightpath between nodes B_{sd} and j'.

We now evaluate the complexity of e-MLDA. For $N(N-1)$ node-pairs, e-MLDA tries to set up multi-hop lightpaths. In order to set up multi-hop lightpaths for a node-pair, e-MLDA searches the available wavelengths among W wavelengths for $\sum_{i=0}^{H-1}(H-i)$ times at most (H is a hop count of a route between a node-pair). This is because e-MLDA tries to set up lightpaths that are one-hop shorter than those that e-MLDA tried to set up before. As a result, e-MLDA tries to set up lightpaths with $H, H-1, \ldots, 1$ hop counts in turn until e-MLDA finds enough wavelengths. The total complexity of e-MLDA is $O(N^2H^2W)$.

3.3 Design Algorithm for Logical Topology with Effective Usage of Wavebands

As we mentioned, we need to install only the different types of fiber amplifiers on a fiber, which would otherwise not fulfill the required bandwidth. In this way, the most cost-effective logical topology can be achieved. In this section, we propose a new method for the design of logical topologies that minimizes the number of optical amplifiers deployed. We call this algorithm MALDA (Minimum number of fiber Amplifiers Logical topology Design Algorithm).

In our MALDA, W_1 ($< W$) wavelengths are initially set for carrying traffic by each fiber. When there is no available wavelength on a certain fiber during the subsequent design of the logical topology, W_i wavelengths are added by introducing an additional fiber amplifier type i ($2 \leq i \leq N_{max}$). Here, we assume that N_{max} kinds of fiber amplifiers can be deployed on the fiber. Note that we select the wavelengths in the waveband that is available with EDFA ($C + L$ band) as W_1. W_i and N_{max} are determined by the technological constraints as Fig. 3.1 shows. If the maximum number of wavelengths that can be multiplexed on a fiber is W, we obtain the following relationship for fiber f,

$$\sum_{i=1}^{N_f} W_i \leq W, \tag{3.1}$$

where N_f ($1 \leq N_f \leq N_{max}$) is the number of fiber amplifier types deployed on fiber f. Adding a new fiber amplifiers means to install an additional type of fiber amplifiers to increase the number of wavelengths of the fiber by an additional waveband. The objective function of MALDA is,

$$minimize \qquad \sum_{f \in F} N_f. \tag{3.2}$$

In practice, various components (e.g., OEO converters) are also required in addition to the optical amplifier to overcome physical impairments (e.g., noise and dispersion) [46]. In this chapter, however, we simply try to minimize the number of wavebands that are actually used because the number of these components required depends on the number of wavebands actually used.

In MALDA, fiber amplifiers are added to fiber when too few wavelengths are available to set up new lightpaths. The algorithm terminates when all the traffic demand has been accommodated and the load on all the IP routers become under their processing capacity. In addition, we expect that the smallest possible number of fiber amplifiers will be deployed in the WDM network. MALDA is similar to e-MLDA described in Section 3.2. The point of difference between them is that MALDA only deploys an additional fiber amplifier when there are too few wavelengths to accommodate the traffic. For this purpose, we need to modify Step 5.2 of e-MLDA. Once a fiber amplifier has been added to a fiber, we are able to connect a lightpath that uses the newly available wavelengths. Whether or not a new amplifier should be added is checked in the new step, Step 6. The following two steps are one of the two differences between e-MLDA and MALDA. Another difference is described in the next subsection.

Step 5.2 If nodes s and d are directly connected via a fiber, we may be able to set up lightpaths between nodes s and d. In this case, we try to accommodate $q_{i'j'}$ by deploying a new fiber amplifier on the fiber, so we go to Step 6. If nodes s and d are not directly connected, on the other hand, then we set $d \leftarrow B_{sd}$ and go back to Step 4.

Step 6 Check the number of fiber amplifiers currently deployed on the fiber between nodes s and d. If N_{max} amplifiers have already been used, it is not possible to accommodate the required traffic and we terminate our algorithm. Otherwise, we add an additional fiber amplifier to increase the number of available wavelengths on the fiber, and connect the existing lightpaths. Note that the wavelengths used by the lightpaths from node s to node d are released and newly available wavelengths provided by the added amplifier are reassigned to those lightpaths. We then set $d \leftarrow j'$ and go back to Step 4 in order to check whether or not we are able to set up new lightpaths between nodes s and d by adding a fiber amplifier.

The reassignment of wavelengths to the lightpaths from node s to node d supposes the situation that newly available wavelengths are likely to be available only on the deployed fiber. Thus those wavelength may not be utilized by the other lightpaths that pass through more than one fiber. So those wavelengths should be used by the lightpath that passes through only one fiber.

After setting up all the lightpaths with the above steps, we next consider adding further optical fiber amplifiers to decrease the traffic load on over-burdened IP routers. This is necessary since the above steps does not ensure that the load on all IP routers are below the processing capacity. By connecting lightpaths until the load on the IP router falls below the maximum amount of traffic the IP router can process, we accommodate more traffic. To explain this, we introduce the following notations.

N_{high}: Set of nodes at which the traffic load on the IP router is beyond the maximum amount of traffic it can process.

$N_{available}$: Set of nodes that have non-utilized waveband(s) on the fibers to which the node is connected.

N_{heavy}: Node that has the heaviest traffic load among the set of nodes, chosen from $N_{high} \cap N_{available}$.

We perform the following steps after setting up the lightpaths enough to accommodate all the traffic demand according to the above steps in MALDA.

Step A: Set $N_p \leftarrow N_{high} \cap N_{available}$. If N_p is an empty set, then go to Step C. Otherwise, go to Step B.

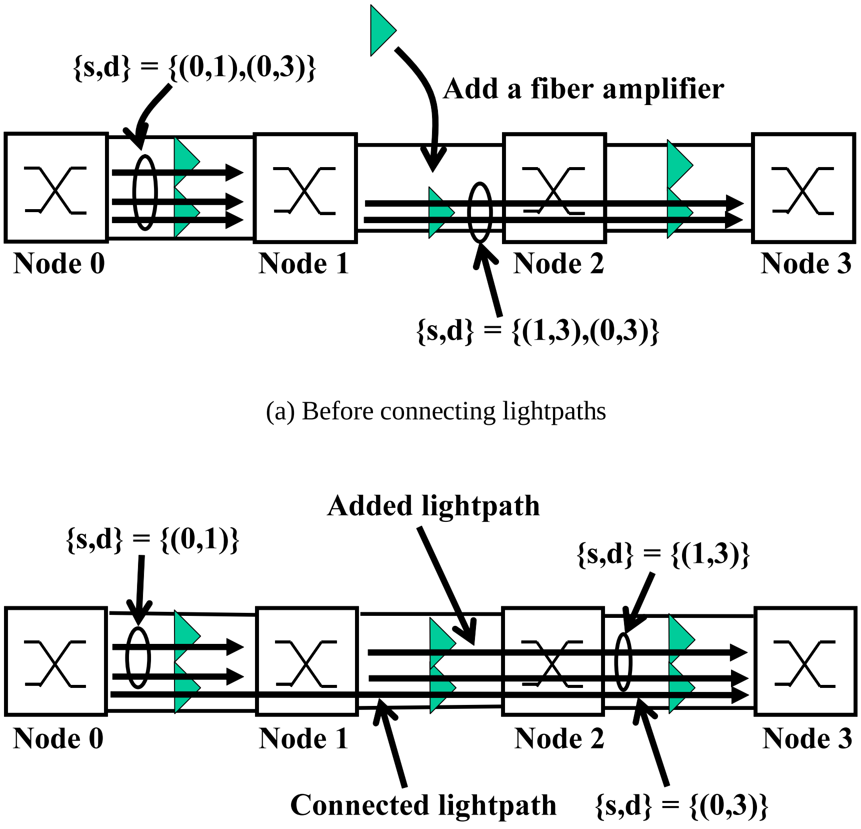
Step B: Randomly choose one fiber from the fibers that are connected to N_{heavy}. Add an optical fiber amplifier to this fiber. Then, try to connect lightpaths through this fiber (see the connecting lightpaths above), and go back to Step A.

Step C: If some nodes have a traffic load that is above the limit of its processing capacity, then the requested traffic cannot be accommodated, and the algorithm is terminated. Otherwise, the new logical topology has successfully accommodated the traffic.

The above three steps decrease the load on overloaded IP routers by connecting lightpaths and bypassing IP routers. If too few wavelengths are available to reduce the load, we deploy additional optical fiber amplifiers. If a node remains in the N_{high} condition even after all possible optical fiber amplifiers have been deployed, we are unable to accommodate the requested traffic.

We explain the algorithm for connecting lightpaths after a new fiber amplifier has been added. The motivation of connecting lightpaths is to prevent IP routers from being overburdened by setting up multi-hop lightpaths. We connect lightpaths at the node selected in descending order of the traffic load on the two nodes, between which a new fiber amplifier is added on the link, since the heaviest loaded node will limit the throughput of the network. We can expect to decrease the load on the IP routers of those nodes.

Let us define x as the node at which we are trying to connect lightpaths. To decrease the traffic load on node x, we try to connect lightpaths in the set of lightpaths that terminate



(a) Before connecting lightpaths

(b) After connecting lightpaths

Figure 3.3: Example of connecting lightpaths

at node x and those in the set of lightpaths that originate at node x, i.e., bypass packet processing at node x. Hereafter, we denote LP_{sx} as the set of lightpaths that originate from node s and terminate at node x, and LP_{xd} as the set of lightpaths that originate from node x and terminate at node d. The operation of the connecting lightpaths is as follows. For any two nodes (say i and j), we try to create LP_{ij} by connecting lightpaths in LP_{ix} and those in LP_{xj}. To do this, we first select the set of node-pairs $\{s,d\}$ that use both LP_{ix} and LP_{xj}. Then, we check whether enough wavelengths are available to connect lightpaths that accommodate the summation of the traffic of the set, i.e., $\sum_{ab\in\{s,d\}} q_{ab}$. If this check is satisfied, there are enough available wavelengths to connect the lightpaths. However, this check is not enough to connect the lightpaths. After we connect the lightpaths, the number of lightpaths in LP_{ix} and LP_{xj} decreases. The traffic overflows by connecting lightpaths. Therefore, we further check whether we are able to accommodate that traffic transmitted via LP_{ix} (or LP_{xj}) that overflows from the connected lightpaths. Only if those two checks are satisfied, we connect the $\lfloor\sum_{ab\in\{s,d\}} q_{ab}/C\rfloor$ lightpaths in LP_{ix} and LP_{xj}.

Figure 3 shows a simple example of the connection of lightpaths. Suppose that the newly added fiber amplifier makes two wavelengths available. Further suppose that $C = 10$ Gbps, and the traffic demands on node pairs $\{0,1\}$, $\{0,3\}$, and $\{1,3\}$ are 15, 7, and 12 Gbps, respectively. The traffic of node pair $\{0,3\}$ is transmitted via a lightpath in LP_{01} and one in LP_{13} since it is not possible to directly set up a lightpath from node 0 to node 3 because of the lack of wavelengths (see Fig. 3a). After the fiber amplifier has been added to the fiber between nodes 1 and 2, we try to connect lightpaths at node 1 and node 2. First, we try to connect lightpaths in LP_{01} and those in LP_{13} at node 1 on which the IP router is more over-burdened. Now we are trying to connect a lightpath that can accommodate the traffic volume for node pair $\{0,3\}$. We first check whether or not it is possible to accommodate traffic that overflows to other lightpaths. If we connect a lightpath on node 1, the number of lightpaths in LP_{01} changes to 2 and that in LP_{13} does to 1. A lightpath in LP_{13} is unable to accommodate the traffic of node pair $\{1,3\}$ (12 Gbps is required, but only 10 Gbps is available). Therefore, we next check whether or not it is possible to accommodate the traffic of node pair $\{1,3\}$ by setting up a new lightpath between node 1 and node 3. Since this is possible in the current case, we set up a new lightpath in LP_{13} and connect a lightpath in LP_{01} and one in LP_{13} as shown in Fig. 3b.

The complexity of MALDA is larger than that of e-MLDA because MALDA adds fiber amplifiers in addition to setting up lightpaths. The complexity of adding fiber amplifiers can be obtained as follows.

A fiber amplifier can be added $L \times B$ times at most. L is the number of links in the network. B is the number of wavebands on a fiber. When MALDA adds a fiber amplifier, it tries to connect W lightpaths at most on the nodes connected by the fiber. So the total complexity of MALDA is larger than that of e-MLDA by $O(LBW)$, that is, the complexity of MALDA is $O(N^2H^2W) + O(LBW)$.

3.4 Numerical Evaluation and Discussions

In the previous section, we proposed a method for the design of the logical topology that has the objective function of minimizing the number of fiber amplifiers. This section is devoted to a comparative evaluation of MLDA, e-MLDA, and MALDA. We introduce the following notations to represent the logical topologies designed by each algorithm.

LT_{MLDA}:	Logical topology designed by MLDA
LT_{e-MLDA}:	Logical topology designed by e-MLDA
LT_{MALDA}:	Logical topology designed by MALDA

3.4.1 Simulation Condition

In this evaluation, we use NTT's 49-node backbone network in Japan (Fig. 3.4) as the network model and two different traffic patterns, P_1 and P_2. P_1 is the publicly available information provided by NTT [47] about the traffic matrix for conventional telephone calls. In traffic pattern P_1, the volume of traffic between large cities and between adjacent cities is large. Traffic pattern P_2 is randomly determined. The value of each element in P_2 is uniformly distributed between 0 Mbps and 1 Mbps. Since the total traffic loads are small (around 3 Gbps in P_1 and 1.2 Gbps in P_2), we introduce a scale-up factor α. We set the actual requested traffic as α times the elements of P_1 and P_2. The bandwidth of each wavelength is set to 10 Gbps, and up to 1,000 wavelengths can be multiplexed on a single

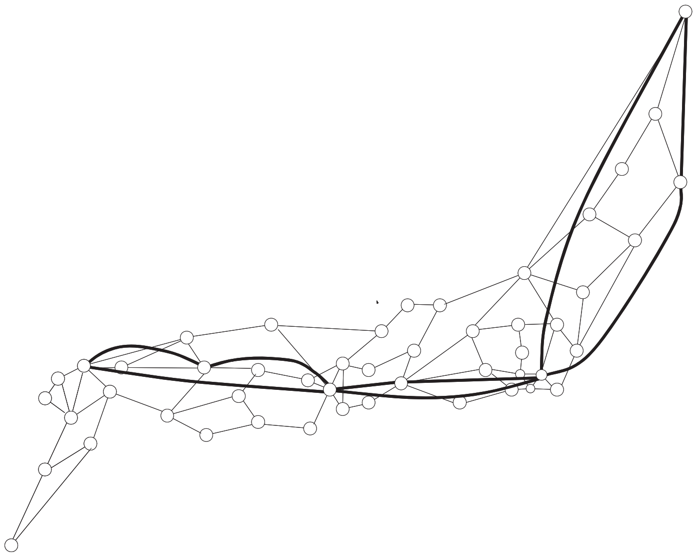


Figure 3.4: NTT's 49-node backbone network

fiber. The processing capacities of the electronic routers (see Fig. 2b), expressed as μ, are set to 5.6 Tbps [48] and 16 Tbps, respectively.

We evaluate the respective logical topology by deriving the average delay, throughput, and number of fiber amplifiers obtained by the corresponding algorithms. The average delay is defined as follows.

$$\bar{T} = \frac{1}{N(N-1)} \sum_{s=1}^{N} \sum_{d=1}^{N} D_{sd} \tag{3.3}$$

where N is the number of nodes in the network and D_{sd} is the delay on the traffic between nodes s and d. In our architectural model shown in Fig. 2b, the delay experienced at a node consists of the processing delay, the transmission delay, and the propagation delay. Thus, D_{sd} is represented as

$$D_{sd} = \sum_{i=1}^{N} a_i^{sd} \cdot QD_i + \sum_{i=1}^{N} \sum_{j=1}^{N} b_{ij}^{sd} \cdot TD_{ij} + \sum_{i=1}^{N} \sum_{j=1}^{N} b_{ij}^{sd} \cdot PD_{sd}. \tag{3.4}$$

The notation used in Eq. (3.4) is as follows.

QD_i: Delay for processing at the IP router on node i. We determine this by using an M/M/1 queueing model.

TD_{ij}: Transmission delay experienced in the buffer of the lightpath between node i and node j. If there are several lightpaths, the IP traffic is divided into flows such that the rate of transmission is identical on each of the lightpaths. The delay at the buffer is also calculated by using an M/M/1 queueing model.

PD_{sd}: Propagation delay of lightpaths between end nodes s and d.

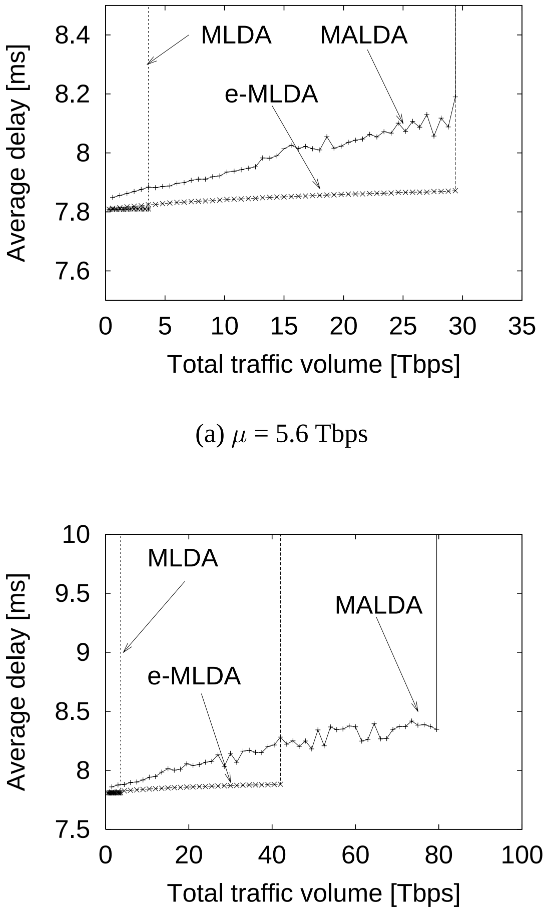
a_i^{sd}: If the IP router on node i processes the traffic from node s to node d, then $a_i^{sd} = 1$. Otherwise $a_i^{sd} = 0$.

b_{ij}^{sd}: If the traffic from node s to node d goes through the lightpath between node i and node j, then $b_{ij}^{sd} = 1$. Otherwise $b_{ij}^{sd} = 0$.

3.4.2 Evaluation Results

To obtain the numerical results, we use the following assumptions and parameter settings. For MLDA, we assume that 1,000 wavelengths are always used. For e-MLDA and MALDA, we set the utilization rate of each lightpath to be below 70%. If the rate of utilization of a lightpath is greater than that value, we set up new lightpaths. For safer operation, we might limit the maximum amount of traffic accommodated at the IP router to, e.g., 70% of its processing capability. In this evaluation, however, we regard the IP router's processing capacity as the maximum amount of traffic accommodated by it for simplicity. In the case of e-MLDA, the logical topology is built on the assumption that 1,000 wavelengths are available. Then, we have simply removed the unnecessary optical amplifiers after the logical topology has been built for fair comparison with MALDA. In MALDA, the number of amplifiers on each fiber is determined by the algorithm presented in Section 3.3. For this, we have assumed that $W_1 = 200$, $W_i = 100$, and $N_{max} = 9$.

Figures 5a, 5b, 6a, and 6b show the dependence of average delay on the total requested traffic for the traffic matrices P_1 and P_2. Each figure depicts the case for IP routers with one of the two capacities. From these figures, we can see that the average delays on LT_{e-MLDA} and LT_{MALDA} may decrease even when the requested traffic volume increases. This is because both of those logical topologies change according to the requested traffic volume. In



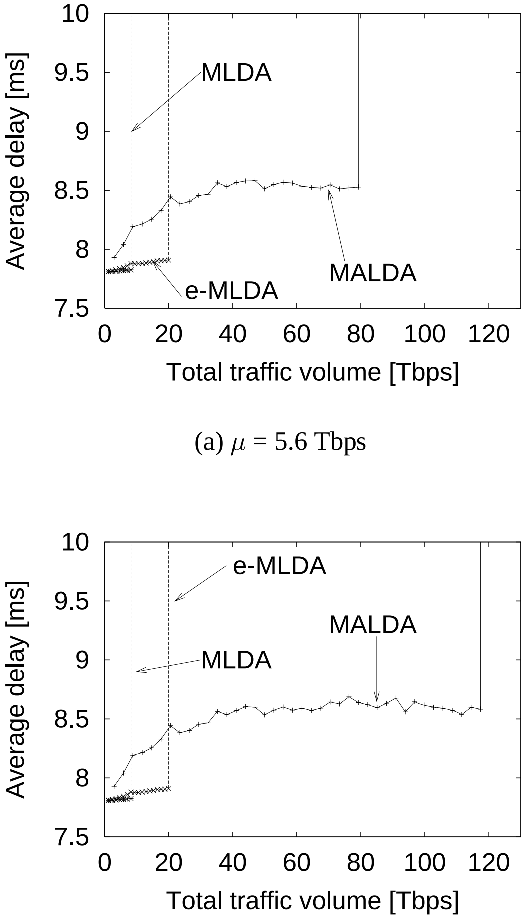
(a) μ = 5.6 Tbps

(b) μ = 16 Tbps

Figure 3.5: Average delay with traffic pattern P_1

Figs. 5a, 5b, 6a, and 6b, the delay on LT_{MALDA} is always larger than that on LT_{e-MLDA} because MALDA tries to accommodate traffic by using existing lightpaths, whereas e-MLDA sets up new lightpaths since e-MLDA is able to utilize more wavelengths than MALDA is on each fiber. This results in a higher rate of utilization of lightpaths by LT_{MALDA} than by LT_{e-MLDA}. LT_{MLDA} shows the smallest delay since MLDA always utilizes all the wavelengths regardless to the requested traffic volume.

We next discuss the throughput of each of the logical topologies. Here, the throughput is defined as the minimum requested traffic volume (more precisely, the scale-up factor α) such that the average delay reaches saturation. When we cannot set up all the lightpaths required or we cannot make the load of all the IP routers under their processing capacity,



(a) μ = 5.6 Tbps

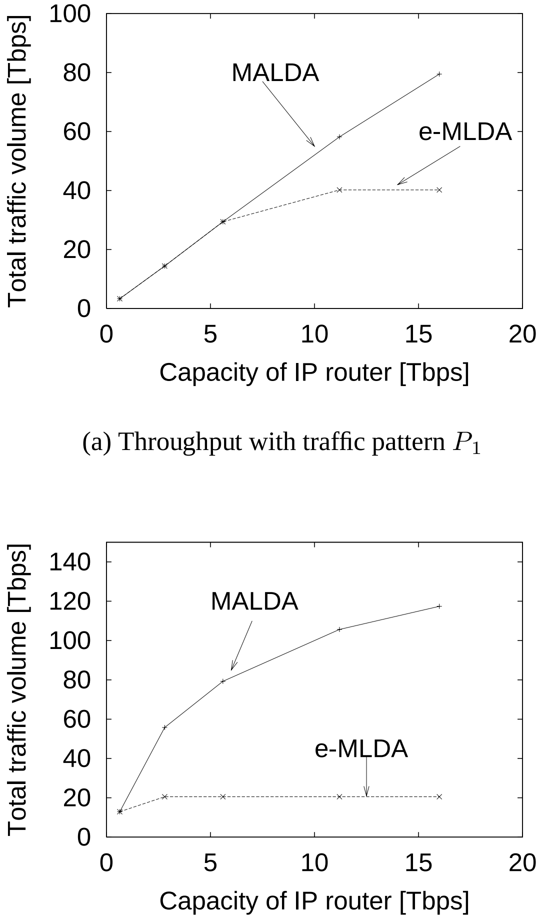
(b) μ = 16 Tbps

Figure 3.6: Average delay with traffic pattern P_2

the average delay goes to infinity. In Fig. 5a ($\mu = 5.6$ Tbps), LT_{MALDA} accommodates as much traffic as LT_{e-MLDA}. This is because the bottleneck for the network in this case is the processing capacity of the IP router. When the processing capacity of the IP router is large ($\mu = 16$ Tbps), LT_{MALDA} shows a higher throughput than LT_{e-MLDA} in Fig. 5b. In this case, the large capacity of the respective IP routers means that the bottleneck for the network is not the processing capacity but the link capacity. In P_2, the node-pairs whose source nodes are apart from their destinations require more lightpaths than those in P_1. As a result, The bottleneck is the processing capacity of a IP router at the intermediate node. MALDA effectively cuts lightpaths at the different intermediate nodes so that the load of IP routers are distributed. This results in higher throughput of LT_{MALDA} than that of LT_{e-MLDA} in Figs. 6a and 6b.

LT_{MLDA} shows much lower throughput than others because MLDA sets up one-hop lightpaths while MALDA and e-MLDA set up multi-hop lightpaths. Setting up one-hop lightpaths leads to a poor utilization rate of each lightpath because the lightpath of each packet flow is limited while the lightpath is shared when multi-hop lightpaths are set up. To see the above discussions clearly, we show the throughput values dependent on the capacity of the IP router in Figs. 7a (traffic pattern P_1) and 7b (traffic pattern P_2). The results show that LT_{MALDA} accommodates more traffic than LT_{e-MLDA} does if the processing capacity of the IP router increases. LT_{e-MLDA} shows constant throughput in spite of increasing capacity of the IP router due to a lack of wavelengths. On the other hand, the throughput of LT_{MALDA} increases as the capacity of the IP router becomes high since only the IP router's capacity is the network bottleneck of the logical topology. The upper bound on the throughput of LT_{e-MLDA} when P_1 is used (40.2 Tbps) is about twice as much as that when P_2 is used (20.5 Tbps). In P_1, the traffic volume requested by neighboring nodes are relatively larger than others. As a result, a lot of lightpaths are set up between neighboring nodes that can be shared by IP packets, which leads to higher throughput in P_1 than that in P_2. Overall, MALDA can more effectively utilize the bandwidth of the lightpaths than e-MLDA does.

The required numbers of optical fiber amplifiers are shown in Figs. 8a, 8b, 9a, and 9b. In LT_{e-MLDA}, unnecessary optical amplifiers are removed. The results of LT_{e-MLDA} are plotted for traffic volumes below 40.2 Tbps in P_1 and 20.5 Tbps in P_2 because it cannot



(a) Throughput with traffic pattern P_1

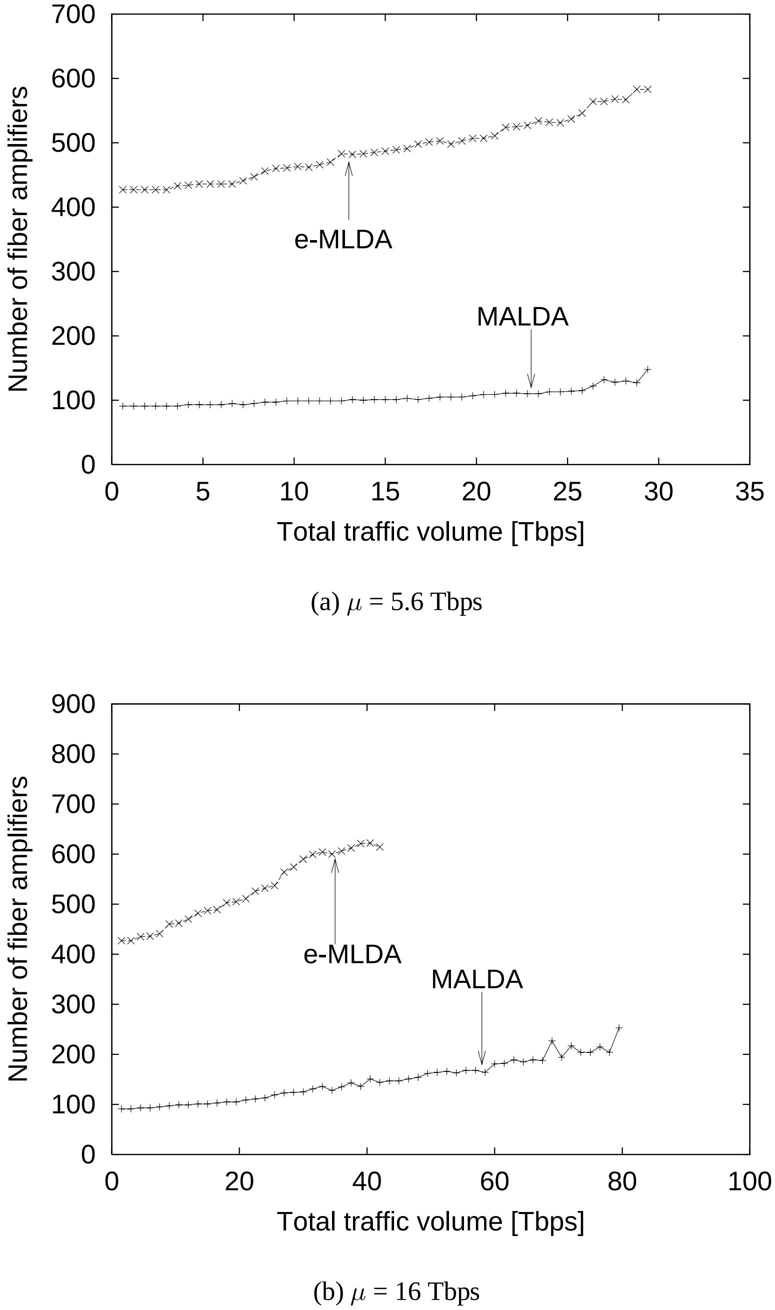
(b) Throughput with traffic pattern P_2

Figure 3.7: Throughput of each logical topology

accommodate traffic volumes beyond 40.2 Tbps and 20.5 Tbps, respectively. The result of LT_{MLDA} is eliminated since it always utilizes all the optical fiber amplifiers (819 amplifiers). Note that the number of optical fiber amplifiers does not always increase as the total traffic volume increases. This is because the number of intermediate nodes at which lightpaths are split may increase when the total traffic volume increases. As such a intermediate node increases, the wavelength continuity constraint is more relaxed, which could result in effective utilization of the wavelengths. We see that LT_{MALDA} only requires about one-fifth of the optical fiber amplifiers that LT_{e-MLDA} needs in P_1 and P_2.

3.5 Conclusion

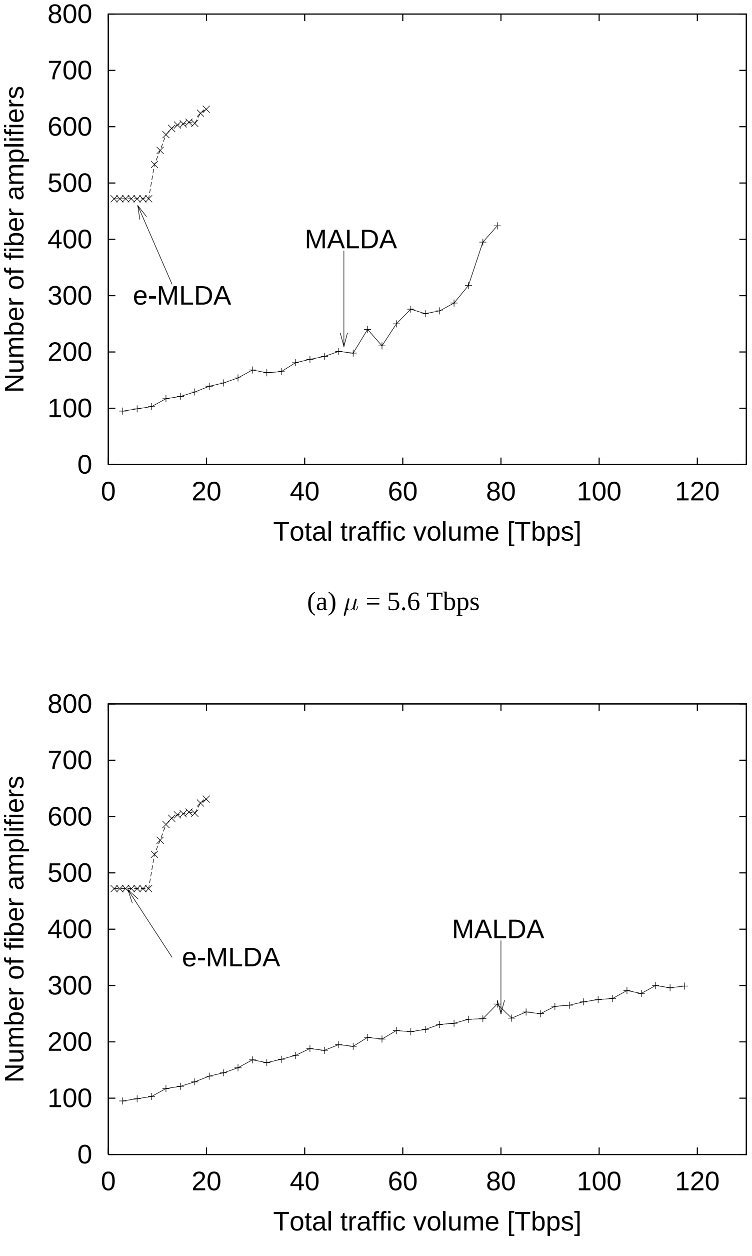
In this chapter, we have proposed e-MLDA (extended MLDA), a new heuristic algorithm for the design of logical topologies to be overlaid on WDM networks. The resulting topology is based on the actual levels of node-to-node traffic demand. We went on to propose MALDA (Minimum number of fiber Amplifiers Logical topology Design Algorithm) for which the objective function is to minimize the number of fiber amplifiers deployed in the logical topology. Our algorithms are evaluated by comparing them with the conventional method in terms of average delay, throughput, and number of optical fiber amplifiers deployed in the network. The results have shown that MALDA only needs about one-fifth of the fiber amplifiers that e-MLDA does, while MALDA is able to accommodate as much traffic as e-MLDA. Furthermore, when the processing capacity of IP routers is high, MALDA can accommodate more traffic than e-MLDA does. Our results indicate that MALDA is preferable in terms of designing a low-cost logical topology.



(a) μ = 5.6 Tbps

(b) μ = 16 Tbps

Figure 3.8: Number of optical fiber amplifiers needed by each logical topology with traffic pattern P_1



(a) μ = 5.6 Tbps

(b) μ = 16 Tbps

Figure 3.9: Number of optical fiber amplifiers needed by each logical topology with traffic pattern P_2

Chapter 4

Design of Edge-Nodes with Effective Wavelength Conversion

In this chapter, we discuss the design of edge nodes, to which links with different numbers of wavelengths are connected, in large-scale wavelength-routed networks. We need to cope with the diversity in the numbers of wavelengths by wavelength conversion on edge nodes. In previous researches for wavelength converter placement problem, the main purpose is eliminating fragmentation of wavelength resources between adjacent links that have the same number of wavelengths multiplexed. In large-scale wavelength-routed networks, however, we also need to utilize wavelength converters to cover the difference in the numbers of multiplexed wavelengths. We propose an edge node architecture that has fixed wavelength converter to solve the above-mentioned difference. This architecture offers total cost reduction at the edge nodes.

4.1 Diversity in the Numbers of Wavelengths in Wavelength-Routed Networks

Appearances of new services such as GRID computing lead to the need for end-to-end lightpath provisioning (e.g., OptIPuter [49] ,Lambda Grid [50] and λ-computing [51, 52]). One promising candidate for the networks that realize the end-to-end lightpath provisioning is a wavelength-routed network with overlay model (i.e., controls of users and a carrier are

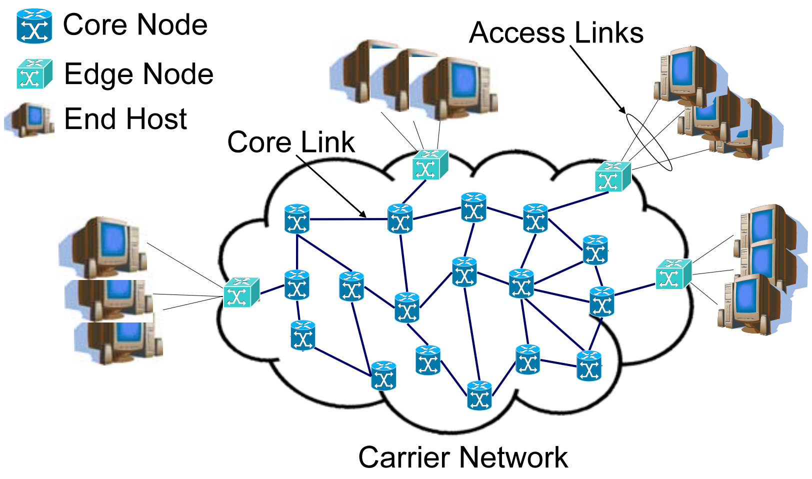


Figure 4.1: Wavelength-routed network with overlay model

separated). In this network, end hosts of users are connected to a carrier network via access links (Fig.4.1).

The important features of the network are the increase in the number of access links and the diversity in the numbers of wavelengths multiplexed on links. More users motivated by the new serivices connect the access links for the lightpath provisioning, which leads to increase in the number of access links. The diversity arises due to the following actions. Users prepare access links and communication interfaces with a few wavelengths multiplexed for cost reduction. In particular, the common wavelengths (e.g., some wavelengths in C band) may be used among most users. On the other hand, a carrier prepares a core link with tens or hundreds of wavelengths multiplexed for accommodating traffic from access links. The above-mentioned increase in the number of access links motivates such large number of wavelengths on core links. In this senario, it is important to cope with the difference in the numbers of wavelengths multiplexed on access and core links because the wavelength continuity constraint (i.e., the same wavelength must be assigned to a lightpath on links along a route) must be satisfied.

Wavelength conversion improves the blocking performance of wavelength-routed networks. Wavelength converters change an input wavelength to another output one, thus eliminate the fragmentation of wavelength resource. As a result, the utilization rate of wavelength resource is improved. Because wavelength converters remain expensive in the near future, we need to minimize the number of wavelength converters deployed for achieving an objective performance. In order to cost-effectively utilize wavelength converters, methods for deployment of wavelength converters have been developed. In [28, 29], deploying wavelength converters only on a few nodes leads to the cost reduction. In [30], deploying wavelength converters on about 1–5% of all ports in a network achieves the blocking performance close to performance of a case for full-complete wavelength conversion where all ports on all nodes are equipped with wavelength converters.

In conventional researches [28, 29, 30], they focus on networks where each link has an identical number of wavelengths multiplexed. In those networks, wavelength converters are used for eliminating fragmentation of wavelength resources between adjacent links that have the same number of wavelengths multiplexed. In a wavelength-routed network with overlay model, however, we also need to utilize wavelength converters to cover the difference between the numbers of wavelengths on links. If the number of wavelength converters used for covering the difference is much larger, covering the difference with the lowest wavelength converter cost is inevitable for constructing the cost-effective network.

In this chapter, we first show that edge nodes, to which both access and core links are attached, need much more wavelength converters than core nodes, to which core links are only attached. Then, we propose an ingress edge node architecture with *fixed wavelength converters* that convert a predetermined input wavelength to another predetermined output wavelength. In our node architecture, fixed wavelength converters evenly distribute wavelengths from input access links to wavelengths on an output core link. Adopting fixed wavelength converters for distribution of input wavelengths leads to lower costs than nodes with *full wavelength converters* that convert any input wavelengths to another output one.

The rest of this chapter is organized as follows. Section 4.2 includes an explanation of wavelength-routed networks and a simulation result that shows edge nodes need most wavelength converters. In section 4.3, we discuss cost models of full and fixed wavelength converters and propose an ingress edge node architecture with fixed wavelength converters. We then compare our node architecture with a node architecture that only uses full wavelength converters and show our node architecture reduces wavelength converter cost. Finally, section 4.4 concludes this chapter.

4.2 Effect of Deploying Wavelength Converters on Edge Nodes in Wavelength-Routed Networks with Overlay Model

4.2.1 Wavelength-Routed Network with Overlay Model

There are some inter-connection models between optical networks and other networks or end hosts [53]. In the peer model, optical networks and others are treated as a single network and they exchange topological and routing information with each other. In the overlay model, on the other hand, they are independent and do not exchange those information. From the security viewpoint, we adopt the overlay model because advertising internal information of the carrier network to end hosts is not safe.

In wavelength-routed networks with overlay model, end hosts are connected to a carrier network via access links. Each end host establishes lightpaths to another one for communication. We assume end hosts as computers providing the grid computing [54]. A carrier network consists of nodes and fibers. We refer to the node, to which access links are attached, as an edge node and another node as a core node. To investigate how many wavelength converters are needed only for covering the difference in wavelength number, we assume that an edge node is connected to a single core node and does not relay lightpaths among core links. We assume that a few wavelengths are multiplexed on an access link and tens or hundreds of wavelengths are multiplexed on a core link.

4.2.2 Node Architecture

Figs. 4.2 and 4.3 depict an edge and a core node architecture. A node consists of demultiplexers (DEMUX), multiplexers (MUX), Optical Cross-Connects (OXC) and full wavelength converters. When a node relays a wavelength for establishment of a lightpath, a DEMUX first demultiplexes an input signal into each wavelength. Then, an OXC switches each wavelength to an appropriate output port. Finally a MUX multiplexes wavelengths into an output signal. When the wavelength same as an input wavelength is not idle on an output fiber, the input wavelength is switched to a full wavelength converter and converted

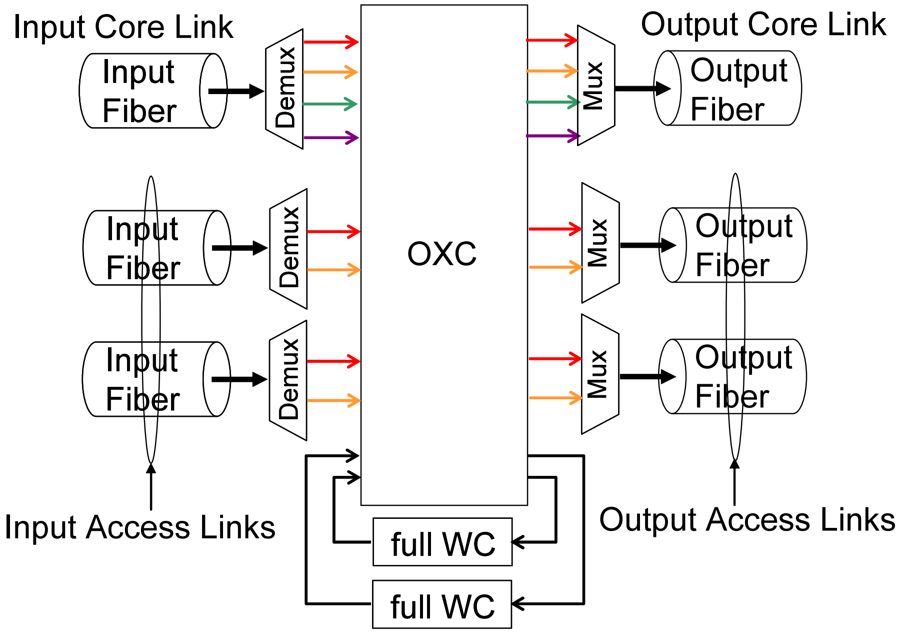


Figure 4.2: Edge node architecture

to another wavelength idle on an output fiber.

Full wavelength converters are deployed on nodes in a trunk-type basis [55]. In the trunk-type, full wavelength converters are shared among input ports. The input port that actually needs wavelength conversion is switched to an output port with a full wavelength converter. As a result, the number of full wavelength converters deployed is reduced.

4.2.3 Optimal Distribution of Full Wavelength Converters to Edge/Core Nodes

We verify that edge nodes need much more full wavelength converters than core nodes in wavelength-routed networks with overlay model. To achieve this, we obtain by simulation an optimal distribution of full wavelength converters to edge and core nodes, which leads to minimizing the call blocking probability with given full wavelength converters. We use NSFNET (Fig. 4.4) as a network model. An edge node is attached to each core node. End hosts are attached to each edge node with access links. Followings are parameters in simulation. Values of parameters are shown in Tab. 4.1.

L_a : Number of access links attached to an edge node.

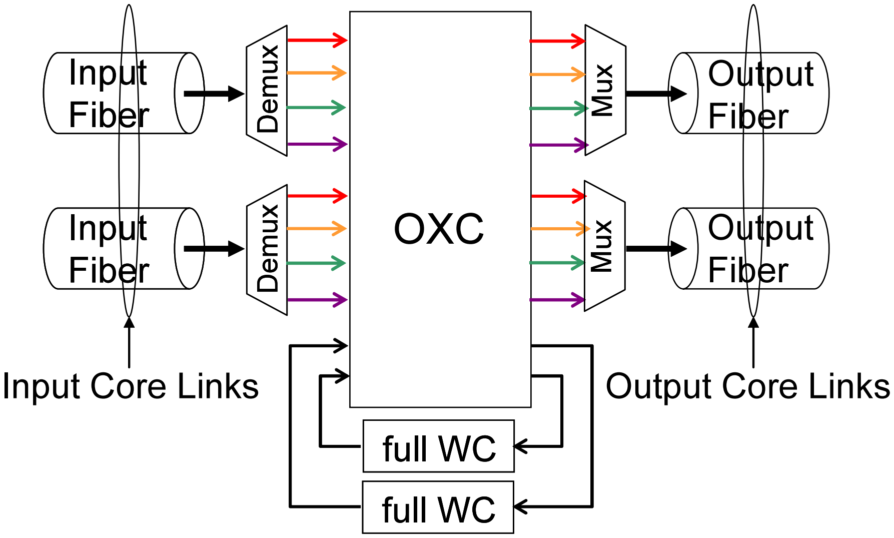


Figure 4.3: Core node architecture

W_a : Number of wavelengths multiplexed on an access link.

W_c : Number of wavelengths multiplexed on a core link.

a : Arrival rate to an end host. Poisson arrival.

$\frac{1}{\mu}$: Average holding time of a lightpath. A holding time follows exponential distribution.

ρ_c : Load on an output core link attached to an edge node. The load on output core link is defined as a ratio of arrival rate to an edge node to the number of wavelength on output core link. $\rho_c = \frac{aL_a}{\mu W_c}$.

In simulation, we obtain blocking probabilities caused by fragmentation of wavelength resource (i.e., there exist idle wavelengths on each link but no identical wavelength is idle on the consecutive links along the route) when ratio between the number of full wavelength converters on edge nodes and that on core nodes varies. After determining the ratio, we uniformly deployed full wavelength converters among edge or core nodes. We selected the total number of full wavelength converters so that blocking by fragmentation of wavelength resource does not occur in an optimal distribution. A lightpath request arrives at end hosts following a. A destination end host is uniformly selected from a set of end hosts that

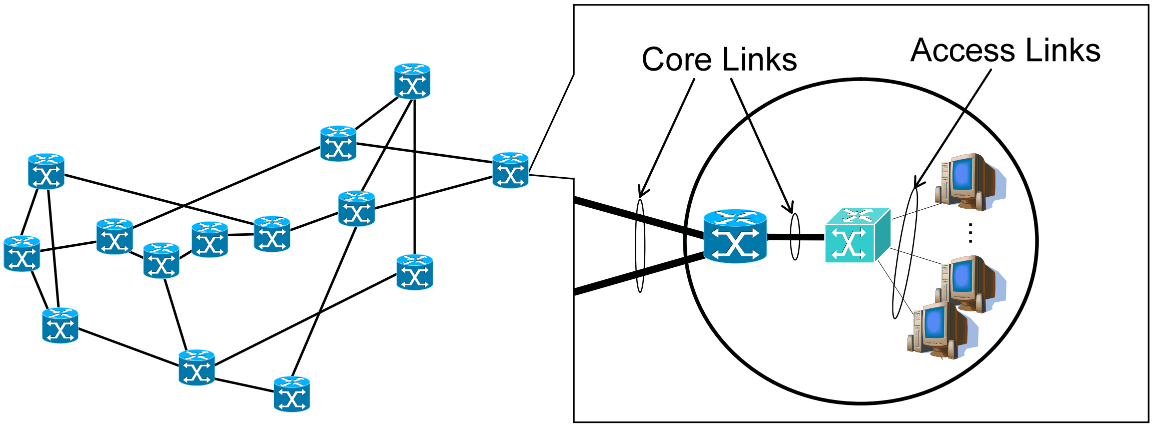


Figure 4.4: NSFNET

Table 4.1: Parameters used in Simulation

L_a	W_a	W_c	a	$\frac{1}{\mu}$	ρ_c
$\frac{\rho_c \mu W_c}{a}$	8	$\{16, 32, 64, 128\}$	4	1	0.5

are not connected to the same edge node as the source end host is. We use a minimum-hop routing algorithm for route selection. For wavelength assignment, we use a modified version of MFF (Modified First-Fit) [30], in which we randomly select an idle wavelength instead of First-Fit policy. Concretely, we divide route of a lightpath into segments, in which wavelength continuity constraint must be satisfied, in following order.

1. A set of links from a source end host to a destination end host corresponds to a segment.

2. Sets of links from a source end host to an ingress edge node, from an ingress edge node to an egress edge node, from an egress edge node to a destination edge node individually correspond to a segment.

3. Each link on a route corresponds to a segment.

Then, we randomly select an idle wavelength in each segment. When two consecutive segments use different wavelengths, a wavelength converter is used in the intermediate node. A lightpath request is blocked if there exists no idle wavelength on a link or required wavelength conversion cannot be performed because of the lack of full wavelength converters.

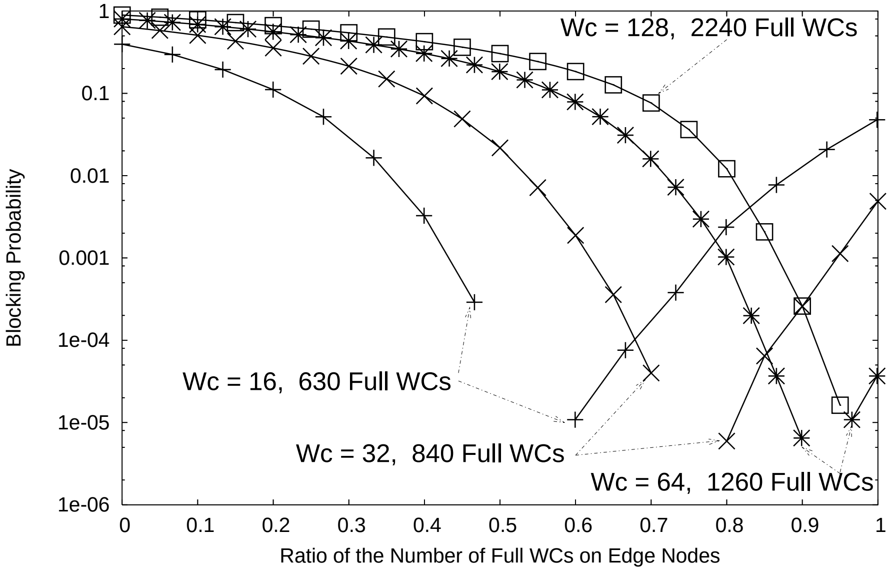


Figure 4.5: Blocking probabilities with different ratio of the number of full wavelength converters on edge nodes

Figure 4.5 shows blocking probabilities caused by fragmentation of wavelength resource with different ratio between the number of full wavelength converters on edge nodes and that on core nodes. Blocking caused by the lack of idle wavelengths is not counted because wavelength conversion cannot avoid it.

When W_c is 16, deploying about 53% of given converters on edge nodes minimizes the blocking probability. Optimal ratios with $W_c = 32$, 64 and 128 are about 75%, 93% and 100%. These results mean that, for minimizing a blocking probability with given converters, more number of full wavelength converters should be deployed on edge nodes as the difference between the numbers of access and core links gets larger. This is because more fragmentation of wavelength resources occurs between access and core links in wavelength-routed networks with overlay model. Therefore, reducing the number of full wavelength converters on edge nodes leads to reducing wavelength converter cost in the whole network. In the next section, we propose an edge node architecture with reduced number of full wavelength converters.

4.3 Edge Node Architecture with Fixed Wavelength Converters

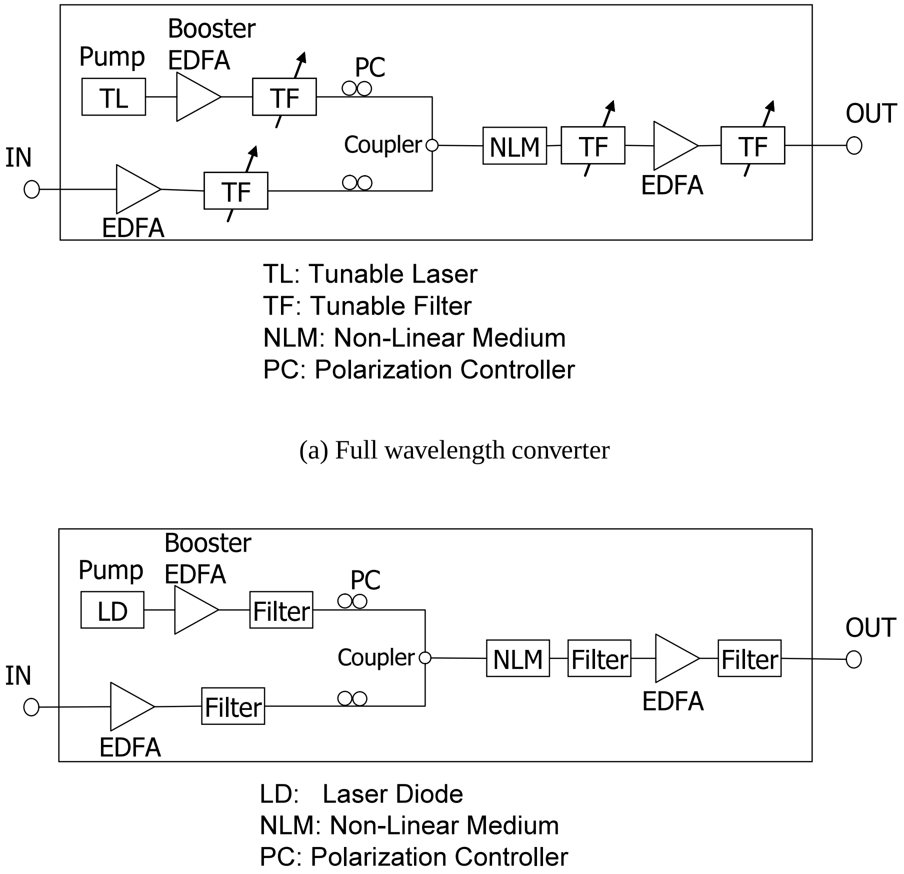
4.3.1 Wavelength Converter Model

In this section, we introduce two kinds of wavelength converters; full wavelength converters and fixed wavelength converters. We further discuss the cost ratio of those converters. Full wavelength converters and fixed wavelength converters are realized with FWM (Four-Wave Mixing) [56, 57, 58].

Architectures for a full wavelength converter and a fixed wavelength converter are shown in Fig. 4.6. The wavelength conversion process is as follows. In both wavelength converters, the input beam is amplified by EDFA (Erbium-Doped Fiber Amplifier). Then, the input beam is combined with the pump beam that was amplified to the a power of more than 20 dBm by a booster EDFA. The combined beam is input into NLM (Non-Linear Medium) and a beam whose wavelength is different from both the input beam and the pump beam is generated. After filtered, amplified and reshaped, the generated beam is output as a converted wavelength. In a full wavelength converter, a tunable laser is used as a pump source because an output wavelength needs to be adjusted by changing a wavelength of the pump beam. In fixed wavelength converter, on the other hand, a laser diode is used as a pump source that produces a fixed wavelength.

To evaluate how much cost-reduction fixed conversion offers, we need to determine the ratio of a full wavelength converter cost to a fixed wavelength converter cost. The ratio depends on costs of a tunable laser, a laser diode and other devices. In this chapter, we decided the ratio based on the following internal study [59].

- The tunable laser cost to a laser diode (with a wavelength locker) cost ratio is about 10.
- A tunable laser costs more than 1 million yen (9,000 dollars) and will not go down in the near future.
- The design cost of non-linear medium is as high as a tunable laser cost. However, the cost of non-linear medium can be much lower than a tunable laser cost when it is



(a) Full wavelength converter

(b) Fixed wavelength converter

Figure 4.6: Architectures for wavelength converters

mass-produced.

- The cost of EDFA can be 30 or 40% of a tunable laser cost when it is a module type and mass-produced.

Another forecast of the cost of EDFA is about 1,000 dollars [60]. The booster EDFA will more expensive than other EDFAs. Therefore, in this chapter, we assume that the cost of booster EDFA follows the value in [59] and total costs for other EDFAs, NLM, polarization controllers is 1/10 of a tunable wavelength converter cost.

From the above discussion, the full wavelength converter cost to fixed wavelength converter cost ratio can be at least 3. We may expect a larger ratio: for example, when opto-electronic conversion instead of all-optical conversion is used, the ratio will be almost the same as 10, the ratio of a tunable laser cost to a laser diode cost. If we apply waveband conversion [61] for fixed wavelength conversions of a set of wavelengths (e.g., wavelengths multiplexed on an access link), much larger ratio may be obtained. In this chapter, based on the above discussion, we investigate whether fixed conversion can reduce wavelength converter cost of a wavelength-routed network with overlay model when the ratio ranges from 3 to 10.

4.3.2 Node Architecture with Fixed Wavelength Converters

Figure 4.7 depicts out node architecture with fixed wavelength converters. Fixed wavelength converters are deployed on input ports from input access links. We utilize fixed wavelength converters to uniformly distribute lightpath requests from input access links to wavelengths on an output core link.

Correspondence between an input wavelength and an output wavelength of a fixed wavelength converter is determined as following. λ_w ($0 \leq w \leq W_a - 1$) on kth ($1 \leq k \leq L_a$) input access link is converted to $\lambda_{((k-1)W_a+w) \bmod W_c}$. When an output wavelength is the same as an input wavelength, no fixed conversion is performed. In Fig. 4.7, Wavelengths on the upper access link do not need wavelength converters because the same wavelengths are assigned to them on an output core link. On the other hand, a fixed wavelength converter is deployed for each wavelength on the lower access link to convert λ_0 and λ_1 to λ_2 and λ_3, respectively. When we cannot avoid competition with only fixed wavelength converters because multiple wavelengths are converted to the same wavelength on an output core link, we use full wavelength converters.

Fixed conversion is not performed on egress edge nodes that relay wavelengths from an input core link to an output access link. This is because even if fixed wavelength converters are on egress edge nodes, the wavelength assigned to lightpaths on core links are seldom identical to expected input wavelengths of the fixed wavelength converters.

4.3.3 Numerical Examples

We compare wavelength converter cost in our edge node architecture with that in an edge node architecture that only uses full wavelength converters by simulation. To evaluate how much wavelength converter cost on an ingress edge node is reduced, we use a network

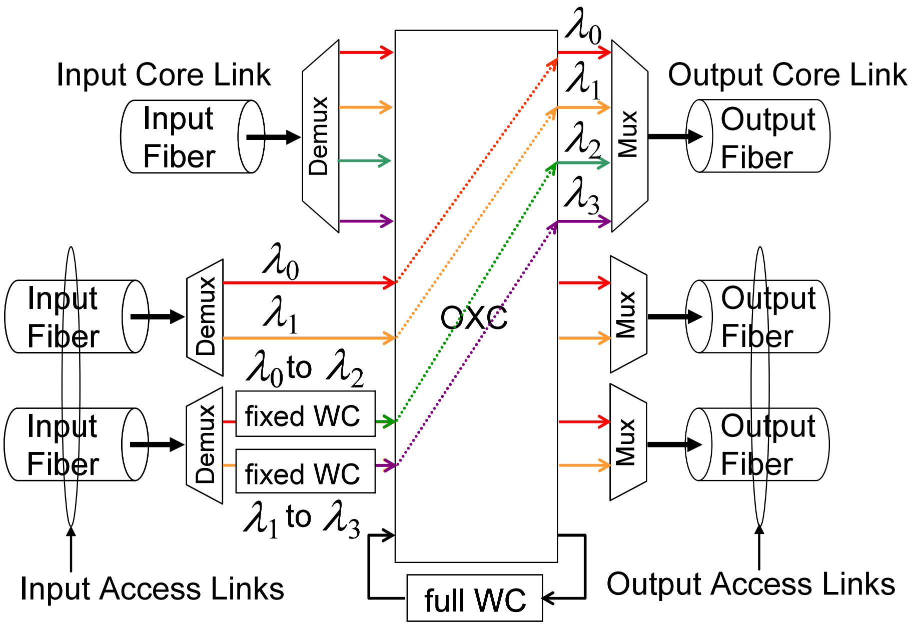


Figure 4.7: Node architecture with fixed wavelength converters ($L_a = 2, W_a = 2, W_c = 4$).

model that consists of two edge nodes, two core nodes and three core links (Fig. 4.8). Performance metrics are (1) the number of full wavelength converters needed on an ingress edge node and (2) total wavelength converter costs needed on an ingress edge node. There are 8 and 128 wavelengths multiplexed on access and core links, respectively. Lightpath requests arrive at source end hosts according to a Poisson process with rate a. A destination end host is selected among all destination end hosts according to uniform distribution. The holding time for lightpaths ($1/\mu$) follows an exponential distribution with an average of 1. We used the wavelength assignment method in section 4.2.3. To investigate whether our architecture reduces cost of an ingress edge node, we focus on the ingress edge node in Fig. 4.8. Core nodes and an egress edge node are equipped with unlimited number of full wavelength converters.

We regard X full wavelength converters as the sufficient number of full wavelength converters on an ingress edge node when the node with X full wavelength converters provides almost the same blocking performance as the node that has unlimited number of wavelength converters. Therefore, we introduce *approximation factor* [62] as following;

$$\frac{P_B(X) - P_B(\infty)}{P_B(0) - P_B(\infty)} < \varepsilon. \tag{4.1}$$

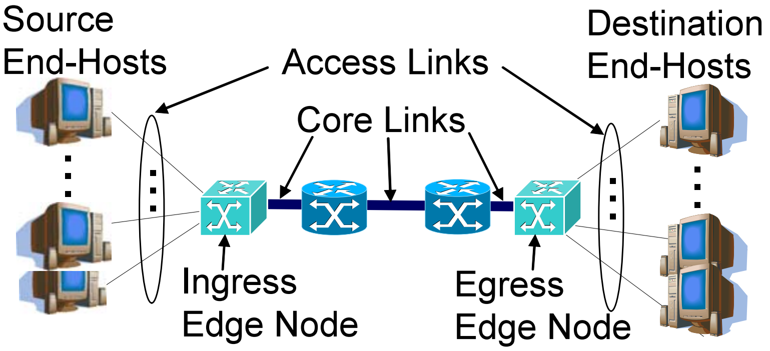


Figure 4.8: Network model (2 core nodes, 2 edge nodes, 3 core links)

$P_B(X)$ is a blocking probability when X full wavelength converters are deployed on an ingress edge node. $P_B(0)$ and $P_B(\infty)$ are blocking probabilities when no and unlimited number of full wavelength converters are deployed on an ingress edge node, respectively. We set ε to 0.001, which is low enough to achieve an objective end-to-end blocking performance in connection-oriented networks (e.g., a target probability of end-to-end blocking is between 0.02 and 0.05 in ISDN [63]). In this case, difference of blocking probabilities between $P_B(\infty)$ and $P_B(X)$ is under ε as following;

$$P_B(X) < \varepsilon(P_B(0) - P_B(\infty)) + P_B(\infty) < \varepsilon + P_B(\infty).$$

Figure 4.9 shows the minimum number of X in Eq. (4.1). The horizontal axis represents load on the output core link that is attached to the ingress edge node (ρ_c). The graph label “full WC” indicates a node architecture that only uses full wavelength converters and “fixed WC” does our architecture. The load is proportional to the number of input access links attached to the ingress edge node.

In the node architecture that only uses full wavelength converters, the number of full wavelength converters increases proportionally to load on the output core link. This is because more lightpath requests from access links compete for the same wavelength on an output core link as the load increases.

In our node architecture, no full wavelength converter is needed when the load is lower than 0.5. This is because input wavelengths on each input access link are converted to different wavelengths on an output core link with fixed converters. When the load is over 0.5,

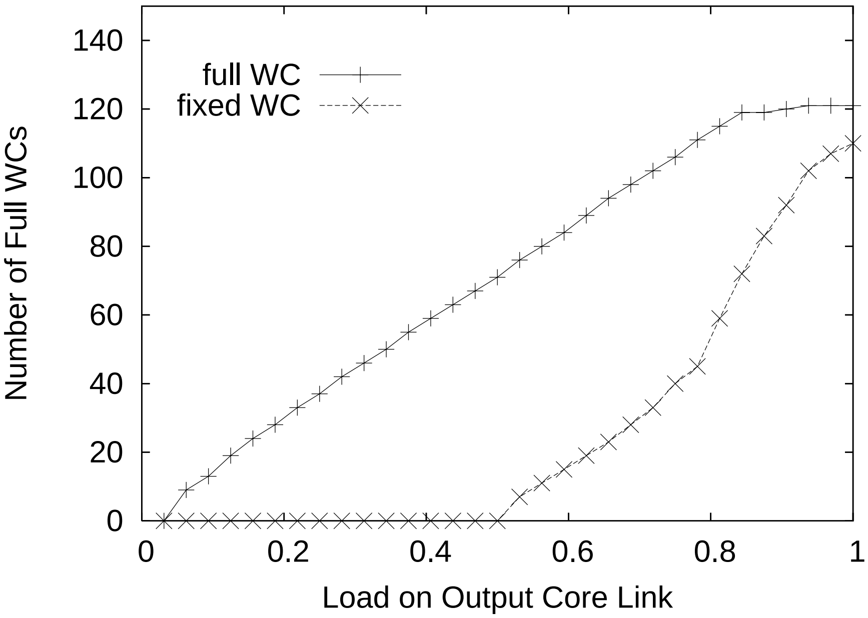


Figure 4.9: Number of full wavelength converters on an ingress edge node ($a = 4$)

the number of full wavelength converters needed increases because we need to perform full wavelength conversion in addition to fixed wavelength conversion. However, the number of full wavelength converters is greatly reduced in our node architecture.

Figure 4.10 shows the number of full wavelength converters when the load on the output core link is fixed and an arrival rate of lightpath requests changes. The horizontal axis represents an arrival rate of lightpath requests at an source end host. In this case, the number of input access links decreases as the arrival rate increases. We set the load to around 0.6, which is an average wavelength utilization when networks are actually under operation [30].

In both architectures, the number of full wavelength converters decreases as the arrival rate increases. This is because larger arrival rate leads to more blocking on input access links and less lightpath requests arrive to the output core link. In our node architecture, no full wavelength converter is needed when the arrival rate is larger than 4 because competition for the same wavelength on an output core link is avoided only with fixed wavelength converters. The above simulation results show that utilizing fixed wavelength converters leads to great reduction of the number of full wavelength converters needed on an ingress edge node.

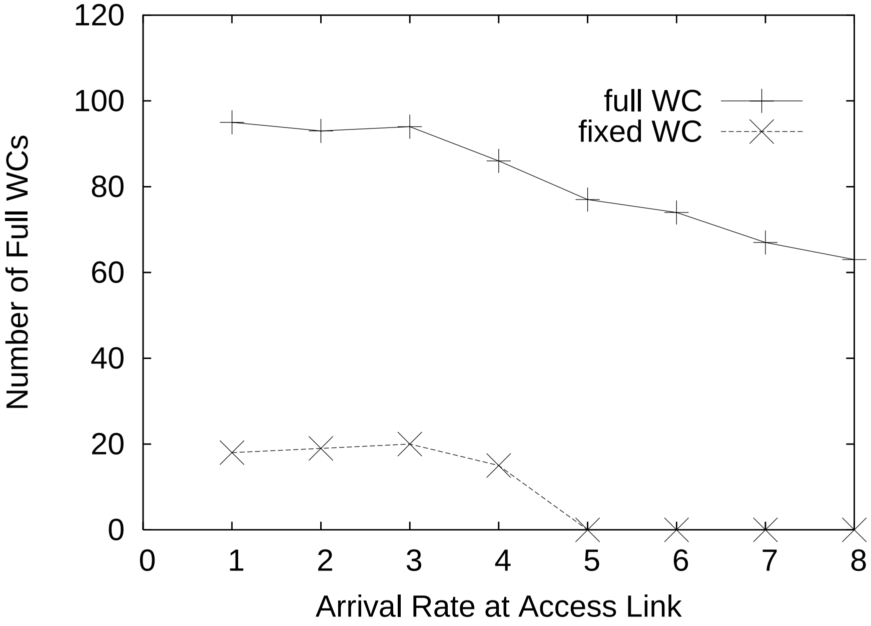


Figure 4.10: Number of full wavelength converters on an ingress edge node when ρ_c is around 0.6

Total wavelength converter costs on an ingress edge node are shown in Fig. 4.11. The horizontal axis is the load on the output core link that is attached to the ingress edge node. A full wavelength converter cost is normalized as 1. In our node architecture, the total wavelength converter cost is the sum of the cost of deployed full wavelength converters and the cost of deployed fixed wavelength converters. We determine ratios of a full wavelength converter cost to a fixed wavelength converter cost as (1) 3:1, (2) 5:1 and (3) 10:1.

In Fig. 4.11, we first focus on wavelength converter costs when the load is around 0.6. When the load is around 0.6, wavelength converter cost in ours are about 79 % (cost ratio 3:1), 56 % (cost ratio 5:1) and 38 % (cost ratio: 10:1) of the cost in the node architecture that only uses full wavelength converters. When the load is lower than 0.5, cost in ours is proportional to the load because the number of fixed wavelength converters only increases. With the load lower than 0.5, cost in ours are about 54 % (cost ratio 3:1), 32 % (cost ratio 5:1) and 16 % (cost ratio 10:1) of the cost in the node architecture with only full wavelength converters.

When load is over 0.8, our architecture shows higher WC cost. However, the load is far higher than that under operation. Therefore, it is important that our node architecture provides lower wavelength converter cost than the node architecture that only uses full

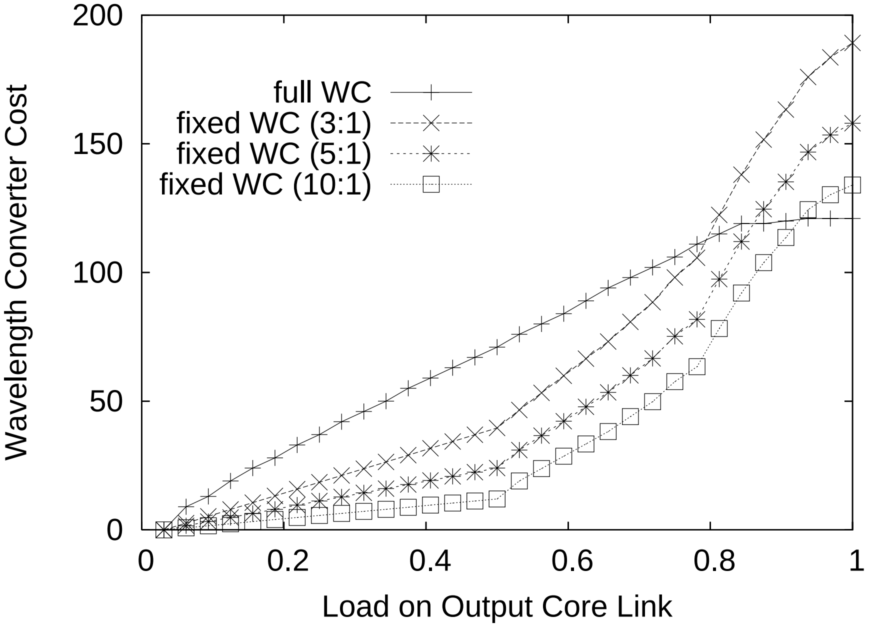


Figure 4.11: Wavelength converter cost ($a = 4$)

WCs when the load is below 0.6.

Wavelength converter costs when the load is fixed to around 0.6 and an arrival rate of lightpath request changes are shown in Fig. 4.12. When a is below 2, our node architecture shows higher cost than the node architecture with only full wavelength converters with cost ratio 3:1 and 5:1. This is because the number of input access links increases as the arrival rate decreases and the increase in the number of input access links leads to more fixed wavelength converters needed. However, in multi-point communication such as grid computing, source end host generally sets up lightpaths to multiple end host, that is, it is important for our node architecture to provide lower wavelength converter cost when a is large.

Figure 4.13 shows the total wavelength converter costs on an ingress edge node when the difference in the numbers of wavelengths on access and core links is relatively small ($W_c = 32$). Our node architecture achieves almost the same cost reduction as that in Fig. 4.11. Therefore utilizing fixed converters leads to the reduction of wavelength converter cost regardless of the difference in the numbers of wavelengths multiplexed on access and core links.

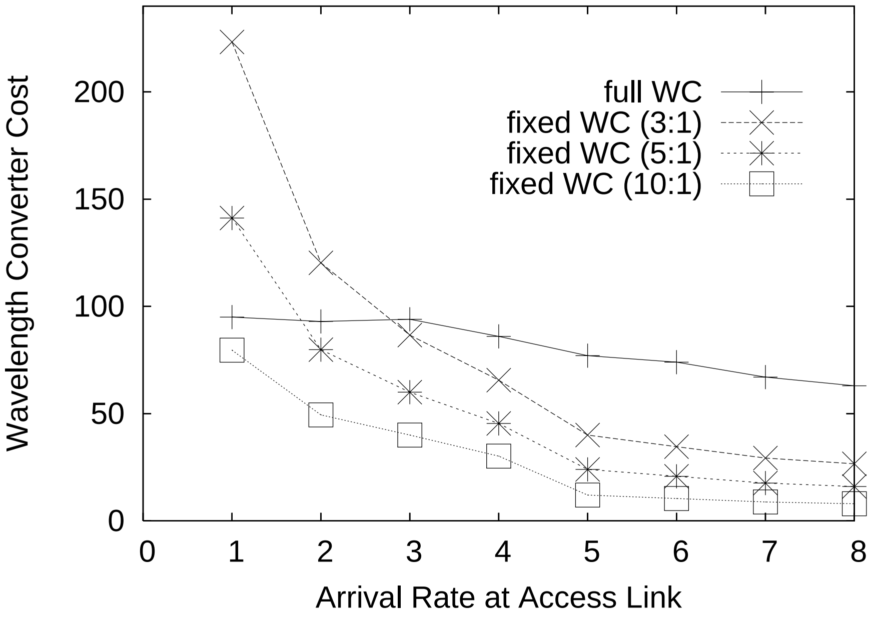


Figure 4.12: Wavelength converter cost when ρ_c is around 0.6

4.4 Conclusion

In this chapter, we investigated the deployment of wavelength converters in wavelength-routed networks with overlay model. We showed that, in wavelength-routed networks with overlay model, most wavelength converters are deployed on edge nodes for covering the difference in the numbers of wavelengths multiplexed on access and core links by simulation. We then proposed an ingress edge node architecture with fixed wavelength converters to reduce the number of full wavelength converters and wavelength converter cost on an ingress edge node. In simulation, our node architecture achieved an objective blocking performance with lower wavelength converter cost than a node architecture that only uses full wavelength converters. When the load on the output core link is in the situation where networks are under operation and wavelength converter cost ratio is 3, our node architecture offered about 21 % cost reduction compared with a node architecture that only uses full wavelength converters. When load is lower, our node architecture offered more than 46 % cost reduction. In addition, fixed wavelength conversion offers more cost reduction as the wavelength converter cost ratio gets larger. Utilizing fixed converters leads to cost reduction regardless of the difference in the numbers of wavelengths multiplexed on access and core links.

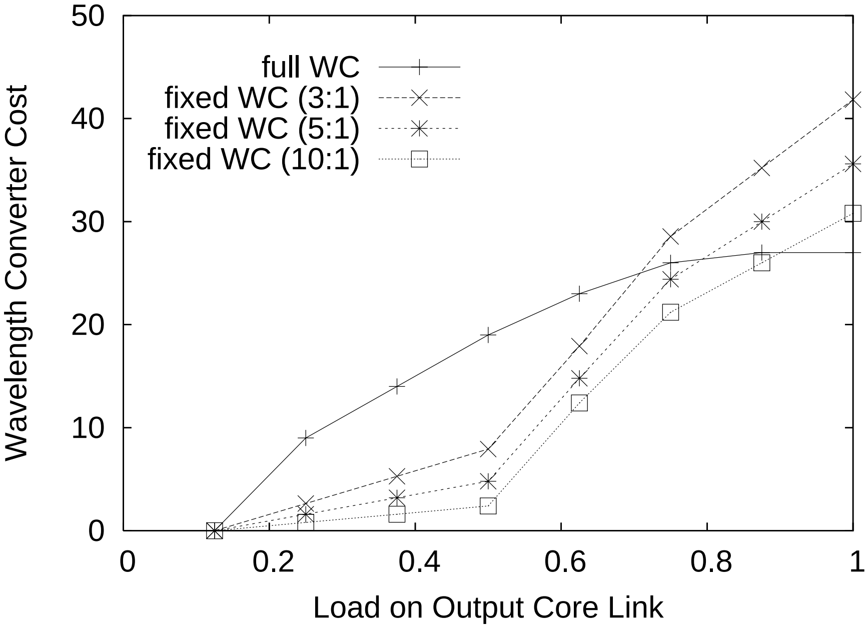


Figure 4.13: Wavelength converter cost ($a = 4$, $W_c = 32$)

Chapter 5

Design of Node-Clusters for Scalable Wavelength Routing

In this chapter, we propose a node-clustering method for hierarchical routing in large-scale wavelength-routed networks. Hierarchical routing scales well by yielding enormous reductions in routing table length, but it can also increase blocking probability because longer paths in hierarchical routing tend to have less free wavelength channels. However, if the routes assigned to longer paths have greater wavelength resources, we can expect that the blocking probability will not increase. Therefore we propose a distributed node-clustering method that maximizes the number of lightpaths between nodes. The key idea behind our method is to construct node-clusters that have much greater wavelength resources from the ingress border nodes to the egress border nodes, which increases the wavelength resources on the routes of lightpaths. We evaluate the blocking probability for lightpath requests and the maximum table length in simulation experiments. We find that the method we propose significantly reduces the table length, while the blocking probability is almost the same as, or even lower than that without clustering.

5.1 Scalability Problem in Routing Protocol for Wavelength Routing

The configuration for lightpaths consists of a route selection phase and a wavelength reservation phase. Route information in the route selection phase is collected via routing protocols such as OSPF [64] or BGP [65]. Then, reservation protocols such as RSVP-TE [66] reserve wavelength resources along the route.

Many researchers have investigated the routing and wavelength reservation protocols for establishing lightpaths in intra-domain networks. Routing and wavelength reservation protocols that target for the inter-domain network have recently been investigated [5, 6, 4, 34]. Bernstein *et al.* [5] specified key requirements for inter-domain routing protocols for optical networks. One of these is the “independence of the internal domain control plane mechanism”. Routing and wavelength reservation protocols in the inter-domain network are independent of protocols in the intra-domain network. BGP is the only existing protocol that conforms to these requirements and is widely deployed in the current Internet. We can use a BGP that is extended to wavelength-routed networks (e.g., Optical BGP [4]) as the inter-domain routing and wavelength reservation protocol.

Li *et al.* [35] pointed out that BGP lacks scalability of number of routes, which results from the increased number of nodes. This is because the BGP router’s memory size limits the routing table size and therefore BGP will not work with a large number of routes. One promising approach to keeping the routing table size scalable is to introduce *hierarchical routing* [14]. The basic idea behind hierarchical routing is to form a set of nodes into a *cluster* to aggregate route information about nodes far from a source node. Each node has complete route information about nodes in the same cluster (i.e., intra-cluster route) and also has aggregated route information about nodes in the other clusters (i.e., inter-cluster route). Therefore, the routing table size is reduced.

Although hierarchical routing reduces the size of the routing table, it generally increases the path length. The main reason is that inter-cluster routes cannot always be the same routes as those in a non-clustered environment. That is, path length is increased when an inter-cluster route with a minimum cluster-hop count differs from the shortest path with a minimum node-hop count (Fig. 1.5). This increased path length is likely to increase the blocking probability for lightpath requests because the probability of finding wavelengths idle on the path decreases as the path length increases. Therefore, it is important to construct clusters to minimize the blocking probability.

In this chapter, we propose a method of clustering in a distributed manner to minimize the blocking probability for lightpath requests. To achieve this, we maximize the number of lightpaths between nodes. The key idea behind our method is to construct the node-clusters that have many wavelength resources from ingress border nodes to egress border nodes, which increases wavelength resources on the routes of lightpaths. We expect the increased number of available lightpaths would lead to decreased blocking probability. Our method is a distributed clustering algorithm that is suited to large-scale wavelength-routed networks.

This chapter is organized as follows. Section 5.2 discusses hierarchical routing, node clustering and the conventional clustering problem. In Sec. 5.3, we propose a distributed method of clustering for wavelength-routed networks. Section 5.4 presents evaluation results obtained by simulation. Finally, we present our conclusions in Sec. 5.5.

Figure 5.1 outlines our network model. The network itself consists of nodes and links that correspond to a domain or an Autonomous System (AS) and a set of optical fibers. Note that each node has its own network (i.e., intra-domain wavelength-routed network) but since we focus on the inter-domain wavelength-routed network, the intra-domain lightpath network is represented as a single node. The numbers attached to the links represent the number of fibers on the link in Fig. 5.1.

When a lightpath is requested, the inter-domain control plane on the gateway of the domain first determines the set of links that the lightpath will traverse (we call the set of links the route) using the route information advertised by the routing protocol, and then reserves wavelength resources along the route using the wavelength reservation protocol. We use a path-vector routing protocol like the BGP for the routing protocol since it meets the requirements of the inter-domain routing protocol in the optical networks [5].

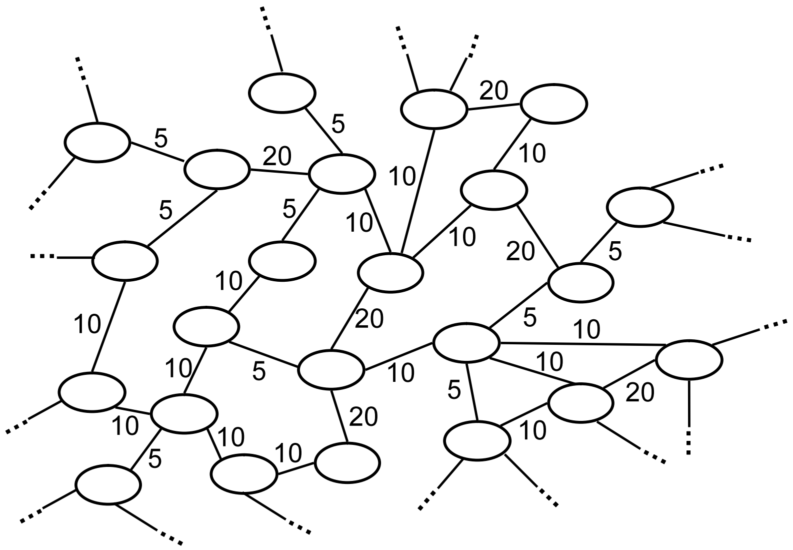


Figure 5.1: Inter-domain wavelength-routed network

5.2 Improvement of Scalability with Hierarchical Routing

5.2.1 Hierarchical Node-Clustering

Figure 5.2 shows an example of hierarchical clustering. We call a set of nodes a cluster. A node whose adjacent node belongs to another cluster is referred to as a *border node*. A level-x cluster consists of level-$(x-1)$ clusters. The minimum level hierarchy is 1-level clustering, where a level-1 cluster includes all nodes. If the level of clustering is more than 1, this is called multi-level clustering or a multi-level hierarchy.

The maximum cluster size is limited to keep the intra-cluster routing table size within a reasonable size. The inter-cluster routing table size can be huge when there are too many clusters. When this happens, the level of clustering is increased and higher-level clusters are constructed to reduce the size of lower-level inter-cluster tables. Although our approach can be extended to a multi-level hierarchy, we only deal with 2-level hierarchical clustering to simplify explanation.

5.2.2 Conventional Clustering Problem

Krishnan *et al.* [67] formulated an optimal clustering problem for communications networks. They treated the problem as a graph partitioning problem and called it the *bounded, connected, min-cut* problem. The objective function of the problem is to minimize the sum of the link cost between clusters.

Bounded, connected, min-cut problem

Given:

- An undirected graph $G = (V, E)$ with edge weights $w : E \to Z_0^+$
- Upper bound on size of clusters $B \in \{1, \ldots, |V|\}$

The optimal clustering is to obtain the set of clusters V_1, V_2, . . . , V_k, such that

$$minimize \sum_e w(e) \tag{5.1}$$

where $e \in E$, $e \notin E_i$, $i \in \{1, 2, \ldots, k\}$ $\forall k \in \{2, \ldots, |V|\}$.

Constraints:

- Graph $G_i = (V_i, E_i)$ that represents the intra-cluster-network of cluster V_i is *connected*
- $1 \leq |V_i| \leq B$, $\forall i \in \{1, 2, \ldots, k\}$

There are two characteristics the clustering problem has in communication networks. First, the clusters need to satisfy *bounded*, *connected* conditions. A *bounded* cluster means the maximum cluster size is bounded by B to keep the intra-cluster routing table within a reasonable size. A *connected* cluster means any two nodes that belong to the same cluster can only reach one another via nodes in that cluster. If the connected condition is not satisfied, two nodes in the same cluster communicate through external clusters. This defeats the purpose of clustering, which is to minimize the storage and exchange of information about external clusters. The second characteristic is that each cluster does not need to be balanced. This is because the construction of balanced clusters does not always result in minimized link costs between clusters.

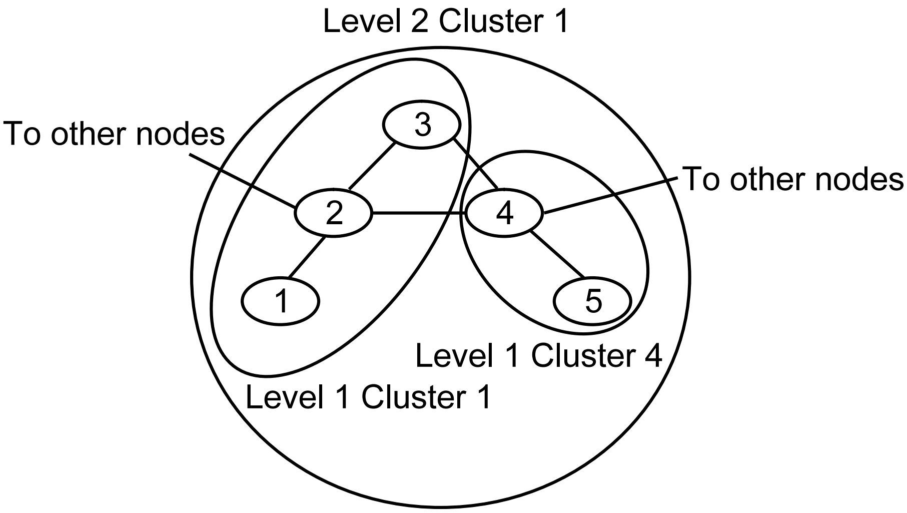


Figure 5.2: Example of hierarchical clustering

Krishnan *et al.* [67] proposed a centralized heuristic algorithm to solve the *bounded, connected, min-cut* problem. The heuristic algorithm consists of three steps: (1) generating initial connected clusters, (2) refining clusters by trading nodes, and (3) refining clusters by merging clusters.

The connected clusters in the initial step are generated through recursive bisection. Since the recursive bisection splits clusters, the heuristic algorithm requires the complete information about the entire network topology. This may cause other scalability problems with the memory having to include complete topological information. We therefore propose a clustering algorithm that is implemented in distributed fashion. Our clustering problem and algorithm will be explained in the next section.

5.3 Node-Clustering for Hierarchical Routing in Wavelength-Routed Networks

5.3.1 Distributed Clustering Algorithm for Hierarchical Routing

As we discussed in Section 5.1, clustering may increase the path length. This increase is a serious problem in wavelength-routed networks because the wavelength assigned to a lightpath must be identical along the route (i.e., wavelength continuity constraint). The increased path length generally leads to increased blocking probability for lightpath requests. The routes for lightpaths in hierarchical routing depend on how the clusters are constructed. It is therefore important to construct clusters to minimize the blocking probability for lightpaths.

In this section, we discuss our development of a distributed clustering algorithm that is suited to large-scale wavelength-routed networks. The requirements for this clustering algorithm are as follows.

1. Keeping the size of routing tables for intra/inter-cluster routing within a certain value
2. Minimizing blocking probability for lightpath requests
3. Constructing clusters in the network with a huge number of nodes

We will explain how these requirements are satisfied with our distributed algorithm after introducing our clustering problem.

To minimize blocking probability in lightpaths, we increase the number of lightpaths available between nodes in wavelength-routed networks. To maximize the number of lightpaths, we first formulate a new clustering problem in wavelength-routed networks that maximizes the number of lightpaths available between nodes. We refer to this problem as the *bounded, connected, max-lightpath* problem. We then propose a distributed clustering algorithm that resolves the *bounded, connected, max-lightpath* problem and satisfies the three requirements.

Bounded, connected, max-lightpath problem

Given:

- $G = (V, E)$ that corresponds to a wavelength-routed network
- Upper bound on size of clusters $B \in \{1, \ldots, |V|\}$

Objective function:

$$maximize \sum_{s=1}^{k} \sum_{i,j \in V_s} F_{ij} + \sum_{s=1}^{k} \sum_{i \in V_s, l \notin V_s} F_{il}, \tag{5.2}$$

where V_1, V_2, ..., V_k are constructed clusters. F_{ij} is the number of lightpaths available on the shortest path from node i to node j, where $F_{ii} = 0, (\forall i = 1, \ldots, N)$.

Constraints:

- Graph $G_i = (V_i, E_i)$ that means the intra-network of cluster V_i is *connected*
- $1 \leq |V_i| \leq B, \forall i \in \{1, 2, \ldots, k\}$

Let us try to maximize the number of lightpaths available between nodes with the above formulation. The number of lightpaths available between nodes consists of (1) those between nodes in the same cluster and (2) those between nodes in different clusters. The latter changes according to the construction of clusters because route with minimum cluster-hop count, which changes depending on the construction of the clusters, is selected as the route of a lightpath between nodes in different clusters. This route selection follows BGP, where route with minimum AS-hop is selected. We use node-hop/cluster-hop counts as a metric for intra/inter-cluster route selection. When there are several routes with the same hop counts, we select the route where the minimum number of fibers on links is largest.

The complexity of our *bounded, connected, max-lightpath* problem is open. The complexity of *bounded, connected, min-cut* problem is also open but the related problems such as the bounded, min-k cut problem, where we need to find a subset of edges such that removing them from the graph results in dividing the graph into k subgraphs and the sum of the edge costs in the subset is minimized, are NP-complete [67]. Krishnan *et al.* therefore proposed a heuristic algorithm for the problem. In this chapter, we also propose a heuristic algorithm, which satisfies the first and second requirements of a clustering algorithm for large-scale wavelength-routed networks. Our method satisfies the first requirement of “keeping the size of routing tables for intra/inter-cluster routing within a certain value” because the constructed clusters are *bounded* and *connected*. *Bounded* condition limits the number of routes maintained in routing tables. *Connected* condition prevents a node from maintaining intra-cluster routes in different clusters. Our method also satisfies the second requirement of “minimizing the blocking probability for lightpath requests” because it maximizes the number of lightpaths available between nodes. In Sec. 5.4, we discuss how maximizing available lightpaths results in decreasing the blocking probability for lightpath requests.

For our proposed method to satisfy the third requirement of "constructing clusters in the network with a huge number of nodes", clusters need to be constructed in a distributed fashion. This is because each border node does not need to maintain all the topological information with our method. After we present information maintained by nodes with our method in Sec. 5.3.1, we will explain our algorithm in Sec. 5.3.1.

Figure 5.3 depicts what information a node and a border node have. All nodes have (1) a *node-to-cluster mapping table* and (2) an *intra-cluster routing table*. In addition, all border nodes have (3) an *inter-cluster routing table*. We will next present the information in each table and when each piece of information is used.

1. Node-to-cluster mapping table:

   This table includes node identifiers and cluster identifiers that include the nodes. We use the minimum node identifier in a cluster as the cluster identifier.

   - When clusters are constructed:

     Each node refers to this table to obtain its cluster identifier, and to find out whether or not it is a border node. Each node can find this out by comparing its cluster identifier with its adjacent nodes' cluster identifiers.

   - When lightpaths are set up:

     Each node refers to this table to obtain the cluster identifier for the destination node.

2. Intra-cluster routing table:

   This table includes the shortest route from a source node to nodes in the same cluster and the minimum number of fibers on links along the route. In the intra-cluster route information to node 2 in Fig. 5.3, "$1, 2$" is a list of nodes on the route and "$F : 5$" means the minimum number of fibers along the route, which is 5.

   - When clusters are constructed:

     Each border node refers to this table to find out the number of fibers available from it to other border nodes in the same cluster.

- When lightpaths are set up:

  Each node refers to this table to find out the route to nodes in the same cluster.

3. Inter-cluster routing table:

   This table includes a list of clusters on routes from the source cluster to other clusters and ingress/egress border nodes for each cluster in the list, and the minimum number of fibers on links along the route. In the inter-cluster route information for cluster 7 in Fig. 5.3, "$(1, 1, 1), (11, 9, 10), (7, 7, -)$" is a list of clusters on the route. Each cluster is expressed as $(ingress\ border\ node\ identifier, cluster\ identifier,$ $egress\ border\ node\ identifier)$. "$F\ :\ 5$" means the minimum number of fibers along the route, which is 5.

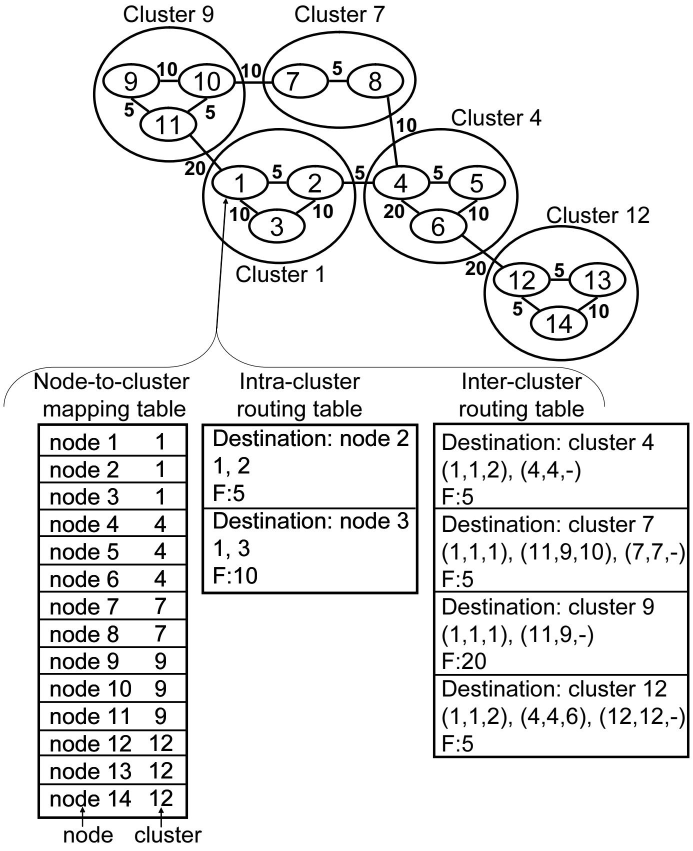
   - When lightpaths are set up:

     Each border node refers to this table to obtain the route to the destination cluster that includes the destination node.

The inter-cluster routing table includes the ingress/egress border nodes for each cluster. This is because we distinguish the routes that pass through the same clusters but pass through different ingress/egress border nodes. We need to distinguish them because the number of fibers available on a route depends on the ingress/egress border nodes in addition to the clusters a lightpath traverses. Note that a node and/or a border node has only one route information for each destination node/cluster because maintaining multiple routes for a destination leads to increasing routing table size. How to realize a diverse routing, which provides multiple paths that do not share the same nodes or links for increasing reliability, in BGP-based inter-domain routing protocol for optical networks is an important problem as described in [5]. However, this problem is beyond the scope of this chapter.

Our algorithm constructs clusters by repeating a *merge* operation. The merge operation makes a cluster merge with an adjacent cluster.

Each cluster performs merge operation with an adjacent cluster so that Eq. (5.2) is maximized. The first term in Eq. (5.2), which means the number of lightpaths whose source and destination belong to the same cluster, is constant despite the construction of the clusters. This is because the routes for those lightpaths are always routes with a minimum node-hop



Node-to-cluster mapping table

node 1	1
node 2	1
node 3	1
node 4	4
node 5	4
node 6	4
node 7	7
node 8	7
node 9	9
node 10	9
node 11	9
node 12	12
node 13	12
node 14	12

node cluster

Intra-cluster routing table

Destination: node 2 1, 2 F:5
Destination: node 3 1, 3 F:10

Inter-cluster routing table

Destination: cluster 4 (1,1,2), (4,4,-) F:5
Destination: cluster 7 (1,1,1), (11,9,10), (7,7,-) F:5
Destination: cluster 9 (1,1,1), (11,9,-) F:20
Destination: cluster 12 (1,1,2), (4,4,6), (12,12,-) F:5

Figure 5.3: Tables maintained by nodes

count. The second term in Eq. (5.2), on the other hand, which means the lightpaths whose source and destination belong to different clusters, changes according to the construction of the clusters because their routes have a minimum cluster-hop count. Consequently, it is important to increase F_{il} in the second term.

In order to maximize F_{il}, it is important to prevent lightpaths that traverse several clusters from being routed on links with few fibers. If the links with few fiber are located between clusters, those links do not tend to be selected as routes for lightpaths. This is because there are multiple links between the clusters and the link with the most fibers is selected as the route among them. Thus, we try to locate links with more fibers in clusters, and to locate links with few fibers between clusters. To achieve this, we use BI (Blocking Island) paradigm [68]. BI provides an efficient way of abstracting resource (e.g., bandwidth) available in a network. BI is a cluster constructed according to the bandwidth availability. β-BI means a cluster in which links composing intra-cluster routes for node-pairs inside have β or more bandwidth.

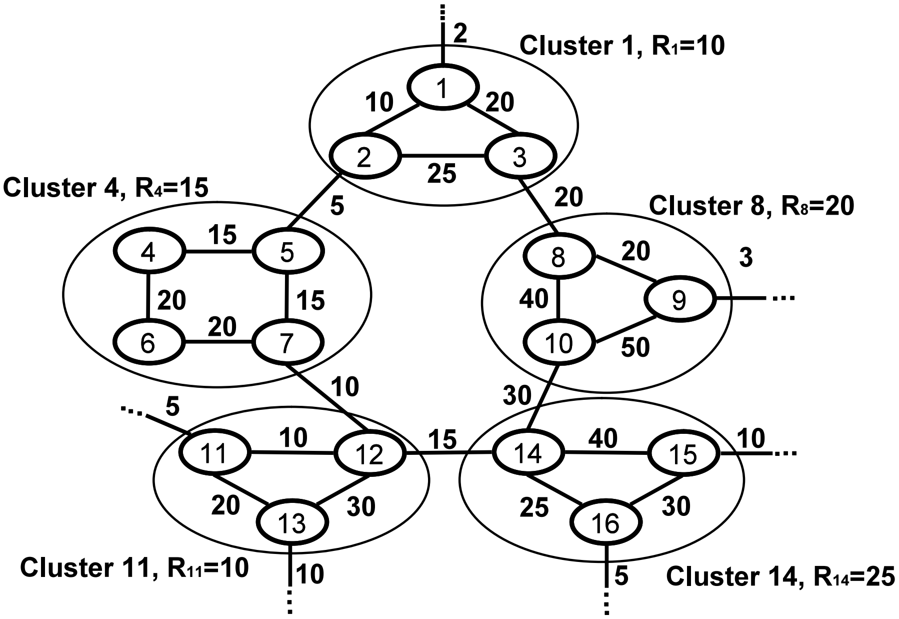


Figure 5.4: Before merge operation

Our algorithm constructs β-BIs by repeating merge operation. There are two differences between the original BI and ours. First, the size of a BI (i.e., a cluster) is bounded in our clustering problem. To maximize the bandwidth from an ingress to an egress border node in a BI, each BI should consist of links with more bandwidth. We realize this by making each cluster give higher priority in taking links with more bandwidth in. Second, we need to bound the maximum node-hop count from an ingress to an egress border node in each BI. This is to prevent the blocking probability from increasing because of increased node-hop count of a lightpath.

The following lists symbols we use in our proposed algorithm.

B : Upper bound for number of nodes that each cluster includes.

β : Lower bound for the number of fibers on links that are taken in clusters.

H : Upper bound for the node-hop counts from an ingress to an egress border nodes in each cluster.

T_w : Waiting time for merge requests to arrive. Each cluster does a merge operation that is requested within T_w.

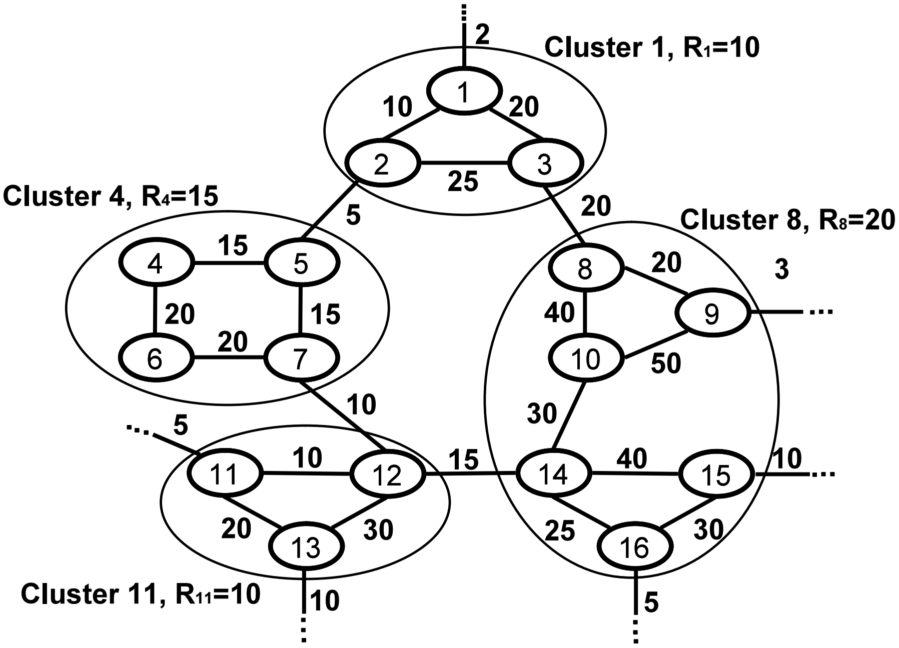


Figure 5.5: Example of merge operation between V_{14} and V_8

R_s : Minimum number of lightpaths available between border nodes in cluster V_s.

R_{st} : Minimum number of lightpaths available on links between cluster V_s and V_t.

$V_{s\cup t}$: Cluster into which cluster V_s merges cluster V_t.

Now, we will present our algorithm, where each cluster V_i individually performs a merge operation. When a hierarchy is not introduced (i.e., no cluster is constructed), each node is regarded as a cluster. When a node is added to the network, the node is regarded as a cluster.

Step 1: Border nodes in V_i set T_w and wait for merge requests from adjacent clusters. Go to Step 2 in time T_w.

Step 2: The border nodes in V_i exchange a received merge request among them. If one or more merge requests arrive, then go to Step 3. Otherwise, go to Step 5.

Step 3: The border nodes in V_i select V_t that sent a merge request with the maximum effect among clusters that sent a merge request to V_i. If there exist more than one candidates for V_t, the cluster having the greatest cluster ID is selected as V_t

[1]. The effect of a merge operation is calculated as $\min(R_i, R_{it}, R_t)$, which is included in a request message. P_i, which is the border node that received the merge request from V_t, sends an *accept merge request* message to V_t. Border nodes that received a merge request from adjacent clusters except V_t send a *refuse merge request* message to the senders of merge requests. Go to Step. 4.

Step 4: P_i informs all nodes in V_i of accepting a merge request. All nodes update (1) node-cluster matching information (change the cluster ID of nodes in $\max(V_i, V_t)$ to $\min(V_i, V_t)$), (2) intra-cluster route information, (3) border node information (whether each node is a border node or not), and (4) $R_{i\cup t}$. Then, border nodes advertise new node-cluster matching information to other clusters. Go back to Step 1.

Step 5: Among adjacent clusters, select $V_{t'}$ such that $\min(R_i, R_{it'}, R_{t'})$ is maximized while satisfying (1) the size of $V_{i\cup t'}$ is B or less, (2) $\min(R_i, R_{it'}, R_{t'}) \geq \beta$, and (3) the maximum node-hop count of intra-route from an ingress to egress node in $V_{i\cup t'}$ is H or less. If there exist more than one candidates for $V_{t'}$, the cluster having the greatest cluster ID is selected as $V_{t'}$. The above selection is done by exchanging information among border nodes in V_i. A border node that is adjacent to $V_{t'}$ and whose node ID is maximum is selected as P_i, which requests a merge operation. If there exists P_i, P_i sends a *merge request* message to $V_{t'}$ and go to Step 6. Otherwise, go to Step 7.

Step 6: If P_i receives an *accept merge request* from $V_{t'}$, P_i informs all nodes in V_i of succeeding in merge request. All nodes update (1) node-cluster matching information (change the cluster ID of nodes in $\max(V_i, V_{t'})$ to $\min(V_i, V_{t'})$), (2) intra-cluster route information, (3) border node information (whether each node is a border node or not), and (4) $R_{i\cup t'}$. Then, border nodes advertise new node-cluster matching information to other clusters. Go back to Step 1. Otherwise (P_i receives a *refuse merge request*), P_i informs all nodes in V_i of failing in merge request and go to Step 1.

[1] The smallest cluster ID is an alternative tie-break condition. We examined it by computer simulation, but the resulted performance was almost the same.

Step 7: Border nodes in V_i advertise new inter-cluster route information. Then, finish this algorithm because there are no adjacent clusters that V_i can perform merge operation with.

In Step 3 and 5, when there exist more than one candidates for V_t ($V_{t'}$), we use the greatest cluster ID as a tie-break condition. This is because border nodes in the same cluster can uniquely determine V_t ($V_{t'}$). If other selection policies (e.g., random) are adopted, border nodes need to exchange additional information to negotiate which cluster each border node selects.

In trying to perform a merge operation, border nodes in V_i approximately calculate $R_{i \cup t}$ as $\min(R_i, R_{it}, R_t)$. Let us now explain why $R_{i \cup t}$ is $\min(R_i, R_{it}, R_t)$. The border node pair where the number of available lightpaths is minimum belongs to (1) V_i, (2) V_t, or (3) both V_i and V_t. In (1) and (2), the minimum number of lightpaths corresponds to R_i and R_t, respectively. In (3), the route between a border node in V_i and one in V_t consists of the route between border nodes in V_i, the link between V_i and V_t, and the route between border nodes in V_t. Thus, the minimum number of lightpaths on these routes and the link, that is, $\min(R_i, R_{it}, R_t)$, corresponds to $R_{i \cup t}$. Note that $R_{i \cup t}$ does not always equal $\min(R_i, R_{it}, R_t)$ because all links between V_i and V_t are not always part of the routes between border nodes in cluster $V_{i \cup t}$. However, V_i do not calculate $R_{i \cup t}$ precisely because this calculation needs hop counts for all the routes between all the border node pairs, which degrades the scalability of our clustering method.

Figures 5.4 and 5.5 have samples of a merge operation. We set the number of wavelengths multiplexed on fibers to one for the sake of simplicity. When cluster 14 merges with cluster 11 in Fig. 5.4, the minimum number of lightpaths available between border nodes, $R_{14 \cup 11}$ is equal to $\min(R_{14}, R_{14,11}, R_{11}) = \min(25, 15, 10) = 10$. When cluster 14 merges with cluster 8, $R_{14 \cup 8} = 20$. Since $R_{14 \cup 8} > R_{14 \cup 11}$, cluster 14 sends a merge request to cluster 8. Figure 5.5 depicts the construction of clusters after cluster 14 merges with cluster 8. The route from cluster 11 to cluster 1 changes from $12 \rightarrow 7 \rightarrow 5 \rightarrow 2$ to $12 \rightarrow 14 \rightarrow 10 \rightarrow 8 \rightarrow 3$. If there are some candidate routes with the same cluster-hop counts, we select a route where the number of available lightpaths is maximum. Note that the number of lightpaths available on the route changes from 5 to 15.

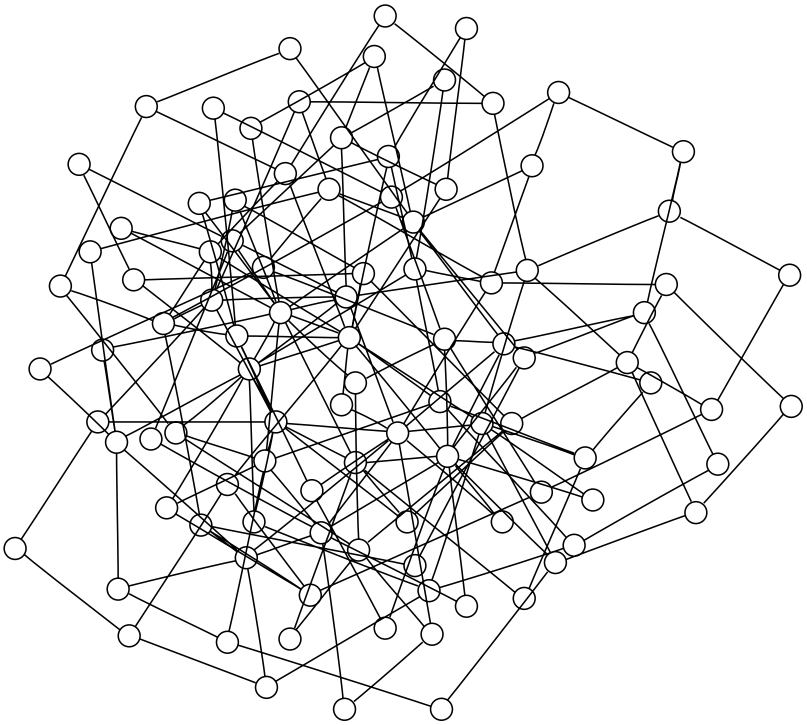


Figure 5.6: Random network ($N = 100$)

5.4 Numerical Evaluation and Discussions

5.4.1 Simulation Condition

We used random networks with 100, 200, 300, 400, and 500 nodes generated by the Waxman algorithm [69] whose parameters α and β were 0.15 and 0.2, respectively. Fig. 5.6 shows the resulting random network with 100 nodes. We assume that there is no propagation delay on each link and no processing delay on each node. Note, however, that even if propagation delay and processing delay are considered, the resulted clustering is identical as long as the time for a pair of clusters to complete a merge operation is smaller than T_w. The number of fibers on link uniformly ranged from 1 to 30. There were 32 wavelengths multiplexed on a fiber.

We compared our distributed clustering method applied to the *bounded, connected, max-lightpath* problem (BI) with (1) a network without any clusters (no cluster) and (2) a distributed clustering method applied to the *bounded, connected, min-cut* problem (min-cut). With min-cut, we set the link cost to 1. In this case, each cluster merges an adjacent cluster that has the maximum number of connected links, which leads to maximizing the

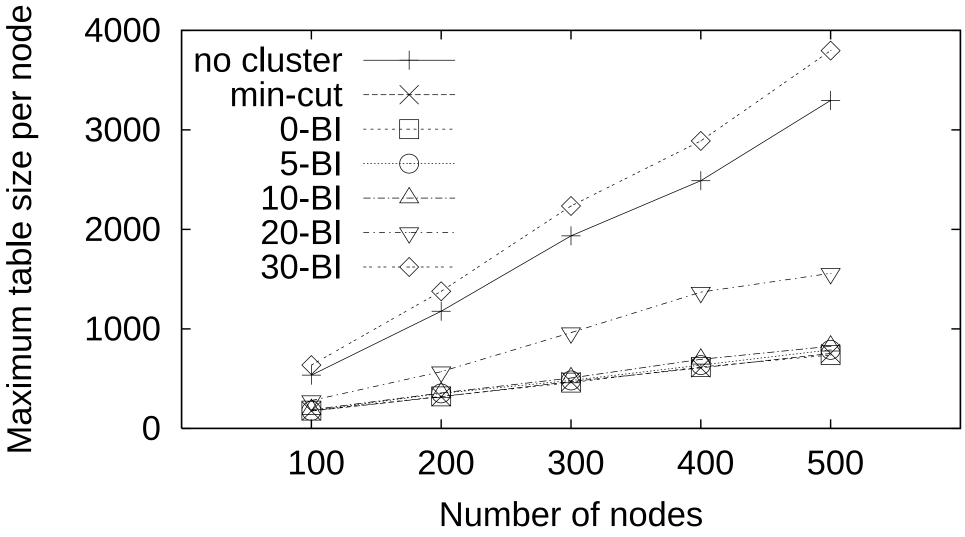


Figure 5.7: Maximum table size maintained by node

Table 5.1: Average number of clusters constructed

no cluster	min-cut	0-BI	5-BI	10-BI	20-BI	30-BI
100	11.5	12.6	14.5	19.4	43.5	100

number of links in merged clusters (i.e., minimizing the number of links between clusters).

With each clustering method, we set B to $\sqrt{N}$ (N was the number of nodes in the network) because setting B to $\sqrt[M]{N}$ in a network with M layers leads to minimized table length [14] and $M = 2$. We set H to $\sqrt{N}$ (upper bound on H) because small H may not lead to increasing the number of lightpaths available between nodes. The waiting time for a merge request, T_w, was set to $\gamma \times T$. γ was a uniform random variable from 1 to 4 and $T = 10(s)$, which was large enough for a merged cluster to update each piece of information in the cluster.

5.4.2 Maximum Table Size

Figure 5.7 shows the maximum table size maintained by a node in the networks with different numbers of nodes. In networks without clusters, each node only maintains a routing table that has a set of routes to all nodes. In clustered networks constructed with BI and min-cut, on the other hand, each node maintains a node-cluster mapping table and an intra-cluster routing table (see Sec. 5.3.1). In addition, each border node maintains an

Table 5.2: Average number of lightpaths available between nodes

no cluster	min-cut	0-BI	5-BI	10-BI	20-BI	30-BI
309.9	243.2	334.5	353.2	358.2	353.6	309.9

Table 5.3: Maximum load on channel

no cluster	min-cut	0-BI	5-BI	10-BI	20-BI	30-BI
2.55	7.41	4.14	1.70	1.91	2.27	2.55

inter-cluster routing table. We defined the table size as the total hop count of routes for intra/inter-cluster routing tables and as the total number of entries for a node-cluster mapping table. In our BI, β is set to 0, 5, 10, 20, and 30. 30-BI does not perform merge operation because there exists no link that has more than 30 fibers.

0-BI and min-cut show the smaller table size than others because merge operation is not limited by the constraint about β in those methods. The table sizes in 0-BI and min-cut are about between 22% and 33% of that without clusters. This is because 0-BI and min-cut reduce the number of routes by aggregating routes to nodes in the same cluster. As the number of nodes increases, the effect of aggregation increases.

0-BI yields almost the same table size as min-cut does because the numbers of clusters and nodes included by each cluster with both methods are similar. 30-BI needs more memory than that without clusters. This is because 30-BI has node-cluster mapping table in addition to inter-cluster routing table that is same as the routing table in the network without clusters.

As for BI, the table size increases as β gets larger. This is because larger β limits the number of merge operations performed in the network. As a result, less routes are aggregated. Table 5.1 shows the average number of clusters constructed with each method in the network with 100 nodes. As more merge operations are performed, the number of clusters constructed gets close to the optimal value ($\sqrt{N} = 10$). When β is relatively small ($\beta = 5$), the table size can be reduced close to the minimum size since most merge operation are not limited by constraint as to β.

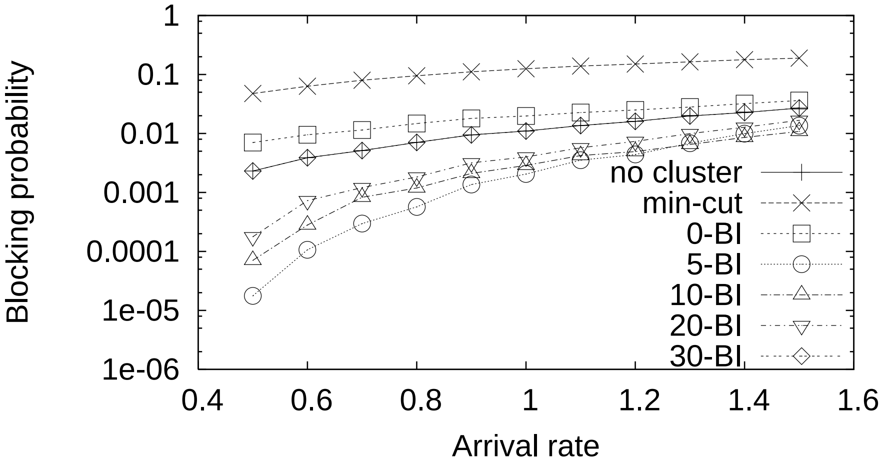


Figure 5.8: Blocking probability for lightpath requests (holding time: 60s)

Table 5.4: Average number of node-hop counts of lightpaths

no cluster	min-cut	0-BI	$5 - BI$	10-BI	20-BI	30-BI
3.33	4.74	4.63	4.60	4.51	4.18	3.33

5.4.3 Blocking Probability for Lightpath Requests

We next evaluate the blocking probability for lightpath requests. Lightpath requests arrive after the clusters are constructed. The requests arrive according to a Poisson process at a rate of λ (requests/s) and the holding time for lightpaths follows an exponential distribution with an average of 60 seconds. From here, we use a random network with 100 nodes. The results are shown in Fig. 5.8. The horizontal axis represents the arrival rate of lightpath requests and the vertical axis represents the blocking probability for lightpath requests.

In Fig. 5.8, the results by BIs outperform the results by min-cut for all arrival rates. This is because more wavelength resources are provided for each node-pair in BIs. Comparing BIs with different β, 5-BI shows the lowest blocking probability among them. Before we explain why 5-BI shows good performance, we show the average number of lightpaths available between nodes in Tab. 5.2, the maximum load on link in Tab. 5.3, and the average number of node-hop counts of lightpaths in Tab. 5.4. Here, we define the load on channel as the ratio of the number of node-pairs that traverses the link to the number of wavelengths on the link. From these tables, we observe that more lightpaths available between nodes make the blocking probability lower while the average hop-count increases by constructing clusters. However, this is not enough. Requests of lightpaths through heavy-load link tends to be rejected, which makes the overall blocking probability increases. Therefore, minimizing the maximum load is also important for decreasing the blocking probability.

Now we explain why 5-BI shows the lowest blocking probability among other algorithms. The reject of lightpath request tends to occur on links with few fibers. To decrease the blocking probability, the number of node-pairs that traverse those links must be minimized. 5-BI realizes this by 1) locating links less than five fibers between clusters, and 2) constructing clusters whose sizes are near to B. As the size of cluster gets larger, the cluster tends to have more links between adjacent clusters. If there are several links between clusters, the link with more fibers is selected as an inter-cluster route. The other links with few fiber are not selected as an inter-cluster route. In 30-BI, the size of each cluster is one and each cluster has only one link between an adjacent cluster. The sizes of clusters in 10-BI and 20-BI are smaller than that in 5-BI. As a result, 10-BI and 20-BI show higher blocking probability than 5-BI does. In 0-BI, each cluster can include links with few fibers, which leads to higher blocking probability.

We conclude that 5-BI provides better performance in terms of blocking probability than others while keeping the routing table size almost the same as 0-BI and min-cut.

We further evaluate our 5-BI-based clustering method when a new node is added to network. In this case, the reconstruction of clusters is needed. To realize this, we introduce a *give* operation, in which a cluster gives one of its border nodes to an adjacent cluster. A give operation is performed when a cluster cannot perform a merge operation. Cluster V_i gives its border node to adjacent cluster V_t if all the following six conditions are satisfied: (1) the size of V_t is $B-1$ or less, (2) the size of V_i is more than 1, (3) the maximum node-hop count of intra-route from an ingress to egress node in V_t is $H-1$ or less, (4) R_i increases, (5) R_{it} decreases, and (6) V_i remains connected. V_i selects a cluster (say V_t) among adjacent clusters such that the increase in R_i is maximized. It is better to increase both R_i and R_t. However, V_i cannot know the increase in R_t before the give operation because detail of intra-cluster route information of R_j is not available.

We compare three kinds of clustering methods: (1) BI*-scratch* where new clusters are constructed from scratch when a new node is added, (2)BI*-incremental merge* where the

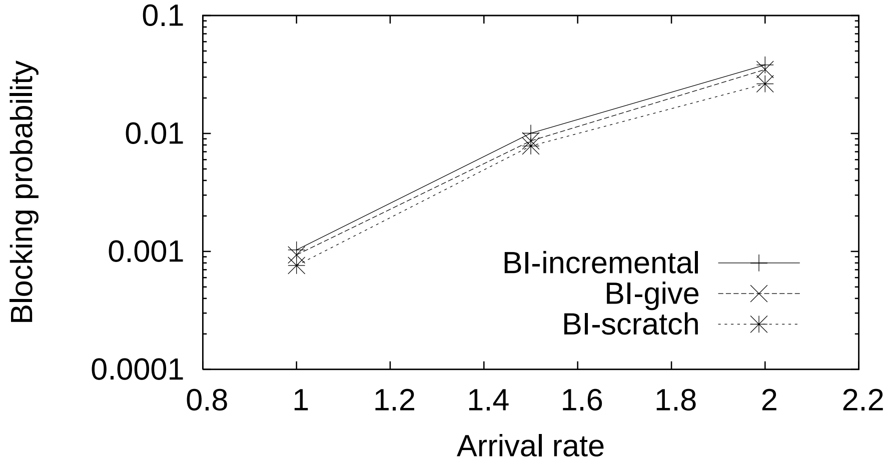


Figure 5.9: Blocking probability for lightpath requests (21 nodes are added)

existing clusters and a new cluster (a new node) try to perform only the merge operation, and (3) BI*-give* where the existing clusters and a new cluster try to perform both merge and give operations.

Figure 5.9 shows the blocking probability when 21 nodes are added one by one to a network with 100 nodes. BI-incremental and BI-give achieve almost the same blocking probability as BI-scratch in spite that BI-incremental and BI-give performs much smaller number of operations than BI-scratch.

Figure 5.10 shows the blocking probability when 44 nodes are added one by one. When more nodes are added, BI-give shows lower blocking probability than BI-incremental. This is because give operation increases the number of wavelengths available in clusters and releases links with few fibers out of cluster.

However, BI-give does not achieve as low blocking probability as BI-scratch does when 44 nodes are added. This means that the number of added nodes that give operation can cope with is limited. When give operation is not effective, we need to reconstruct clusters from scratch.

5.5 Conclusion

We proposed a distributed node-clustering method for hierarchical routing in wavelength-routed networks. The method based on Blocking Island paradigm maximizes the number of

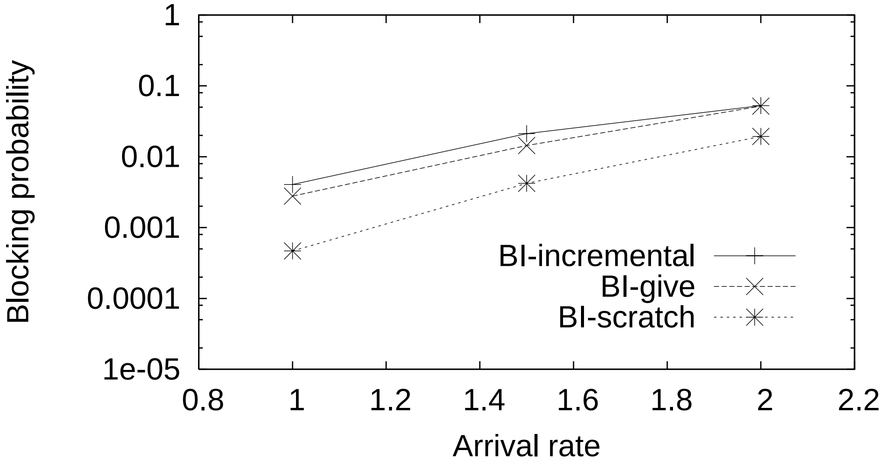


Figure 5.10: Blocking probability for lightpath requests (44 nodes are added)

lightpaths between nodes. Throughout our simulation, we found that the table size with our BI with appropriate β ranged between 22% and 33% of that in a cluster-less network. The effect of aggregating the route information increased as the number of nodes increased. In terms of the blocking probability for lightpath requests in a network with 100 nodes, we found that locating links with fewer fibers between clusters was important in addition to increasing the number of lightpath in cluster for decreasing blocking probability. We further evaluated a method to restructure clusters (give operation) when new nodes are added to a network. We found that our give operation is effective until a certain number of nodes are added.

Chapter 6

Conclusion

In this thesis, we have proposed methods for flexible and scalable wavelength-routed networks. Flexibility is indispensable in the real world where traffic patterns drastically change. Scalability in terms of the number of nodes and wavelengths multiplexed is also essential because the realization of inter-domain lightpath provisioning drives the increase in the number of nodes and wavelengths multiplexed in wavelength-routed networks.

In Chapter 2, we have proposed a novel design method of WDM network that is flexible against traffic changes. Through the simulation, we evaluated how cost-effectively we use the network equipment by comparing the network that our proposed method designs with those that the conventional methods design, both of which need almost the same OXC cost. As a result, we have shown the network that our proposed method designs achieves lower ratio of blocked lightpaths than the one obtained by the over-provisioning approach does. We conclude that our proposed method designs a flexible WDM network in the cost-effective way.

In Chapter 3, we have proposed e-MLDA (extended MLDA), a new heuristic algorithm for the design of logical topologies to be overlaid on WDM networks. The resulting topology is based on the actual levels of node-to-node traffic demand. We went on to propose MALDA (Minimum number of fiber Amplifiers Logical topology Design Algorithm) for which the objective function is to minimize the number of fiber amplifiers deployed in the logical topology. Our algorithms are evaluated by comparing them with the conventional method in terms of average delay, throughput, and number of optical fiber amplifiers deployed in the network. The results have shown that MALDA only needs about one-fifth of the fiber amplifiers that e-MLDA does, while MALDA is able to accommodate as much traffic as e-MLDA. Furthermore, when the processing capacity of IP routers is high, MALDA can accommodate more traffic than e-MLDA does. Our results indicate that MALDA is preferable in terms of designing a low-cost logical topology.

In Chapter 4, we investigated the deployment of wavelength converters in wavelength-routed networks with overlay model. We showed that, in wavelength-routed networks with overlay model, most wavelength converters are deployed on edge nodes for covering the difference in the numbers of wavelengths multiplexed on access and core links by simulation. We then proposed an ingress edge node architecture with fixed wavelength converters to reduce the number of full wavelength converters and wavelength converter cost on an ingress edge node. In simulation, our node architecture achieved an objective blocking performance with lower wavelength converter cost than a node architecture that only uses full wavelength converters. When the load on the output core link is in the situation where networks are under operation and wavelength converter cost ratio is 3, our node architecture offered about 21 % cost reduction compared with a node architecture that only uses full wavelength converters. When load is lower, our node architecture offered more than 46 % cost reduction. In addition, fixed wavelength conversion offers more cost reduction as the wavelength converter cost ratio gets larger. Utilizing fixed converters leads to cost reduction regardless of the difference in the numbers of wavelengths multiplexed on access and core links.

We proposed a distributed node-clustering method for hierarchical routing in wavelength-routed networks in Chapter 5. The method based on Blocking Island paradigm maximizes the number of lightpaths between nodes. Throughout our simulation, we found that the table size with our BI with appropriate β ranged between 22% and 33% of that in a clusterless network. The effect of aggregating the route information increased as the number of nodes increased. In terms of the blocking probability for lightpath requests in a network with 100 nodes, we found that locating links with fewer fibers between clusters was important in addition to increasing the number of lightpath in cluster for decreasing blocking probability. We further evaluated a method to restructure clusters (give operation) when new nodes are added to a network. We found that our give operation is effective until a certain number of nodes are added.

The flexibility against traffic change and the scalability for the increase in the number of nodes, the number of wavelengths multiplexed are key metrics in design of wavelength-routed networks. We believe the discussions in this thesis contribute to realize large-scale and flexible wavelength-routed networks.

Bibliography

[1] K. G. Coffman and A. M. Odlyzko, “Internet growth: Is there a ”Moore’s Law” for data traffic?,” `http://www.research.att.com/~amo/doc/internet.moore.pdf`.

[2] “Flashwave 7700.” `http://www.fujitsu.com/global/services/telecom/flashwave-7700.html`.

[3] I. Chlamtac, A.Ganz, and G.Karmi, “Lightpath communications: An approach to high bandwidth optical WAN’s,” *IEEE Transactions on Communications*, vol. 40, pp. 1171–1182, July 1992.

[4] M. J. Francisco, S. Simpson, L. Pezoulas, C. Huang, J. Lambadaris, and B. St. Arnaud, “Interdomain routing in optical networks,” in *Proceedings of Opticomm2001*, pp. 120–129, Aug. 2001.

[5] G. Bernstein, B. Rajagopalan, D. Pendarakis, A. Chiu, J. Strand, V. Sharma, D. Cheng, R. Izmailov, L. Ong, and S. Dharanikota, “Optical inter domain routing considerations,” *Internet Draft* draft-ietf-ipo-optical-inter-domain-02.txt, Feb. 2003.

[6] D. Wang, J. Strand, J. Yates, C. Kalmanek, G. Li, and A. Greenberg, “OSPF for routing information exchange across metro/core optical networks,” *Optical Networks Magazine*, vol. 3, pp. 34–43, Sept. 2002.

[7] R. Dutta and G. N. Rouskas, “A survey of virtual topology design algorithms for wavelength routed optical networks,” *Optical Networks Magazine*, vol. 1, pp. 73–89, Jan. 2000.

[8] R. Ramaswami and K. N. Sivarajan, “Design of logical topologies for wavelength-routed optical networks,” *IEEE Journal on Selected Areas in Communications*, vol. 14, pp. 840–851, June 1996.

[9] B. Mukherjee, D. Banerjee, S. Ramamurthy, and A. Mukherjee, “Some principles for designing a wide-area WDM optical networks,” *IEEE/ACM Transactions on Networking*, vol. 4, pp. 684–696, Oct. 1996.

[10] S. Baroni and P. Bayvel, “Wavelength requirements in arbitrarily connected wavelength-routed optical networks,” *IEEE/OSA Journal of Lightwave Technology*, vol. 15, pp. 242–251, Feb. 1997.

[11] N. Nagatsu, S. Okamoto, and K. Sato, “Optical path cross-connect system scale evaluation using path accommodation design for restricted wavelength multiplexing,” *IEEE Jounal on Selected Areas in Communications*, vol. 14, pp. 893–902, June 1996.

[12] Y. Miyao and H. Saito, “Optimal design and evaluation of survivable WDM transport networks,” *IEEE Jounal on Selected Areas in Communications*, vol. 16, pp. 1190–1198, Sept. 1998.

[13] B. V. Caenegem, W. V. Parys, F. D. Turck, and P. M. Demeester, “Dimensioning of survivable WDM networks,” *IEEE Journal on Selected Areas in Communications*, vol. 16, pp. 1146–1157, Sept. 1998.

[14] L. Kleinrock and F. Kamoun, “Hierarchical routing for large networks,” *Computer Networks*, vol. 1, pp. 155–174, Jan. 1977.

[15] Y. Fukushima, H. Harai, S. Arakawa, and M. Murata, “On the robustness of planning methods for traffic changes in WDM networks,” *OSA Journal of Optical Networking*, vol. 4, pp. 11–25, Jan. 2005.

[16] Y. Fukushima, H. Harai, S. Arakawa, and M. Murata, “Planning method of robust WDM networks against traffic changes,” in *Proceedings of Optical Network Design and Modeling 2004 (ONDM2004)*, pp. 695–714, Feb. 2004.

[17] Y. Fukushima, H. Harai, S. Arakawa, M. Murata, and H. Miyahara, "Planning and design methods for WDM networks robust against traffic changes," *Technical Report of IEICE,* (PS2003-3) *(in Japanese),* pp. 11–16, Apr. 2003.

[18] Y. Fukushima, H. Harai, S. Arakawa, M. Murata, and H. Miyahara, "A minimum interference routing algorithm for multi-period planning of WDM lightpath networks without traffic prediction," in *Proceedings of 28th European Conference on Optical Communication (ECOC 2002),* Sept. 2002. P4.9.

[19] Y. Fukushima, H. Harai, S. Arakawa, M. Murata, and H. Miyahara, "An enhanced minimum interference routing algorithm for multi-period planning of WDM lightpath networks without traffic prediction," *Technical Report of IEICE,* (IN2002-34) *(in Japanese)*, pp. 7–12, July 2002.

[20] A. A. Kuehn and M. J. Hamburger, "A heuristic program for locating warehouses," *Management Science,* vol. 9, pp. 643–666, July 1963.

[21] M. Kodialam and T. V. Lakshman, "Minimum interference routing with applications to MPLS traffic engineering," in *Proceedings of IEEE INFOCOM 2000*, pp. 884–893, Mar. 2000.

[22] M. Kodialam and T. V. Lakshman, "Integrated dynamic IP and wavelength routing in IP over WDM networks," in *Proceedings of IEEE INFOCOM 2001*, pp. 358–366, May 2001.

[23] Y. Fukushima, S. Arakawa, and M. Murata, "Design of logical topology with effective waveband usage in IP over WDM networks," to appear in *Photonic Network Communications,* Aug. 2006.

[24] Y. Fukushima, S. Arakawa, M. Murata, and H. Miyahara, "A design method for logical topologies with consideration of wavebands," in *Proceedings of Optical Network Design and Modeling 2002 (ONDM2002)*, Feb. 2002.

[25] Y. Fukushima, S. Arakawa, M. Murata, and H. Miyahara, "Design of logical topologies in consideration of available wavebands," *Technical Report of IEICE,* (NS2001-67) *(in Japanese)*, pp. 33–38, July 2001.

[26] Y. Fukushima, H. Harai, S. Arakawa, and M. Murata, “Design of wavelength-convertible edge nodes in wavelength-routed networks,” to appear in *OSA Journal of Optical Networking*, 2006.

[27] Y. Fukushima, H. Harai, S. Arakawa, and M. Murata, “Deployment of wavelength converters in wavelength-routed overlay networks,” *Technical Report of IEICE,* (IN2005-90) *(in Japanese)*, pp. 13–18, Oct. 2005.

[28] H. Harai, M. Murata, and H. Miyahara, “Heuristic algorithm for allocation of wavelength convertible nodes and routing coodination in all-optical networks,” *IEEE/OSA Journal of Lightwave Technology*, vol. 17, pp. 535–545, Apr. 1999.

[29] S. Subramaniam, M. Azizoglu, and A. K. Somani, “On optimal converter placement in wavelength-routed networks,” *IEEE/ACM Transactions on Networking*, vol. 7, pp. 754–766, Oct. 1999.

[30] X. Chu, J. Liu, and Z. Zhang, “Analysis of sparse-partial wavelength conversion in wavelength-routed WDM networks,” in *Proceedings of IEEE INFOCOM 2004*, pp. 1363–1371, Mar. 2004.

[31] Y. Fukushima, H. Harai, S. Arakawa, and M. Murata, “A distributed clustering method for hierarchical routing in large-scaled wavelength routed networks,” *IEICE Transactions on Communications*, vol. E88-B, pp. 3904–3913, Oct. 2005.

[32] Y. Fukushima, H. Harai, S. Arakawa, and M. Murata, “Distributed clustering method for large-scaled wavelength routed networks,” in *Proceedings of 2005 IEEE Workshop on High Performance Switching and Routing (HPSR)*, May 2005.

[33] Y. Fukushima, H. Harai, S. Arakawa, and M. Murata, “Node clustering method for hierarchical routing in WDM lightpath networks,” *Technical Report of IEICE,* (NS2004-172) *(in Japanese)*, pp. 1–6, Dec. 2004.

[34] C. Pelsser and O. Bonaventure, “Extending RSVP-TE to support inter-AS LSPs,” in *Proceedings of 2003 Workshop on High Performance Switching and Routing (HPSR 2003)*, pp. 79–84, June 2003.

[35] W. Li, “Inter-domain routing: Problems and solutions,” *Technical Report TR-128, Department of Computer Science, State University of New York*, Feb. 2003.

[36] X. Cao, V. Anand, Y. Xiong, and C. Qiao, “Performance evaluation of wavelength band switching in multi-fiber all-optical networks,” in *Proceedings of IEEE INFOCOM 2003*, pp. 2251–2261, Apr. 2003.

[37] H. Harai, M. Murata, and H. Miyahara, “Performance analysis of wavelength assignment policies in all-optical networks with limited-range wavelength conversion,” *IEEE Journal on Selected Areas in Communications*, vol. 16, pp. 1051–1060, Sept. 1998.

[38] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin, “Network Flows: Theory, Algorithms, and Applications,” *Prentice Hall*, 1993.

[39] A. Dacomo, S. D. Patre, G. Maier, A. Pattavina, and M. Martinelli, “Design of static resilient WDM mesh networks with multiple heuristic criteria,” in *Proceedings of IEEE INFOCOM 2002*, pp. 1793–1802, June 2002.

[40] M. Murata, “Challenges for the next-generation Internet and the role of IP over photonic networks,” *IEICE Transaction on Communications*, vol. E83-B, pp. 2153–2165, Oct. 2000.

[41] K. Kitayama and M. Murata, “A 1,000-channel WDM network can resolve network bottleneck,” in *Proceedings of the 7th Asia-Pacific Conference on Communications (APCC 2001)*, pp. 113–116, Sept. 2001.

[42] H. Yokota, H. Kanamori, and Y. Ishiguro, “Ultra-low-loss pure-silica-core single-mode fiber and transmission experiment,” in *Proceedings of 1986 Optical Fiber Conference (OFC 8́6)*, p. PD3, 1986.

[43] X. Chu, B. Li, and Z. Zhang, “A dynamic RWA algorithm in a wavelength-routed all-optical network with wavelength converters,” in *Proceedings of IEEE INFOCOM 2003*, pp. 1795–1804, Mar. 2003.

[44] G. N. Rouskas and M. H. Ammar, “Dynamic reconfiguration in multihop WDM networks,” *Journal of High Speed Networks*, vol. 4, pp. 221–238, June 1995.

[45] S. Ishida, S. Arakawa, and M. Murata, “Reconfiguration of logical topologies with minimum traffic disruptions in reliable WDM-based mesh networks,” *Photonic Network Communications*, vol. 6, pp. 265–277, Nov. 2003.

[46] Xi Yang and Byrav Ramamurthy, “Interdomain dynamic wavelength routing in the next-generation translucent optical internet,” *Journal of Optical Networking*, vol. 3, pp. 169–187, Mar. 2004.

[47] “NTT Information Web Station.” `http://www.ntt-east.co.jp/info-st/network/traffic/index.html` (in Japanese).

[48] “Avici systems.” `http://www.avici.com`.

[49] “OptIPuter.” `http://www.optiputer.net/`.

[50] “CA*net4.” `http://www.canarie.ca/canet4/`.

[51] H. Harai and M. Murata, “Establishing lightpaths of an optical ring for distributed computing environment,” in *Proceedings of IEEE/Create-Net GRIDNETS 2005*, pp. 488–495, Oct. 2005.

[52] H. Nakamoto, K. Baba, and M. Murata, “Proposal of a shared memory access method for lambda computing emvironment,” in *Proceedings of IFIP Optical Networks and Technologies Conference (OpNeTec)*, pp. 210–217, Oct. 2004.

[53] B. Rajagopalan, D. Pendarakis, D. Saha, R. S. Ramamoorthy, and K. Bala, “IP over optical networks: Architectural aspects,” *IEEE Communications Magazine*, vol. 38, pp. 94–102, Sept. 2000.

[54] I. Foster and C. Kesselman, *The grid: blueprint for a new computing infrastructure*. Morgan Kaufmann Publishers, 1998.

[55] E. Oki, D. Shimazaki, K. Shiomoto, N. Matsuura, and W. Imajuku, “Performance of distributed-controlled dynamic wavelength-conversion GMPLS networks,” in *Proceedings of ICOCN 2002*, Nov. 2002.

[56] S. Yoo, “Wavelength conversion technologies for WDM network applications,” *IEEE/OSA Journal of Lightwave Technology*, vol. 14, pp. 955–966, June 1996.

[57] J. H. Lee, T. Nagashima, T. Hasegawa, S. Ohara, N. Sugimoto, T. Tanemura, and K. Kikuchi, “Wavelength conversion of 40-Gbit/s NRZ signal using four-wave mixing in 40-cm-long bismuth oxide based highly-nonlinear optical fiber,” in *Proceedings of 2005 Optical Fiber Conference (OFC 0́5)*, Mar. 2005. PD6.

[58] K. Onohara, Y. Awaji, N. Wada, F. Kubota, and K. Kitayama, “Agile and highly efficient wavelength conversion using highly nonlinear fiber for optical code-labeled packets,” *IEEE Photonics Technology Letters*, vol. 17, pp. 627–629, Mar. 2005.

[59] Y. Awaji and H. Harai, “NICT internal study (personal communication),” June 2004.

[60] `http://www.furukawa.co.jp/jiho/fj109/fj109_15.pdf`, Jan. 2002.

[61] J. Yamawaku, E. Yamazaki, A. Takada, and T. Morioka, “Field trial of virtual-grouped-wavelength-path switching with QPM-LN waveband converter and PLC matrix switch in JGN II test bed,” *IEE Electronics Letters*, vol. 41, pp. 88–89, Jan. 2005.

[62] K. Xi, S. Arakawa, and M. Murata, “How many wavelength converters do we need?,” in *Proceedings of Optical Network Design and Modeling 2005 (ONDM2005)*, pp. 347–358, Feb. 2005.

[63] “Series E: Overall network operation, telephone service, service operation and human factors; quality of service, network management and traffic engineering – traffic engineering – ISDN traffic engineering,” *ITU-T Recommendation E.721*, May 1999.

[64] K. Kompella and Y. Rekhter, “OSPF extensions in support of generalized MPLS,” *Internet Draft* draft-ietf-ccamp-ospf-gmpls-extensions-12.txt, Oct. 2003.

[65] Y. Rekhter and T. Li, “A Border Gateway Protocol 4 (BGP-4),” *IETF RFC 1771*, Mar. 1995.

[66] L. Berger, “Generalized multi-protocol label switching (GMPLS) signaling resource reservation protocol-traffic engineering (RSVP-TE) extensions,” *IETF RFC 3473*, Jan. 2003.

[67] R. Krishnan, R. Ramanathan, and M. Steenstrup, “Optimization algorithms for large self-structuring networks,” in *Proceedings of IEEE INFOCOM’99*, pp. 71–78, Mar. 1999.

[68] D. Zhemin and M. Hamdi, “Resource management in multi-segment optical networks using the blocking island paradigm,” in *Proceedings of 2003 Workshop on High Performance Switching and Routing (HPSR 2003)*, pp. 43–48, June 2003.

[69] B. M. Waxman, “Routing of multipoint connections,” *IEEE Journal on Selected Areas in Communications*, vol. 6, pp. 1617–1622, Dec. 1988.