A Study on Flexible, Reliable, and Scalable Wavelength-Routed Optical Networks

Shinya Ishida

Department of Information Networking

Graduate School of Information Science and Technology

Osaka University

February 2007

List of Publications

Journal Papers

1. 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, no. 3, pp. 265–277, Nov. 2003.

2. S. Ishida, S. Arakawa, and M. Murata, "Virtual fiber configuration for dynamic lightpath establishment in large-scaled optical networks," *Photonic Network Communications*, vol. 12, no. 1, pp. 87–98, July 2006.

3. S. Ishida, S. Arakawa, and M. Murata, "Analyses of soft-state signaling protocols in GMPLS-based WDM networks," submitted to *IEEE/OSA Journal of Lightwave Technology*, July 2006.

Refereed Conference Papers

1. S. Ishida, S. Arakawa, and M. Murata, "Proposal of procedures to reconfigure logical topologies in reliable WDM-based mesh networks," in *Proceedings of SPIE Asia-Pacific Optical and Wireless Communications (APOC 2002) Optical Networking II*, vol. 4910, (Shanghai, China), pp. 115–125, Oct. 2002.

2. S. Ishida, S. Arakawa, and M. Murata, "Dynamic reconfiguration of logical topologies in WDM-based mesh networks," in *Proceedings of the 7th IFIP Working Conference on Optical Network Design and Modelling (ONDM2003)*, vol. 1, (Budapest, Hungary), pp. 93–112, Feb. 2003.

3. S. Ishida, S. Arakawa, and M. Murata, "Virtual fiber configuration method for dynamic lightpath establishment in large-scaled WDM networks," in *Proceedings of 9th Conference on Optical Network Design and Modelling (ONDM 2005)*, (Milan, Italy), pp. 153–161, Feb. 2005.

4. S. Ishida, S. Arakawa, and M. Murata, "Performance analysis of soft-state lightpath management in GMPLS-based WDM networks," in *Proceedings of Third International Conference on Broadband Communications, Networks, and Systems (Broadnets 2006)*, (San José, CA), Oct. 2006.

5. S. Ishida, S. Arakawa, and M. Murata, "Local recovery from massive failures in large-scaled WDM networks," submitted to *11th Conference on Optical Network Design and Modelling (ONDM 2007)*, May 2007.

Non-Refereed Technical Papers

1. S. Ishida, S. Arakawa, and M. Murata, "An algorithm to reconfigure logical topologiees in reliable WDM networks," *Technical Report of IEICE (PS2002-1)*, pp. 49–54, Apr. 2002. (*in Japanese*).

2. A. Shin'ichi, S. Ishida, and M. Murata, "Management of logical topologies for dynamically changing traffic in reliable IP over WDM networks," *Technical Report of IEICE (DC2002-2)*, pp. 7–14, Apr. 2002. (*in Japanese*).

3. S. Ishida, S. Arakawa, and M. Murata, "On a power-law relationship in wavelength-routed networks," *Technical Report of IEICE (PN2003-27)*, pp. 13–16, Dec. 2003. (*in Japanese*).

4. S. Ishida, S. Arakawa, and M. Murata, "Quasi-static lightpath configuration method in large-scaled WDM networks," *Technical Report of IEICE (CS2004-8)*, pp. 37–42, May 2004.

5. S. Ishida, S. Arakawa, and M. Murata, "Performance analysis of soft-state signaling protocols in wavelength-routed networks," *Technical Report of IEICE (PN2005-46)*,

pp. 1–6, Dec. 2005. (*in Japanese*).

Preface

The volume of the Internet traffic has been increasing rapidly due to the growth of the population of Internet users and the popularization of online applications and services that require high bandwidth, such as voice chat, video streaming, P2P file sharing, and grid computing. To accommodate the Internet traffic, the capacity of backbone networks has been enhanced by WDM (Wavelength Division Multiplexing). WDM is the technology that multiplexes and carries signals of different optical wavelengths in a single optical fiber. Although WDM resolves the shortage of the fiber capacity, the nodes connected to the WDM-capable fiber become the bottlenecks because those nodes must convert received signals between the optical format and the electric format and process the signals at the speed of light. Therefore, WDM-based networks are extended to wavelength-routed networks by installing optical switches in the nodes. Wavelength-routed network is based on the circuit-switching paradigm; nodes are connected with dedicated virtual circuits called *lightpaths*. By configuring lightpaths, a *logical topology* is constructed over a wavelength-routed network.

A wavelength-routed network consists of *data plane* and *control plane*. Lightpaths are established on the data plane including optical switches and fibers while those lightpaths are managed in the control plane. There are two standard control architectures for wavelength-routed networks, GMPLS (Generalized Multi-Protocol Label Switching) and ASON (Automatically Switched Optical Network). These architectures have been being designed so as to interconnect multiple wavelength-routed networks. The progress of the standardizations of these architectures grows the scale of wavelength-routed networks. As a result, large-scaled wavelength-routed networks are comprised. Due to this enlargement, the volume of the traffic carried over the networks increases. The probability that a network

failure occurs gets higher because of the increase of the number of network components. In addition, the amount of the information for managing networks also increases. Hence, flexibility, reliability, and scalability are serious issues for large-scaled wavelength-routed networks.

In this thesis, we propose a method to reconfigure logical topologies to retain the flexibility of wavelength-routed networks, at first. A lot of logical topology design algorithms have been proposed in previous studies. In these studies, most of those algorithms commonly assume that the traffic demands are known in advance. In spite that, as the network scale gets larger, the volume of traffic grows and the traffic pattern changes. To accommodate the increasing and changing traffic, the current logical topology should be reconfigured to a new optimal logical topology. The new logical topology can be obtained by using any of the logical topology design algorithms, but the traffic carried over the working lighptaths are lost since those lightpaths are torn down to setup new lightpaths. Hence, we develop a reconfiguration method diminishing the traffic loss during reconfiguration. The results of simulations show that our method can reconfigure logical topologies without traffic loss.

We next focus on the property of the physical topologies of large-scaled wavelength-routed networks. The scale of wavelength-routed networks grows by interconnecting a lot of small wavelength-routed networks. The Internet has also been growing in the similar way; by interconnecting ASes (Autonomous Systems). In addition, it is known that the topology of the Internet has power-law property on its degree distribution. In the topology having the power-law property, there are a few nodes that have lots of links (called hub nodes) while most of the other nodes have only a few links. According to the analogy of the growth of the Internet, it is speculated that the topologies of large-scaled wavelength-routed networks also have the power-law property. In such networks, the hub nodes are likely to be the bottlenecks of the resource utilization since a lot of wavelength requests conflict at those nodes. As a result, a number of wavelengths are required to accommodate the traffic although there are available wavelengths at non-hub nodes. Hence, we introduce the concept of *virtual fiber* and propose a wavelength routing method to distribute the load on the hub nodes. We construct logical topologies over physical topologies by configuring virtual fibers. Then we route lightpaths in logical topologies, not in physical topologies. By adopting our method, performances of WDM networks with the power-law connectivity

are improved without any cost for network equipments and link state based routings. We evaluate our method by computer simulations and the results show that our method reduces more than one order of magnitude of blocking probability.

The information about lightpath management is exchanged among nodes using a certain signaling protocol. Signaling protocols are classified into two classes; *soft-state* and *hard-state*. In soft-state signaling, the reservation states at each node are managed with timeout timers and periodic refresh of the reservation states are required to keep them. If the timer is expired, the corresponding reservation state is deleted; that is, the reserved wavelength is released. On the other hand, hard-state protocols manage the reservation states explicitly with control messages. Although the number of control messages of soft-state signaling is greater, this timeout mechanism is significant especially for large-scaled networks. This is because the mechanism guarantees the release of the reserved wavelengths even if release messages cannot be delivered due to message losses or control channel failures. In hard-state signaling, if the message to release a reserved wavelength do not reach the destination node due to message loss or control channel failure, the reservation state becomes orphaned and the wavelength to be released is kept reserved. Soft-state signaling protocols have some control parameters but the effects of tuning those parameters are not understand well. Therefore, we analyze the performance of soft-state signaling using Markov model. We also evaluate the effect of the message retransmission extension for signaling protocols with various parameters. As a result, it is revealed that soft-state signaling with the message retransmission extension would reserve a wavelength uselessly about 10 times as long as soft-state signaling without the extension when there are 1000 sessions at a node.

We finally introduce a local recovery scheme against multiple node failures in a certain region of a network (we call them *massive failures*). Massive failures would be caused by natural disasters, such as earthquakes, or by power-cut. There are a lot of studies on protection of lightpaths against a single node or link failure. However, when it comes to a massive failure, such protection schemes using backup lightpaths are ineffective since the backup lightpaths would also be failed. In addition, it is difficult for remote nodes to know the exact failed place in large-scaled networks. Therefore, the performance of the end-to-end recovery is degraded. In contrast to this, our proposed scheme locally configures a cycle enclosing the part of a massive failure, called *diverting cycle*. Then, disrupted

lightpaths are diverted along the diverting cycle. Our recovery scheme also reduces the amount of control messages since a huge number of control messages are exchanged for failure notification, link-state update, and lightpath recovery after a massive failure. The results of computer simulations show that our scheme recovers the lightpath connectivity to almost 100% more quickly than the path restoration scheme when the scale of massive failures is not large. When the scale of massive failures is large, our scheme reduces the number of control messages to about the half comparing to the path restoration scheme.

Acknowledgments

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, Prof. Teruo Higashino, and Prof. Hirotaka Nakano of Osaka University for reading my dissertation and providing many valuable comments.

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.

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.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

The volume of the Internet traffic has been increasing rapidly [1,2]. The number of Internet users has also been increasing and these users demand more bandwidth for a variety of applications and services, such as WWW, voice chat, video streaming, P2P file sharing, and grid computing. Enterprise users may demand a certain dedicated bandwidth to bind their branches' offices online. The capacity of the Internet is being exhausted and, therefore, the backbone networks have to enhance their capacity to accommodate the huge traffic. WDM (Wavelength Division Multiplexing) is the technology that multiplexes and carries signals of different optical wavelengths in a single optical fiber (Figure 1.1) [3,4]. At this moment, 14Tbps (140 channels $\times$ 111 Gbps/channel) transmission is possible with a single fiber [5].

Optical signals carried via optical fibers have to be converted to electrical signals. Transponders of optical signals convert electric packets and data streams into the optical format and send them into fibers while receivers convert received optical signals into the electric format and pass them to electric routers or switches. It is very hard to convert signals between optical and electric formats at each intermediate node when transponders are multi-hop distant from receivers. To omit O-E/E-O conversion at intermediate nodes and prevent intermediate nodes from being the bottleneck, OXCs (Optical Cross Connects) are installed at each node. OXC is the device to redirect optical signals from

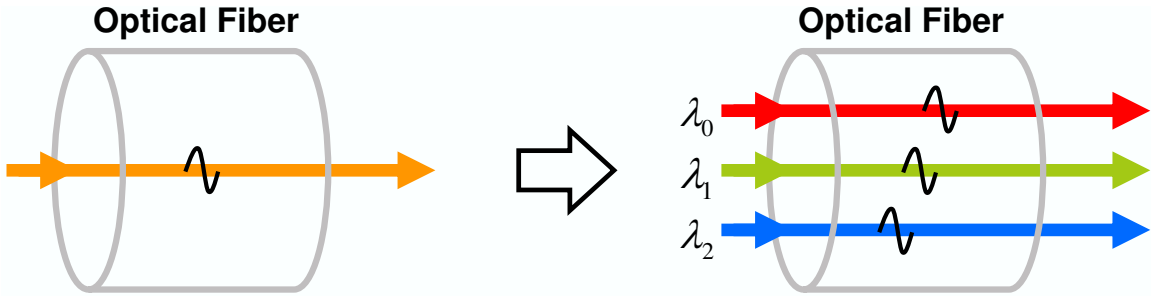


Figure 1.1: Wavelength division multiplexing

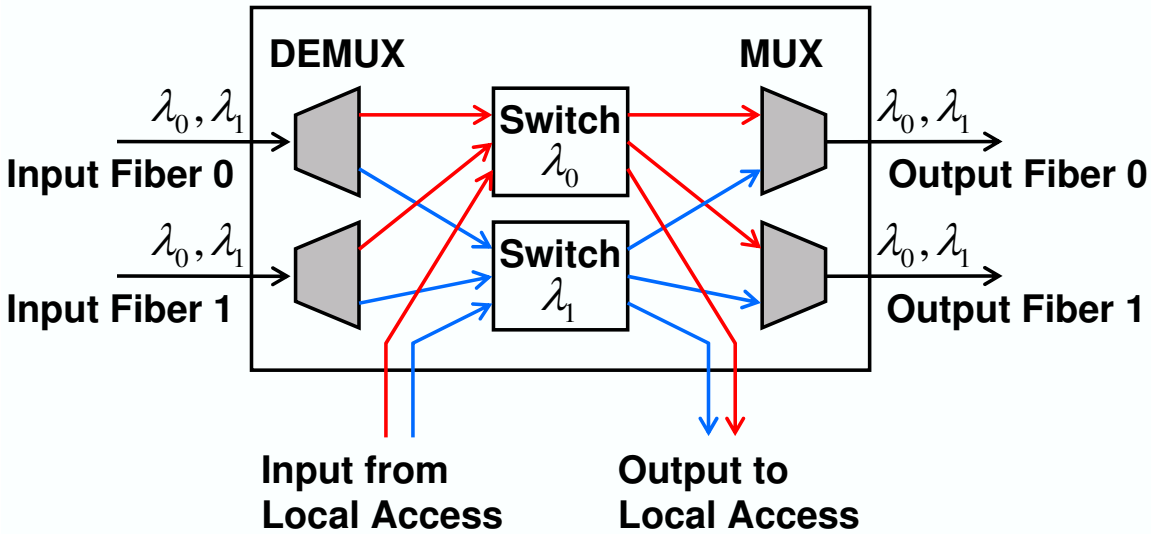


Figure 1.2: Optical cross connect

input fibers to output fibers. Figure 1.2 shows the architecture of OXC whose the degree of the wavelength multiplexing is two. Multiplexed optical signals from input fibers are de-multiplexed into two wavelengths, λ_0 and λ_1. All the input signals of λ_0 are passed into an optical switch. This optical switch redirects the input signals to the output fibers as illustrated in Figure 1.3. By configuring OXCs, the routes of signals are decided. This redirection of wavelength signals by OXCs or optical switches is called *wavelength routing* and optical networks consist of optical fibers and OXCs are called *wavelength-routed networks* [3, 4].

In wavelength-routed networks, virtual circuits are configured for each wavelength by configuring OXCs to transmit data. These virtual circuits are called *lightpaths*. Since there is no electrical processing of packets at the intermediate nodes lightpaths pass through, two edge nodes of a lightpath seem to be directly connected from the upper layer's view.

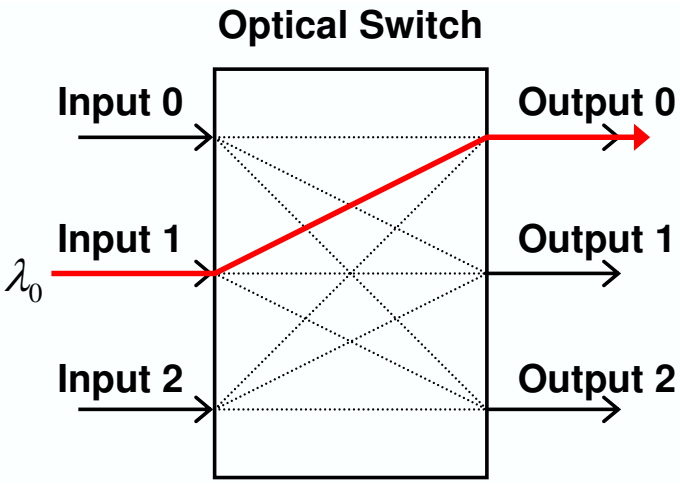


Figure 1.3: Optical switch

Therefore, by configuring lightpaths in a wavelength-routed network, the set of lightpaths makes up a logical topology on the top of the physical topology, and the logical topology is served for the upper layer protocols. Figure 1.4 illustrates a wavelength-routed network consists of six OXCs and seven fibers. Each OXC is connected with an access node R_i ($0 \leq i \leq 5$). The arrows in that figure are lightpaths of wavelengths λ_0 and λ_1. The lightpaths of λ_0 are single hop while the lightpaths of λ_1 pass a few hops. In this situation, the topology of the access nodes' layer is expressed as in Figure 1.5. Node pairs that are not adjacent in the physical topology, (R_0, R_{R3}), (R_0, R_4), and (R_1, R_3) have direct connections whose entities are lightpaths of λ_1.

Where to route lightpaths and which wavelengths to be assigned to those lightpaths are issues so as to utilize wavelength resource efficiently because the number of wavelengths per a fiber is limited (tens, hundreds or at most 1000 wavelengths) and because the assigned wavelengths are dedicated to lightpaths. In addition, there is a restriction for setting up lightpaths that degrades the resource utilization; to setup a lightpath along a path, an identical wavelength must be reserved at each hop of that path. This is known as *the wavelength-continuity constraint* [3,4]. Although wavelength converters that can convert a wavelength to another one resolve this constraint. However, network operators are unwilling to use them since the cost for deploying wavelength converters is expensive. For this reason, it is a quite important issue how to design wavelength-routed networks in order to achieve the high resource utilization.

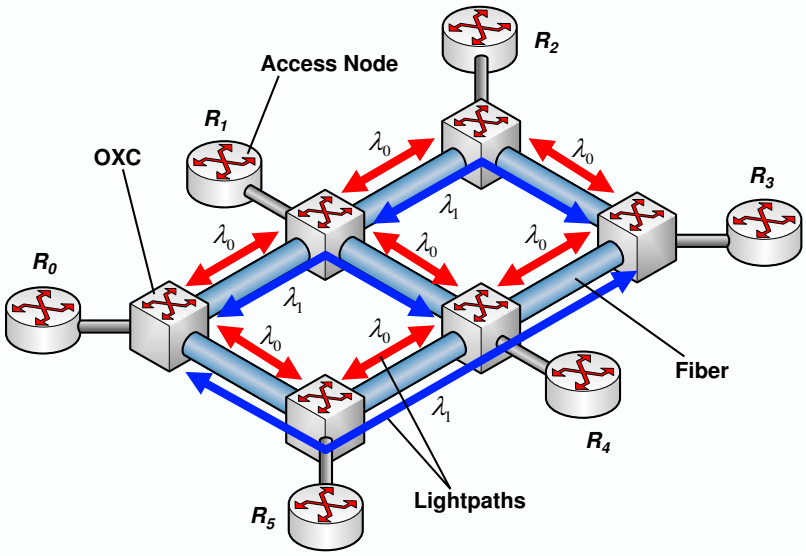


Figure 1.4: Wavelength-routed network

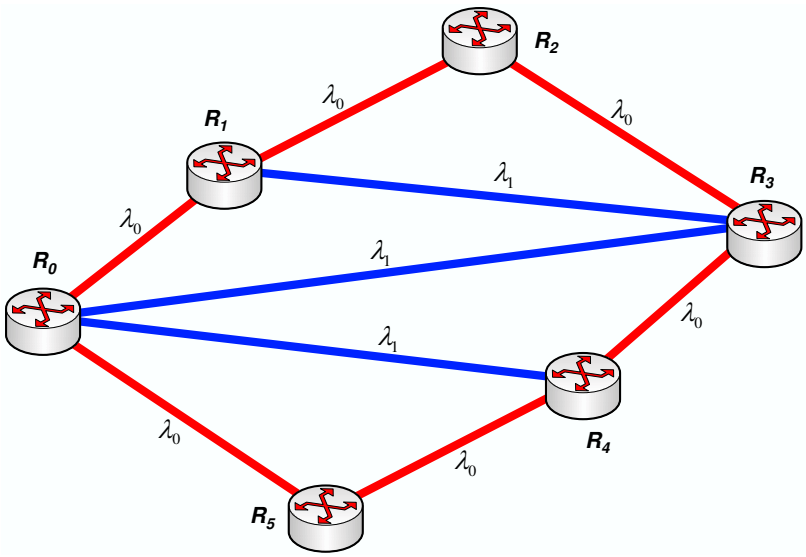


Figure 1.5: Logical topology

Wavelength-routed networks are constructed with two parts; *data plane* and *control plane* [6–8]. The data plane contains optical fibers and OXCs while the control plane is composed of controllers to configure OXCs and control channels interconnecting the controllers. In the control plane, routing for path calculation, signaling for wavelength reservation, and management of network components are done. Currently, CCAMP (Common Control and Measurement Plane) working group IETF (Internet Engineering Task

Force) has been standardizing the specification GMPLS (Generalized Multi-Protocol Label Switching) [6]. GMPLS is an architecture and a protocol suite to control wavelength-routed networks. ITU-T (International Telecommunication Union Telecommunication) has been progressing the standardization of ASON (Automatically Switched Optical Networks) [9, 10]. The both of the architectures currently aim to interconnect optical domain networks and to develop large-scaled wavelength-routed networks. In such large-scaled wavelength-routed networks, remote LANs (Local Area Network), MANs (Metropolitan Area Networks), and WANs (Wide Area Networks) may be connected by lightpaths. Dedicated networks for grid computing are established with some lightpaths. Although such users demand static connections, some users would temporarily setup lightpaths to exchange huge data. Following this trend, we need to consider the design of large-scaled wavelength-routed networks.

1.2 Design Issues for Wavelength-Routed Networks

As the scale of wavelength-routed networks makes becomes larger, these three issues more serious; flexibility against the changing traffic, reliability against the network failure, and scalability against the network size. We survey previous studies and have a discussion for each issue.

1.2.1 Flexibility

Although methods to design logical topologies are investigated in many past studies [11, 12], it has commonly been assumed that the traffic demands are known in advance. However, in practice, it is difficult to predict changes in the traffic demand precisely due to some factors like the start of new services in networks, such as video streaming and contents delivery services. As the scale of networks become larger, the pattern of the traffic demands changes more rapidly and drastically. Therefore, the number of wavelengths required to accommodate the traffic demands would be increased rapidly when the logical topology design is inappropriate for the traffic patterns. Thus, design of flexible networks is a more important method than static network design [13].

Incremental capacity dimensioning is proposed to achieve flexibility and reliability in wavelength-routed mesh networks [14, 15]. In this method, logical topology design is first applied for a given traffic demand. With the incremental traffic demand, a one-by-one assignment of the primary lightpath as well as the backup lightpaths based on the user's perception of performance is done. The one-by-one assignment may result in a topology far from the optimal one. Therefore, we reconfigure the running topology to the optimal (or sub-optimal) logical topology obtained by again applying the design method for the current traffic demand. However, during reconfigurations of logical topologies, working lightpaths are disrupted by releasing wavelengths reserved for them in order to establish new optimal lightpaths. That is, reconfigurations of logical topologies results in the loss of traffic. Hence, to reduce the volume of lost traffic during reconfigurations of logical topologies is an important issue for the flexibility of wavelength-routed networks as well as the reliability.

1.2.2 Reliability

Wavelength-routed networks carry huge amount of data traffic due to their nature. Only a single fiber cut could make the communications over that fiber disrupted and cause tremendous amount of economic damages. Hence, how fast the failed lightpath connections are recovered is an important performance metric for reliable wavelength-routed networks. To assure the reliability of wavelength-routed networks, lots of recovery schemes has been introduced. The schemes of lightpath recovery are classified into two groups; *protection* and *restoration* [16, 17].

Protection schemes reserve extra wavelengths as backup lightpaths for working lightpaths [17, 18]. When working lightpaths are torn down by network failures, the traffics carried over those lightpaths are switched to their backup lightpaths. Therefore, protection schemes assure 100% recovery from failures on the working lightpaths as long as its corresponding backup lightpaths are not failed. However, since the number of wavelengths is limited, protection schemes degrade the utilization of the wavelength resources. It is possible to avoid this problem by sharing backup wavelength resources among working lightpaths [19]. However, this sharing diminishes the efficiency of protection because only

a lightpath is recovered with the shared wavelength at a time.

Restoration schemes reserve wavelengths after a network failure occur and lightpaths are failed [17,20]. Hence, in contrast to protection, restoration schemes allow more flexible use of the wavelengths. However, they cannot guarantee the recovery of working lightpaths at all.

1.2.3 Scalability

The rapid growth in the traffic volume of the Internet has led to demands of higher capacities for backbone networks. Wavelength-routing based on WDM is expected as an approach to satisfy those demands and optical networks employing the technology has been employed to improve the capacity of the cores of Wide Area Networks (WANs) [11, 12, 21–24] and Metropolitan Area Networks (MANs) [25, 26]. These area networks are interconnected by the standardized technologies, such as GMPLS and ASON, and comprise large-scaled wavelength-routed networks. The centralized management of the lightpaths in such large networks is difficult. Hence, the distributed lightpath management should be adopted [27–29].

In the distributed lightpath management, lightpaths are managed by their source nodes. Source nodes of lightpaths send trigger messages to establish the lightpaths when they receive lightpath setup requests. The routes of lightpaths are computed at each source node or decided hop-by-hop. Unlike centralized management, it is not guaranteed that the requested lightpaths are successfully established. If some requests conflict a certain wavelength at a hop, one of them reserves the wavelength, but the other requests are rejected. To improve the resource utilization in distributed lightpath management, it is a significant problem to reduce the probability of this rejection, called *blocking probability*.

There are routing algorithms using link-states in order to calculate more suitable routes for lightpaths [30, 31]. These link-states are advertised to each node periodically. Those routing algorithms do not perform well if the period of the link-state advertisement is long because the link-states held at each node are old and do not correspond to the actual states of the wavelengths [32, 33]. The routes and wavelengths of lighpaths calculated with such

old and inconsistent link-states would be unavailable and the reservations of such lightpaths are blocked. Although frequent update of the link-states accelerates the performance of the routing algorithms, it causes lots of link-state advertisements. As the number of links increases, each node is more heavily loaded for processing link-state advertisements. Hence, efficient and scalable routing schemes are required for large-scaled wavelength-routed networks.

1.3 Outline of Thesis

Following above discussions, it is required to enhance the flexibility, the reliability and the scalability of wavelength-routed networks in order to develop large-scaled wavelength-routed networks.

Dynamic Reconfiguration of Logical Topologies [34–37]

During reconfiguration between two logical topologies, packet loss or delayed arrival may occur due to the deletion of older lightpaths. It may result in a loss of traffic on those lightpaths and decline of the performance of a network. Therefore, there is a trade-off in the reconfiguration between improved network performance obtained by the reconfiguration itself and the traffic loss penalty due to the deletion of lightpaths during the reconfiguration [38]. There are various studies on reconfiguration to minimize lightpath deletion [38–42].

To relieve the influence of tearing down working lightpaths, In Ref. [41], branch-exchange method is proposed. This method tries to minimize traffic loss by reducing the number of steps required in a reconfiguration. Reconfigurations in ring networks are also considered [42]. However, most of the reconfiguration methods proposed in the previous studies are for wavelength-routed networks with star-based or ring-based topologies, not mesh topologies.

In Chapter 2, we propose a reconfiguration algorithm for wavelength-routed mesh networks to provide flexible and reliable backbones. Our basic idea is to use wavelength resources reserved for backup lightpaths which are not always utilized. Our algorithm is

based on five procedures to set up and tear down lightpaths. In addition to simply setting up or tearing down lightpaths, we have considered three other procedures to incorporate wavelength resources for backup lightpaths. Since the backup lightpaths are not always used for transporting the actual traffic, we exploit their wavelength resources assuming that failure does not occur during reconfiguration.

Routing Scheme for Large-Scaled Wavelength-Routed Networks [43–46]

In Chapter 3, we propose a scalable routing scheme for large-scaled wavelength-routed networks. To achieve this objective, we first investigate the structure and the property of the network topology of large-scaled wavelength-routed networks. According to the analogy between the process of the Internet's growth and that of wavelength-routed networks' growth, it is speculated that the physical topologies of large-scaled wavelength-routed networks have the *power-law* connectivity. Briefly, in the networks having the power-law connectivity (we call such networks power-law networks, hereafter), most of the nodes have just a few links although some nodes have a number of links.

The differences of the topological properties between random and power-law networks and evaluate the performances of those two types of topologies. Node degrees in random networks are almost uniform while those in power-law networks are biased as described above. Because of the biased degree distribution, distributions of loads on links in power-law networks are also unbalanced; some links are heavily loaded while most links are lightly loaded. As a result, blocking performance of power-law networks are worse than those of random networks. There are some way to resolve this problem. The simplest solution is enhancement of network equipments; installing or upgrading OXCs and fibers at heavily loaded parts. However, this solution requires too much investment in equipments to resolve the problem by itself. Next solution is using a link state based routing. Such a routing realizes well-balanced load distributions. But, meanwhile, it requires to distribute link state information and to update routing tables at each node frequently so as to perform well. This penalty is undesirable in particular for large-scaled networks.

Then we propose another solution based on *virtual fiber* configuration. We construct

logical topologies over physical topologies by configuring virtual fibers and route lightpaths in logical topologies, not in physical topologies as in the normal way. By adopting our method, performances of WDM networks with the power-law connectivity are improved without any cost for network equipments and link state based routing schemes.

Performance Analysis of Signaling State Managements [47–49]

Wavelength-routed networks are composed of data plane and control plane: Data plane is composed of optical fibers and OXCs while control plane is composed of control channels and controllers for signaling and routing. The control plane exchanges signaling messages and configures states of OXCs, according to a signaling protocol. RSVP-TE (Resource reSerVation Protocol - Traffic Engineering) [50] is a soft-state signaling protocol for GMPLS networks.

In soft-state signaling, each node sets timers for control states and initializes control states when corresponding timers expire. If a node receives a refresh message before a timer expires, it resets the timer and maintains the corresponding state. Since reserved resources are released due to timeout, resource utilization would be worse than that in hard-state control. In addition, soft-state signaling requires more signaling messages than hard-state signaling in order to refresh states. However, nodes managing states in soft-state control initialize states even when signaling messages do not reach them due to network failures. In actual networks, not only message losses but also control plane failures would occur. Therefore, soft-state mechanism is required to achieve tolerant network management.

Many signaling protocols for lightpath establishment in wavelength-routed networks have been proposed: BR (Backward Reservation) [28], FR (Forward Reservation) [28], IIR (Intermediate-Initiated Reservation) [29], and PR (Parallel Reservation) [27]. The main purpose of these works has been to improve blocking performance. These protocols have been evaluated as hard-state signaling protocols since it is supposed that signaling messages are never lost in those performance evaluations. In hard-state signaling, states are managed by explicit signaling messages; that is, nodes continue to reserve unnecessary wavelengths when signaling messages are lost. An infrequent lack of signaling messages

could be dealt with by message retransmission. However, when nodes cannot communicate with each other due to failures of their control planes or for some other reasons, unnecessary wavelengths are not released until the control plane is recovered. Resource utilization thus deteriorates.

In Chapter 4, we evaluate the performance of GMPLS RSVP-TE; we investigate how control parameter settings affect the performance of GMPLS RSVP-TE and when the message retransmission of GMPLS RSVP-TE works effectively. To more precisely understand the influence of each control parameter to the network performance and the relation between control parameter settings, we extend the Markov model in [51] for GMPLS RSVP-TE. Using the Markov model, we describe the behavior of GMPLS RSVP-TE in detail and analyze the steady-state probabilities of a lightpath session. We then investigate the network performance, such as resource utilization and LSP setup delay of GMPLS RSVP-TE.

Local Recovery from Massive Failures [52]

Connectivity is a key issue for transport networks. The importance is quite high for wavelength-routed networks that are placed at backbones and carry huge amount of traffic. As the number of nodes and links increase, the probability of failures rises. Therefore, not only single node or link failures but also multiple node or link failures have to be taken into consideration. Multiple failures occur by independent single failures in various places of networks and by failures in a certain region due to earthquakes or accidental power cut. We call the failures in the latter case *massive failures*.

There has been lots of studies on recovery of lightpath connections [17–20, 53]. The recovery schemes are categorized into two groups; *protection* and *restoration*. In protection schemes, extra wavelength resources are provisioned for backup of the working lightpaths. Protection schemes guarantee 100% recovery against only the predicted failure scenarios. However, they cannot deal with the other scenarios that are not taken into account. Hence, it is difficult to deal with many kinds of failure scenarios with only protection schemes.

On the other hand, restoration schemes reactively search a new path and reserve wavelength after the failure of a working lightpath. Restoration schemes provide more flexible recovery from failures. However, restoration schemes take time for their signaling and

the time increases proportionally to the distance between end nodes. The increase of the hop-length of lightpaths results in the high blocking probability of wavelength reservation during the restoration. Although there are link restoration schemes [17] other than path restoration schemes, they are not available in the cases that all the divert routes between the edge nodes of a failed link due to a massive failure.

The other problem is the amount of control messages for the recovery process. To start restoring disrupted lighptath connections, failure notifications and link-state information are required by the nodes responsible for the restoration. Since the nodes start signaling for the restoration simultaneously, a large amount of control messages are distributed into the control plane. This results in the increase of the propagation delay and the message loss probability and would influence the control sessions for lightpaths having nothing to do with the failures.

In Chapter 5, we propose a restoration scheme that is applicable to any kinds of failures and reduces the number of control messages for the restoration. Our scheme calculates a cycle enclosing the failed part of the network, called a *diverting cycle*, in a distributed way and diverts disrupted lightpaths along the cycle. Our scheme also reduces the number of control messages during the recovery and avoids the congestion in the control plane. We evaluate the performance of our scheme by computer simulations. The results show that our scheme recovers the lightpath connectivity to almost 100% more quickly than the path restoration scheme when the scale of massive failures are not large. When the scale of massive failures are large, our scheme reduces the number of control messages to about the half comparing to the path restoration scheme.

Finally, we conclude this thesis in Chapter 6.

Chapter 2

Dynamic Reconfiguration of Logical Topologies

A Wavelength Division Multiplexing (WDM) network offers a flexible networking infrastructure by assigning the route and wavelength of lightpaths. We can construct an optimal logical topology, by properly setting up the lightpaths. Furthermore, setting up a backup lightpath for each lightpath improves network reliability. When traffic demand changes, a new optimal (or sub-optimal) topology should be obtained by again applying the formulation. Then, we can reconfigure the running topology to the logical topology obtained. However, during this reconfiguration, traffic loss may occur due to the deletion of older lightpaths. In this chapter, we consider reconfiguring the logical topology in reliable WDM-based mesh networks, and we propose five procedures that can be used to reconfigure a running lightpath to a new one. Applying the procedures one by one produces a new logical topology. The procedures mainly focus on utilizing free wavelength resources and the resources of backup lightpaths, which are not used usually for transporting traffic. The results of computer simulations indicate that the traffic loss is remarkably reduced in the 14-node network we used as an example.

2.1 Procedures to Reconfigure Logical Topologies

In this section, we introduce five procedures to reconfigure logical topologies: SWITCH, APPEND, BACKUP, RELEASE, and DELETE. Here, we define *working lightpath* as a lightpath on a current working logical topology on which data traffic is actually transported. We also define *target lightpath* as a new lightpath organizing a part of the new logical topology obtained by a certain logical topology design algorithm. In followings, the shared protection method is assumed, but these procedures are easily applied for the dedicated protection.

2.1.1 Notations

First, let us explain the symbols used in our algorithm.

N: Number of nodes in a network. Each node is assigned a number from 1 to N, respectively.

W: Degree of wavelength multiplexing. Each wavelength is assigned an index number from 1 to W, respectively.

L_1: Set of working lightpaths included in a current logical topology.

L_2: Set of target lightpaths included in a target logical topology.

$b(l)$: Backup lightpath of a lightpath l.

$s(l)$: Source node of a lightpath l.

$d(l)$: Destination node of a lightpath l.

$\lambda(l)$: Wavelength allocated to a lightpath l. $1 \leq \lambda(l) \leq W$.

2.1.2 SWITCH Procedure

Traffic loss is one of the fatal problem during the reconfiguration of logical topologies. A reconfiguration of a logical topology has to be implemented rapidly and smoothly even though there may be significant traffic flow through the logical topology. If a working

lightpath is deleted carelessly, the traffic on it is of course lost and the network performance gets worsen. However, SWITCH procedure can reduce such traffic loss remarkably by switching traffic on a current lightpath l_1 to a target lightpath l_2. These two lightpaths have the same source and destination nodes. The SWITCH procedure is as follows:

Step 1: Reserve wavelength resources for a target lightpath l_2, where $s(l_1) = s(l_2)$ and $d(l_1) = d(l_2)$ are identical, i.e., the source and destination nodes of the working lightpath are identical to the source and destination node of a target lightpath. If the resource reservation is succeeded, go to Step 2. Otherwise quit this procedure.

Step 2: Set the target lightpath l_2. Go to Step 3.

Step 3: Switch the traffic on the working lightpath l_1 to the target l_2. Go to Step 4.

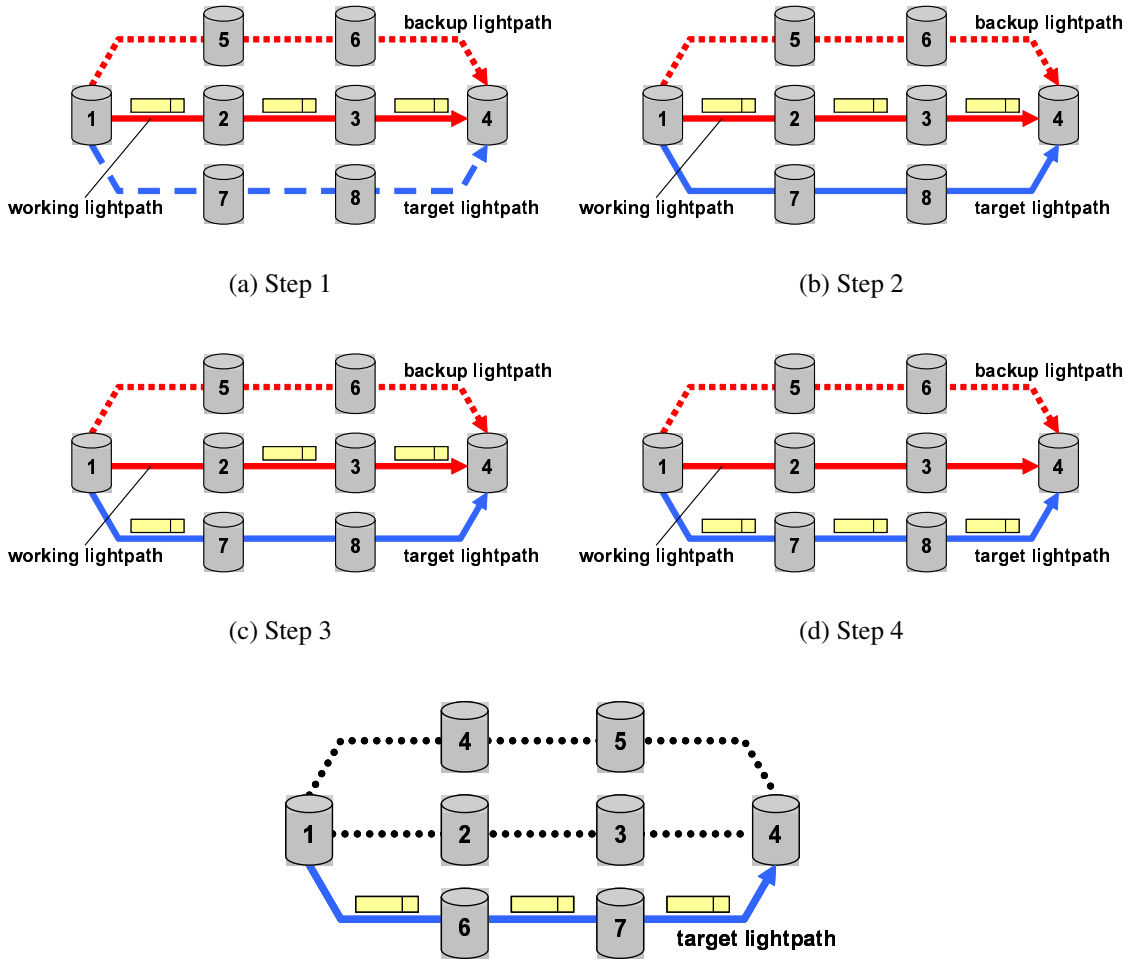
Step 4: If the last packet on the working lightpath reaches the destination node $d(l_1)$, go to Step 5.

Step 5: Delete the working lightpath l_1 and its backup lightpath $b(l_1)$.

Figure 2.1 details the SWITCH procedure. If a portion of the wavelength resources utilized in a working lightpath are required to set up a target lightpath, the working lightpath is to be deleted. Here, we search a target lightpath which has the same source and destination node as the working lightpath. The traffic on the working lightpath is switched to the target lightpath before the former is deleted. Thus, traffic loss doesn't occur. This procedure progresses the reconfiguration effectively because a target lightpath is set up and a working lightpath is released without traffic loss. Hereafter, we describe $SWITCH(l_1, l_2)$ as the SWITCH procedure call for a working lightpath l_1 and a target lightpath l_2.

2.1.3 APPEND Procedure

SWITCH procedure has a harder constraint that a working and target lightpath must have the same source and destination nodes to apply the procedure between these two lightpaths. If there is a target lightpath which cannot be set with the SWITCH procedure, the target



(a) Step 1

(b) Step 2

(c) Step 3

(d) Step 4

(e) Step 5

Figure 2.1: SWITCH procedure

lightpath is to be set with following APPEND procedure. This procedure simply set the target lightpath. We note that the SWITCH procedure is more efficient than the APPEND procedure because the former releases the wavelength resources reserved for a working and its backup lightpath without traffic loss. $APPEND(l_2)$ works as follows to create a target lightpath l_2.

Step 1: Reserve wavelength resources for a target lightpath l_2. If the reservation is succeeded, go to Step 2. Otherwise, quit this procedure.

Step 2: Set the target lightpath l_2.

2.1.4 BACKUP Procedure

If a portion of the wavelength resources for a working lightpath is required by one or more target lightpaths, and if there are no target lightpaths whose source and destination nodes are identical to those of the working lightpath, the working lightpath will be discarded without protecting its traffic. This usually results in traffic loss, but in a reliable WDM network with backup lightpaths, it is possible to avoid this loss by utilizing the backup lightpath prepared for the working lightpath (BACKUP procedure). The BACKUP procedure is as follows:

Step 1: Reserve wavelength resources for the backup lightpath $b(l_1)$ of a working primary lightpath l_1. Go to Step 2.

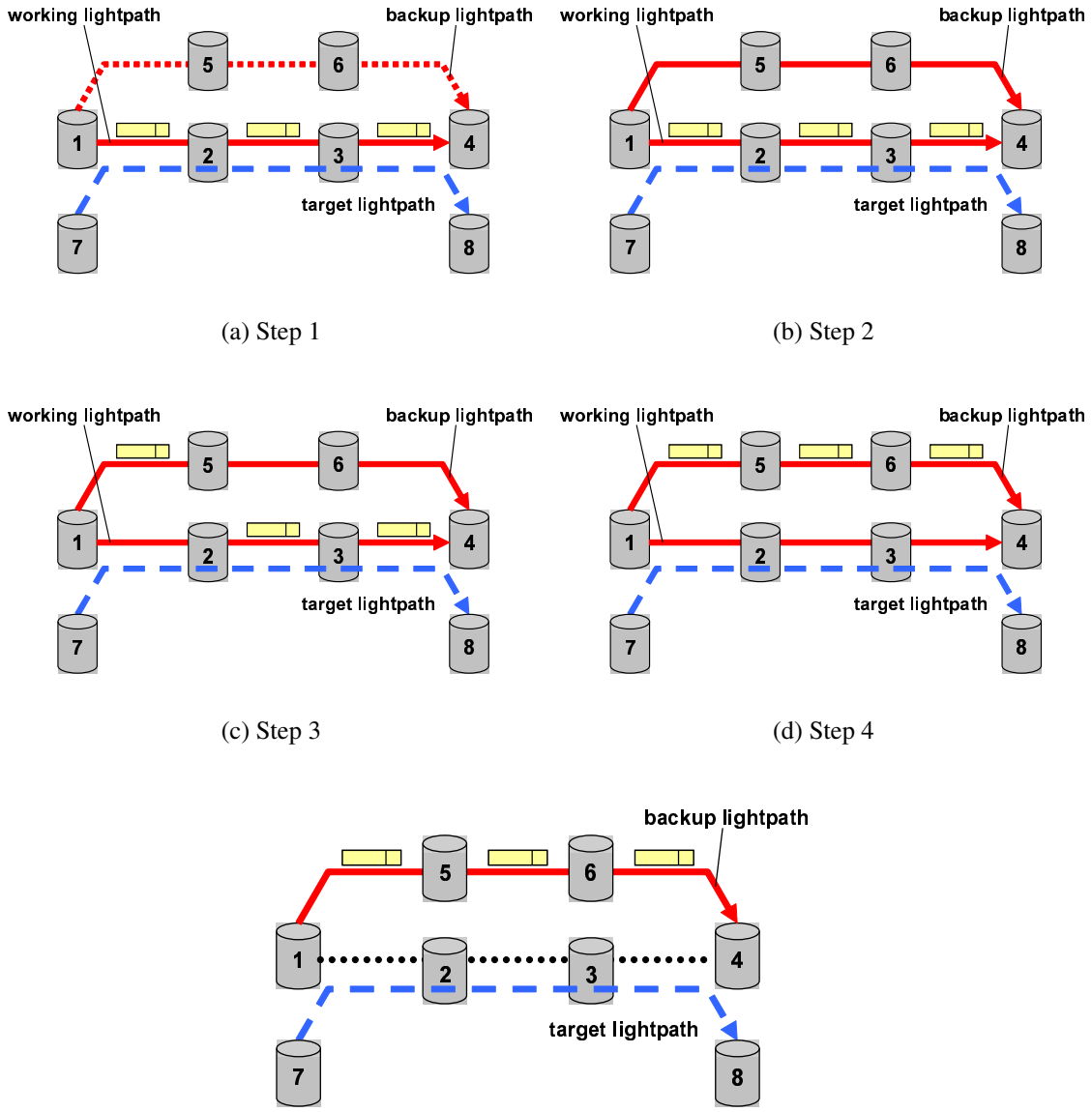
Step 2: Set the backup lightpath $b(l_1)$. Go to Step 3.

Step 3: Switch the traffic on the working lightpath l_1 to its backup $b(l_1)$. Go to Step 4.

Step 4: If the last packet on the working lightpath reaches the destination node $d(l_1)$, go to Step 5.

Step 5: Delete the working lightpath l_1.

Figure 2.2 illustrates the BACKUP procedure. Here, the working and its backup lightpath are prepared from nodes 1 to 4. Suppose that the working lightpath is not necessary in the target logical topology and that there is not a target lightpath pair for the SWITCH procedure. In this situation, the working lightpath will be finally deleted. However, if the backup lightpath of the working lightpath is available, i.e., if all the wavelength resources of the backup lightpath are not shared with other backup lightpaths and are not required to set up target lightpaths, the traffic running through the working lightpath can be switched onto the backup lightpath. The backup lightpath is left until the reconfiguration of all the target primary lightpaths finished. We describe $BACKUP(l_1)$ as the procedure call to switch the traffic on a lightpath l_1 to its backup lightpath $b(l_1)$.



(a) Step 1

(b) Step 2

(c) Step 3

(d) Step 4

(e) Step 5

Figure 2.2: BACKUP procedure

2.1.5 RELEASE Procedure

Suppose that there is a working lightpath whose resources are required to set up target lightpaths and that it cannot be applied SWITCH, APPEND or BACKUP procedure. In this case, traffic of the working lightpath tends to be lost by deleting it. However, with the RELEASE procedure, we can avoid the unnecessary deletions of working lightpaths.

The RELEASE procedure releases the wavelength resources of the backup lightpath for

a working lightpath. After that, the reconfiguration continues to make progress using the SWITCH or APPEND procedure with the released wavelength resources. $RELEASE(l_1)$ is the procedure call to release the wavelength resources of a backup lightpath $b(l_1)$:

Step 1: Release wavelength resources reserved for a backup lightpath.

2.1.6 DELETE Procedure

The DELETE procedure deletes a working lightpath without protecting its traffic. The traffic on the deleted lightpath is lost. Note that the DELETE procedure is the only procedure that results in a traffic loss. The working lightpath should be applied the RELEASE procedure before applying the DELETE procedure. $DELETE(l_1)$ is the procedure call to tear down a lightpath l_1:

Step 1: Release wavelength resources reserved for a working lightpath.

2.1.7 Wavelength Re-allocation

Reconfigurations of logical topologies can be achieved using those five procedures. However, there is plenty of room for improvement in making the reconfigurations more efficient. Wavelength re-allocation provides one means. It re-allocates a new wavelength to a target lightpath, if another wavelength resources on the same route of the target lightpath are available. This extension is applied to SWITCH, APPEND, or BACKUP procedures. Suppose that the reserved wavelength resources on a wavelength (say λ_1) are required to set up a target lightpath. Wavelength re-allocation assigns another available wavelength (assume λ_i as $i \neq 1$) to the target lightpath, which solves the conflict on λ_1. After that, the SWITCH or APPEND procedure set up the target lightpath that uses the new λ_i wavelength resources.

This re-allocation is done when a portion of the wavelength resources of the target lightpath are reserved for working or backup lightpaths, and when there are free wavelength resources that are not required to set up other target lightpaths. When we use the wavelength re-allocation, Step 1 of SWITCH procedure is revised as follows:

Step 1.1: Reserve wavelength resources for a target lightpath l_2, where $s(l_1) = s(l_2)$ and $d(l_1) = d(l_2)$. If the resource reservation is succeeded, go to Step 2. Otherwise go to Step 1.2.

Step 1.2: For each wavelength w $(1 \leq w \leq W,\ w \neq \lambda(l_2))$, try Step 1.3. If all trials fail, quit this procedure.

Step 1.3: Check the resources of wavelength w along the route of the target path. If the resources are not reserved and not required to set up any other lightpaths, go to Step 1.4. Otherwise go back to Step 1.2.

Step 1.4: Re-allocate w to the target lightpath and reserve the re-allocated resources. Go to Step 2.

Here, we search free wavelength by the First-Fit policy in this re-allocation. The network performance of the re-allocated lightpath is identical to the original because the routes of both lightpaths seen from the upper layer are the same. Therefore, this wavelength re-allocation does not have any bad effect on the reconfigurations of logical topologies.

2.2 Reconfiguration Algorithm

To relieve the unbalance of the traffic load on WDM mesh networks, we need reconfigurations of logical topologies. In this section, we propose a reconfiguration algorithm of logical topologies in WDM mesh networks. This algorithm is composed of five procedures described above. Since only the DELETE procedure leads to traffic loss, we heuristically suppress the number of DELETE procedure calls during a reconfiguration. It requires two logical topologies: a current logical topology and a target logical topology to be reconfigured. Therefore, the increase or decrease of traffic volume during the reconfiguration is not considered here.

We use a variable P in our reconfiguration algorithm to store the number of SWITCH, APPEND, BACKUP procedure calls in each iteration. We also define C as a set of working lightpaths which are candidates for a pair of a target lightpath to execute SWITCH procedure. In this chapter, we assume that no network failures occur during reconfiguration,

because reconfiguration is invoked once per, say, one week or month.

2.2.1 Flow of Reconfiguration Algorithm

The reconfiguration algorithm we propose is as follows:

Step 1: For each target lightpath $l_2 \in L_2$, if there is a working lightpath $l_1 \in L_1$ which has the same route and wavelength as l_2, delete the elements from L_1 and L_2. $P \leftarrow 0$. $C \leftarrow \phi$. Go to Step 2.

Step 2: For each target lightpath l_2, try following steps. After that, go to Step 3.

- Step 2.1: Add working lightpaths $l_1 \in L_1$ into C which fulfills these conditions: $s(l_1) = s(l_2)$, $d(l_1) = d(l_2)$ and l_1 does not reserve wavelength resources required for l_2. If $C \neq \phi$, go to Step 2.2. Otherwise, go to Step 2.3.
- Step 2.2: Among the elements in C, select a working lightpath $l_1^{'}$ whose wavelength resources of both primary and backup lightpaths are most utilized to set up target lightpaths in L_2. Execute $SWITCH(l_1^{'}, l_2)$. Delete $l_1^{'}$ and l_2 from L_1 and L_2, respectively. $P \leftarrow P+1$. $C \leftarrow \phi$. Go back to Step 2.
- Step 2.3: Execute $APPEND(l_2)$. Delete l_2 from L_2. $P \leftarrow P + 1$. Go to Step 2.

Step 3: If $L_2 = \phi$, go to Step 6. Otherwise, go to Step 4.

Step 4: For each working lightpath $l_1 \in L_1$, which meets that there are no target lightpaths $l_2 \in L_2$ such as $s(l_1) = s(l_2)$ and $d(l_1) = d(l_2)$, execute $BACKUP(l_1)$ and, if it succeeds, delete l_1 form L_1 and $P \leftarrow P + 1$. Go to Step 5.

Step 5: If $P > 0$, $P \leftarrow 0$ and go back to Step 2. Otherwise, go to Step 5.1.

- Step 5.1: If there are working lightpaths whose backup lightpaths are not released, go to Step 5.2. Otherwise, go to Step 5.3.

Step 5.2: Select a working lightpath l_1 where the wavelength resources of backup lightpath ($b(l_1)$) are most utilized to set up target lightpaths in L_2. Execute $RELEASE(l_1)$. $P \leftarrow 0$. Go back to Step 2.

Step 5.3: Select a working lightpath l_1 whose wavelength resources of primary lightpath are most utilized to set up target lightpaths in L_2. Execute $DELETE(l_1)$. Delete l_1 form L_1. $P \leftarrow 0$. Go back to Step 2.

Step 6: Delete all of the remaining working lightpaths in L_1 and those backup lightpaths. Go to Step 7.

Step 7: Restore the re-allocated wavelengths of target lightpaths to the original wavelengths. Go to Step 8.

Step 8: Reserve the wavelength resources of the backup lightpaths for the target lightpaths.

In Step 1, we detect working lightpaths which are also included in L_2. These working lightpaths are left as target lightpaths. From Steps 2 through 5, we set all the target lightpaths. Backup lightpaths for each target lightpath are set in Step 8. This is based on the assumption that no network failures occur during a reconfiguration.

In Step 2, we check whether the wavelength resources to set target lightpaths are available or not. We give priority to the SWITCH procedure over the APPEND procedure because of the differences in their efficiency (see Subsection 2.1.3). Hence, we try to apply the SWITCH procedure in setting up the target lightpath at first (Step 2.2). In Step 2.2, a working lightpath, l_1', is chosen as a pair of a target lightpath, l_2, from the set of the candidates.

To make the SWITCH procedure more efficient, we select the pair heuristically as follows. The wavelength resources released by the SWITCH procedure can be utilized for setting up remaining target lightpath. We therefore choose a working lightpath as a pair of a target lightpath, such that the working lightpath holds the most amounts of wavelength resources required to set up other remaining target lightpaths. Other strategies are to select

lightpaths in descending (ascending) order of the number of hop-counts. We call these strategies longest-first-strategy (shortest-first-strategy).

As the reconfiguration continues by the SWITCH or APPEND procedure, target lightpaths reserve their wavelength resources. And available wavelength resources are decreased. Ultimately, no target lightpaths can be set because the available wavelength resources are exhausted. In Step 3, if all target lightpaths in L_2 have already been created, we can go to Step 6. Otherwise, in Step 4, we try to find free wavelength resources to utilize set up other remaining target lightpaths by applying BACKUP procedures. If one or more trial succeeds, we obtain new available wavelength resources without traffic loss. Then we go back to Step 2 and try to set up the rest of the target lightpaths in L_2.

In Step 6, all target lightpaths are created and traffic in a network is accommodated by these target lightpaths. Hence, we can delete the old working lightpaths in L_1. If there are lightpaths whose wavelengths are re-allocated by wavelength re-allocation, we tune the re-allocated wavelength to the original one in Step 7. In Step 8, reserve the wavelength resources for the backup lightpaths of the target lightpaths and finish the reconfiguration.

In this chapter, we assume that one by one operations of reconfiguration procedures. The operations of these reconfigurations can be controlled in either centralized or distributed systems. Multiple operations may be executed at the same time as long as the corresponding lightpaths are independent of the operations each other. The time for a reconfiguration is shortened progressing the reconfiguration in parallel, but it is out of the scope of this chapter.

2.2.2 Heuristic Selection Strategy

Among the five procedures described in Section 2.1, the SWITCH, RELEASE and DELETE procedures enables wavelength resources to be available. If there are several working lightpaths to which these procedures can be applied, we select one working lightpath (and then release or delete it) such that other procedures can be applied efficiently.

In this chapter, we propose a heuristic selection strategy to select the working lightpath. Our strategy selects the working lightpath which holds most conflicts with target lightpaths. By releasing or deleting it maximize the number of target lightpaths to be set up. This

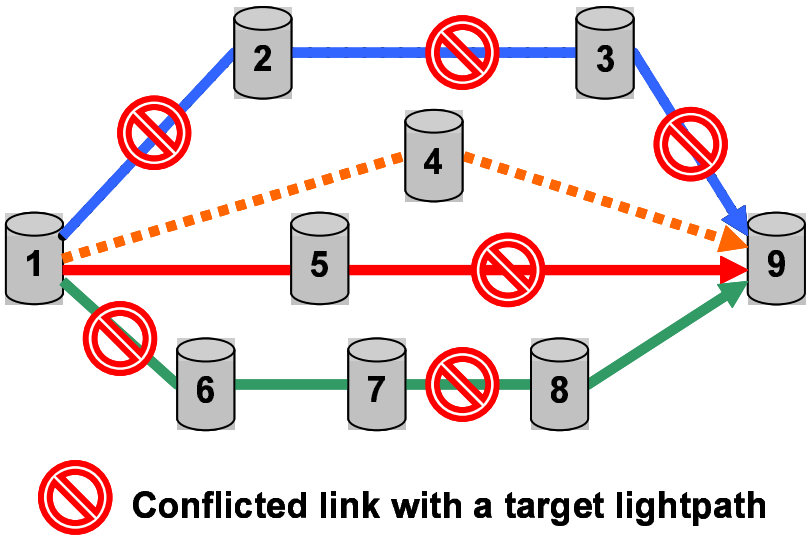


Figure 2.3: Heuristic selection strategy

strategy is used in Step 2.2, Step 5.2 and Step 5.3. Figure 2.3 depicts an example of our strategy. Here, we assume that a target lightpath from node 1 to node 9 via node 4 is being set up by the SWITCH procedure and there exists threee candidate working lightpath (1 $\rightarrow$ 2 $\rightarrow$ 3 $\rightarrow$ 9, 1 $\rightarrow$ 5 $\rightarrow$ 9 and 1 $\rightarrow$ 6 $\rightarrow$ 7 $\rightarrow$ 8 $\rightarrow$ 9). In such a situation, our strategy selects the working lithapth, 1 $\rightarrow$ 2 $\rightarrow$ 3 $\rightarrow$ 9, as a pair of the target lightpath because it holds three conflicts. After the traffic on the working lightpath is switched into the target lightpath, three wavelength resources are available and used for the other target lightpaths.

2.3 Evaluation

Our algorithm selects a working lightpath heuristically when the lightpath is required to delete in a SWITCH or DELETE procedure. We evaluated the effectiveness of the heuristic selection at first.

2.3.1 Performance of Proposed Reconfiguration Algorithm

We evaluated our reconfiguration algorithm with a wide area network model and various degrees of wavelengths.

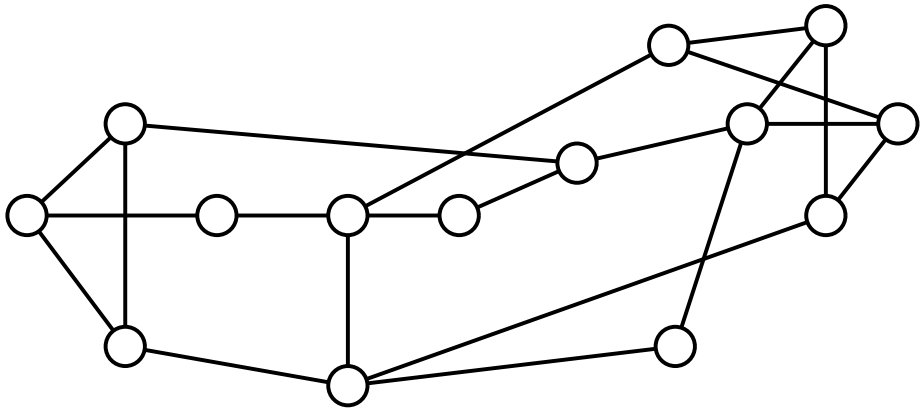


Figure 2.4: NSFNET (14 nodes and 21 links)

Table 2.1: Properties of logical topologies to be reconfigured

Number of wavelengths	16	32	64	128	256
Number of primary lightpaths	210	404	779	1527	3082

Evaluation Model

Here, we explain our evaluation model. We used the NSFNET, which has 14 nodes and 21 links, as a network model. This network topology is shown in Figure 2.4. We generated series of traffic matrices $T^1, T^2, \cdots, T^k$, where the elements of each traffic matrix are set a random value between zero and the transmission capacity of a fiber.

We evaluated the performance of our algorithm with logical topologies where the patterns are changed randomly since this is the most difficult case to reconfigure logical topologies without traffic loss. We examined the performance of our algorithm in the worst case. In this section, we set $k = 5$. We simulated reconfigurations when the degree of wavelength multiplexing is 16, 32, 64, 128, or 256.

Logical Topology Design Algorithm

To generate the logical topologies for those traffic matrices, we used a simple design algorithm (SDA). SDA works as follows. Given a traffic matrix, SDA select a node pair which has a largest traffic demand in the traffic matrix. Then, SDA sets up a primary lightpath and a backup one between the nodes, and reduces the transmission capacity of a lightpath from the traffic demand (in this section, we set the transmission capacity of a lightpath as

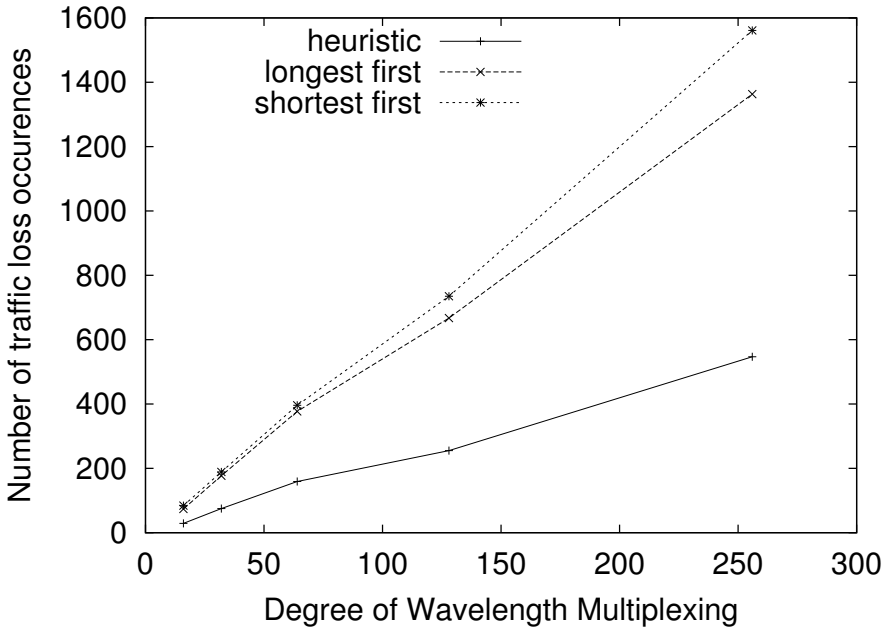


Figure 2.5: Number of traffic loss occurrences during a reconfiguration of a logical topology

10 Gbps). Each of primary and backup lightpath is selected on the shortest route, in terms of the propagation delay, among currently available routes. Backup lightpaths are selected on a disjoint sets of links of corresponding working lightpaths. And SDA also deals with shared protection strategy. For each lightpath, its wavelength is assigned based on First-Fit policy. If SDA fails to set up a primary and a backup lightpaths for the traffic demand from a source node to a destination node, SDA sets the traffic demand zero.

The properties of the logical topologies obtained by SDA are shown in Table 2.1. The second row of the table shows the average number of lightpaths excluding backup lightpaths in a logical topology. The average utilization of wavelength resources on links is 95%. We do not consider the traffic demands which are not accommodated by the logical topology design algorithm because only the loss of traffic during reconfiguration is our concern.

Effectiveness of Heuristic Selection Strategy

We examined the effectiveness of those three strategies by the number of times of DELETE procedure calls (i.e., the number of traffic loss occurrences) in a reconfiguration. The

Table 2.2: Combinations of the procedures for reconfigurations

Algorithm	*1*	*2*	*3*	*4*
SWITCH	Enabled	Enabled	Enabled	Enabled
APPEND	Enabled	Enabled	Enabled	Enabled
BACKUP	–	Enabled	–	Enabled
RELEASE	Enabled	Enabled	Enabled	Enabled
DELETE	Enabled	Enabled	Enabled	Enabled
re-allocation	–	–	Enabled	Enabled

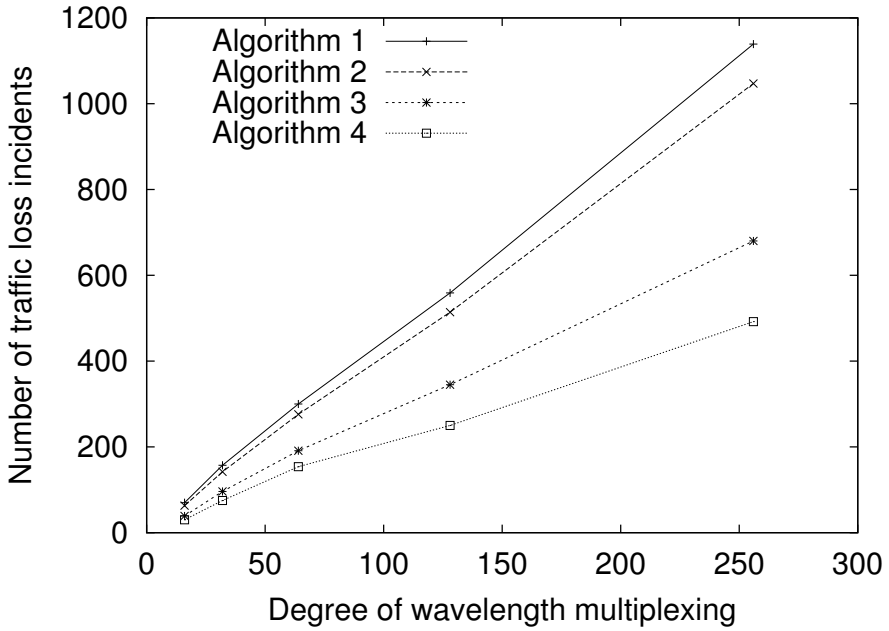


Figure 2.6: Effectiveness of the procedures for reconfigurations

results are shown in Figure 2.5 where the average numbers of DELETE procedure calls are dependent on the degree of wavelength multiplexing. This figure shows traffic loss of the algorithm with heuristic strategy is more than twice as little as those of the algorithms with longest/shortest-strategy. The heuristic strategy is effective and thus we will obtain the further results by adopting the heuristic strategy.

Effectiveness of the Procedures to Reconfigure Logical Topologies

Next, we consider four algorithms for comparison: Algorithms 1, 2, 3, and 4 listed in Table 2.2. Algorithm 1 is a basic algorithm composed of SWITCH, RELEASE, APPEND,

and DELETE procedures. Algorithm 2 allows BACKUP procedure moreover, whereas Algorithm 3 allows wavelength re-allocation. Algorithm 4 is the proposed algorithm, which employs all procedures and wavelength re-allocation.

We also examined the performances of those four algorithms by the number of times of DELETE procedure calls in a reconfiguration (i.e., the number of traffic loss occurrences). From Figure 2.6, traffic loss decreases in order from Algorithm 1 to 2 to 3 to 4, and there is a relatively large gap between Algorithms 2 and 3. This result shows that wavelength re-allocation is useful in reconfigurations of logical topologies. On the other hand, BACKUP procedure is still less effective than wavelength re-allocation. We believe that the shared protection strategy makes BACKUP procedure less effective.

2.3.2 Effectiveness of Reconfiguring Logical Topologies

We had another evaluation to examine the effectiveness of reconfiguring logical topologies. In this subsection, we observed the performance of our reconfiguration algorithm when the dynamic changing of traffic model is applied and the changing of the average utilization of wavelength resources on links.

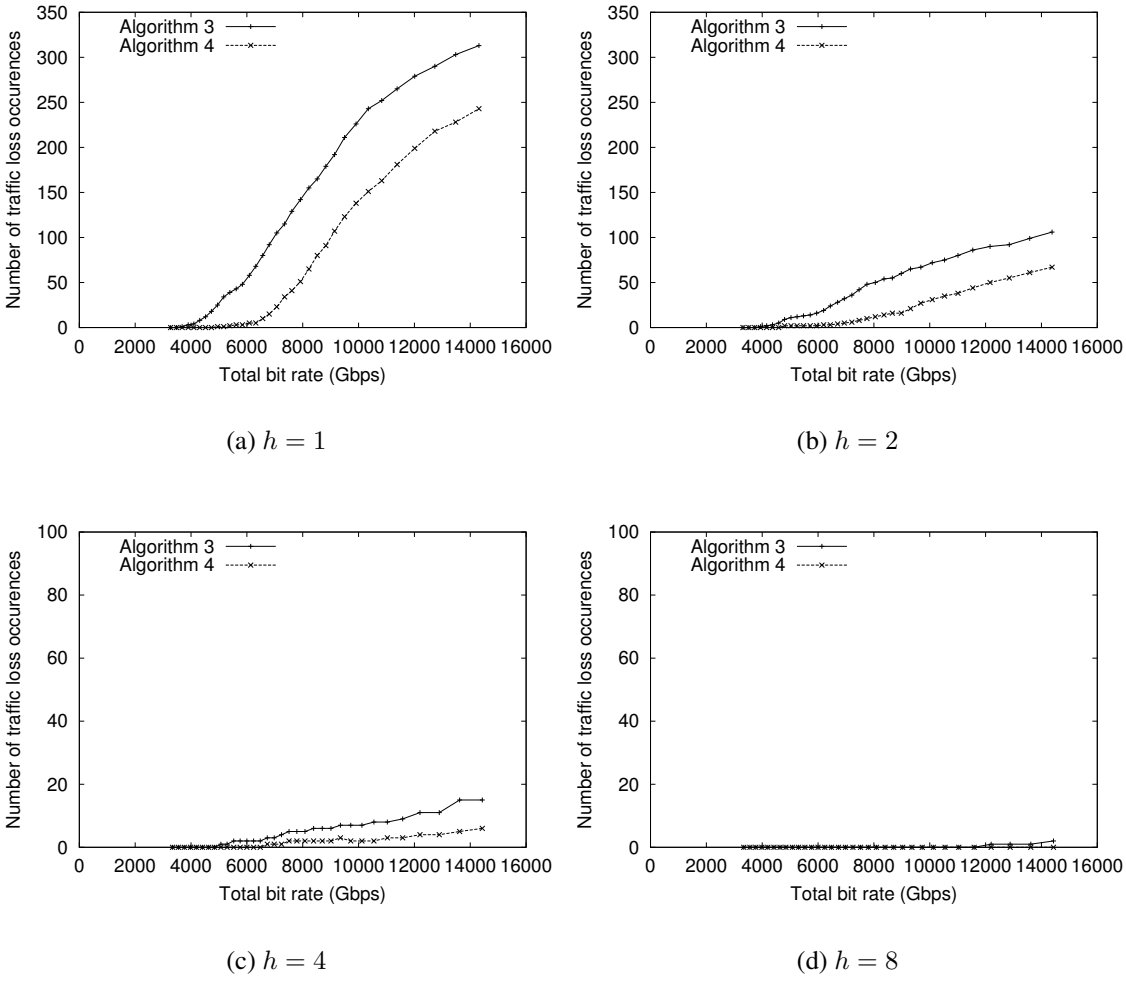
Traffic Transition Model

We have generated a series of traffic matrices $T^1, T^2, \cdots, T^k$, where those elements are random number between zero and the transmission capacity of a fiber. Here, we use another traffic model similar to the model described in Ref. [42]. Eq. (2.1) shows the traffic model. $T^1, T^2, \cdots, T^k$ are the same traffic matrices used in the above simulation. The parameter h gives h-step traffic transitions between T^i and T^{i+1}.

$$T^{k,l} = round\left[(1 - \frac{l}{h})T^k + \frac{l}{h}T^{k+1}\right]. \tag{2.1}$$

Consideration of Effective Reconfiguration

We use NSFNET as a network model to show the effect of parameter h. The degree of wavelength multiplexing is 128 or 256. We set $k = 5$ and $h = 1, 4, 8$. Logical topologies are generated by applying SDA to traffic matrices $T^{m,l}$ $(1 \leq m \leq k,\ 0 \leq l \leq h)$.



(a) $h = 1$

(b) $h = 2$

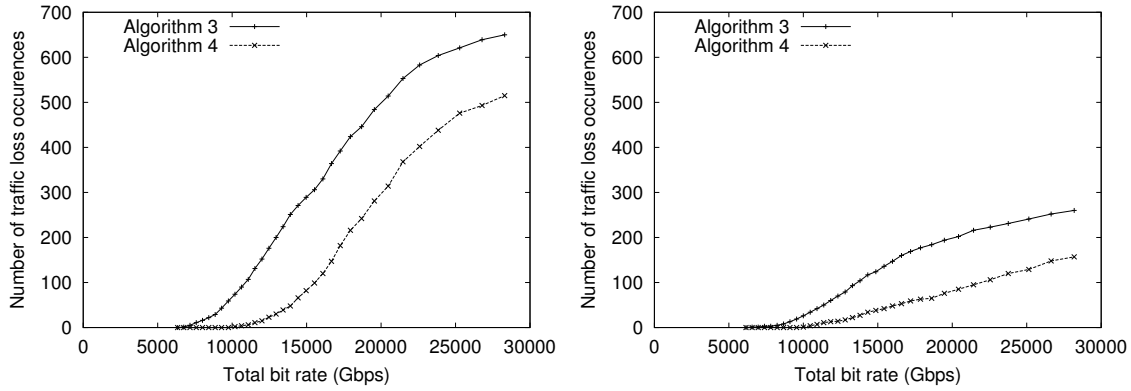
(c) $h = 4$

(d) $h = 8$

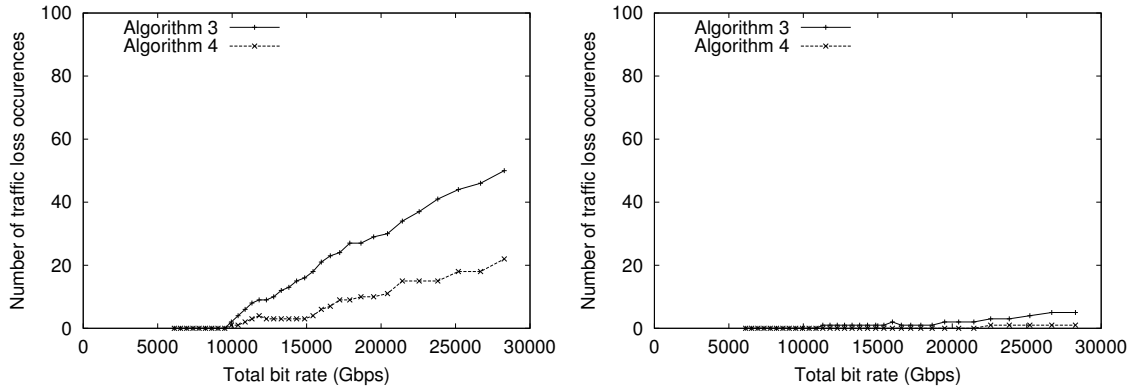
Figure 2.7: Relation between the performance of reconfiguration algorithms and the bit rate of the whole of logical topology ($W = 128$)

We examined the performances of two algorithms, Algorithms 3 and 4, by changing the average utilization of wavelength resources from 30% to 96% by a step of 2%. We realize the target utilization of wavelength resources in logical topologies by inserting the check process of wavelength usage in SDA.

The results are shown in Figures 2.7 and 2.8. The vertical axis shows the average number of traffic loss occurrences and the cross axis is the average of total bit rate carried in the whole network, where the bit rate is obtained by the number of primary lightpaths times the transmission capacity of the lightpath. According to Figures 2.7(a) and 2.8(a), the number of lightpaths Algorithm 4 can reconfigure without traffic loss is about one and a half times as much as the number of lightpaths Algorithm 3 can do. However, when h gets



(a) $h = 1$ (b) $h = 2$



(c) $h = 4$ (d) $h = 8$

Figure 2.8: Relation between the performance of reconfiguration algorithms and the bit rate of the whole of logical topology ($W = 256$)

larger, i.e., when logical topologies change gradually, the difference of the performance between Algorithms 3 and 4 becomes smaller. In the figures, when $h = 8$, the results of Algorithms 3 and 4 are almost same. It indicates that BACKUP procedure is much less efficient when the utilization of wavelength resources is high and when the changes of logical topologies are small.

To explain the differences which result from changing values of h, we compare the number of procedure calls by our algorithm, where the utilization of the wavelength resources in NSFNET is 96%, in Table 2.3. From this table, as the value of h becomes larger, the number of the SWITCH procedure calls also becomes larger, whereas the number of the APPEND procedure calls decreases. The SWITCH procedure protects traffic on working

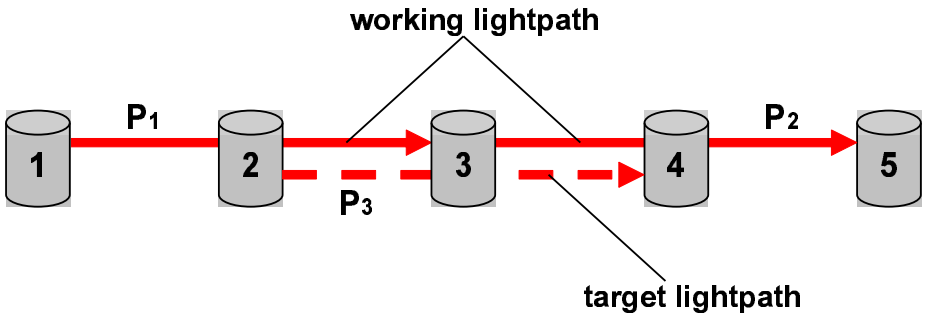
Table 2.3: Number of procedure calls in a reconfiguration

waves	h	*SWITCH*	*APPEND*	*BACKUP*	*RELEASE*	*DELETE*	*total*
128	1	373	913	110	973	243	2512
128	8	932	164	11	131	1	2212
256	1	727	1839	226	1961	495	4995
256	8	1856	329	22	282	1	4449

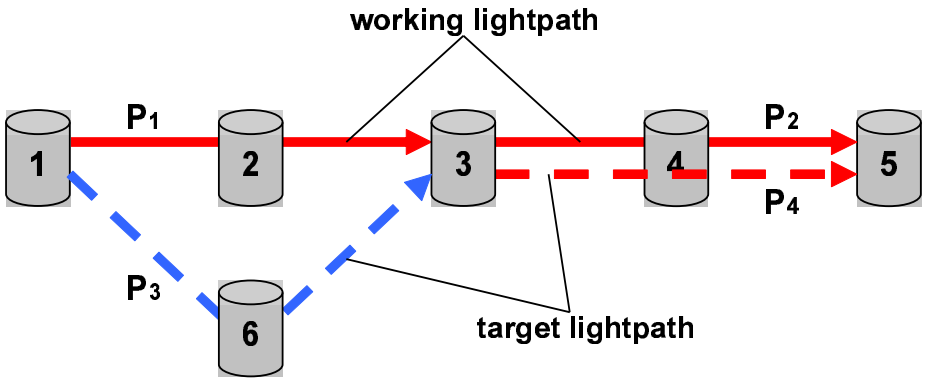
lightpaths, and thus the number of traffic loss occurrences becomes to a few or zero. We illustrate this situation in Figure 2.9. The working lightpaths are to be disrupted and the target lightpaths are set up in a reconfiguration. When h is 1, each element of traffic matrices is changed randomly. Therefore, there may be many wavelength resource conflicts between working lightpaths and target lightpaths. Suppose that two working lightpaths P_1 and P_2 require a certain wavelength on the links 2–3 and 3–4, respectively, and a target lightpath, P_3, requires the same wavelength on the link 2–4. In this case, $APPEND(P_3)$ is tried at first. Then, $BACKUP(P_1)$ and $BACKUP(P_2)$ are next tried. If these trials do not succeed, $RELEASE(P_1)$ and $RELEASE(P_2)$ are executed. Finally, $DELETE(P_1)$ and $DELETE(P_2)$ are executed. Thus, the numbers of APPEND, BACKUP, RELEASE, and DELETE procedure calls become larger.

On the other hand, if h is large, a gradual change on the traffic matrices results in the requesting almost the same number of lightpaths between node pairs. In Figure 2.9(b), the working lightpath P_2 and the target lightpath P_4 are assigned the same route and wavelength. Therefore, P_2 is remained as the target lightpath P_4, and need not be reconfigured. Or the route of the target lightpath P_3 may be selected on the different route from that used by the working lightpath P_1. In this case, $SWITCH(P_1, P_3)$ is executed. Therefore, the number of SWITCH procedure calls gets larger and other procedure calls decrease.

We go back to Figures 2.7 and 2.8. The results show that changing to the very different logical topology, which may be an optimal logical topology, will be tolerable against the moderate utilization of wavelength resources. However, for the higher utilization of wavelength resources, the gradual changes on the traffic matrices greatly reduce the traffic loss during the reconfiguration. Although Algorithm 4 reduces the number of DELETE procedure calls compared to Algorithm 3, we need a logical topology design algorithm which generate sub-optimal topology with less changes on lightpaths, rather than the optimal one.



(a) Typical case where $h = 1$



(b) Typical case where $h = 2$

Figure 2.9: Illustrative examples of the effect of parameter h

To investigate such a design algorithm is beyond the scope of this study and constitutes the scope of future work.

2.4 Summary

In this chapter, we proposed an algorithm to reconfigure logical topologies in reliable WDM mesh networks. The algorithm is composed of five procedures to set up or tear down lightpaths. We first evaluated the performance of the algorithm with randomly generated traffic, and then evaluated with changes of network load using a dynamic traffic model. The results show that changing to the very different logical topology will be tolerable for the moderate utilization of wavelength resources. We also found that the gradual changes on the traffic demand greatly reduce the traffic loss during the reconfiguration.

The objective of our current and future work is to investigate the logical topology design algorithm which generates sub-optimal topology with fewer changes on lightpaths.

Chapter 3

Routing Scheme for Large-Scaled Wavelength-Routed Networks

Recently, progress has been made in the Generalized Multi-Protocol Label Switching (GMPLS) and Automatic Switched Optical Networks (ASON) standardizations. These technologies realize construction of large-scaled optical networks, interconnections among single-domain Wavelength Division Multiplexing (WDM) networks, and direct communication over multi-domain WDM networks. Meanwhile, it is known that the topology of the Internet exhibits the power-law attribute. Since the topology of the Internet, which is constructed by interconnecting ASs, exhibits the power-law, there is a possibility that large-scale WDM networks, which are constructed by interconnecting WDM networks, will also exhibit the power-law attribute. One of the structural properties of a topology that adheres to the power-law is that most nodes have just a few links, although some have a tremendous number of them. Another property is that the average distance between nodes is smaller than in a mesh-like network. A natural question is how such a structural property performs in WDM networks.

In this chapter, we first investigate the property of the power-law attribute of physical topologies for WDM networks. We compare the performance of WDM networks with mesh-like and power-law topologies, and show that links connected to high-degree nodes are bottlenecks in power-law topologies. To relax this, we introduce a concept of virtual fiber, which consists of two or more fibers, and propose its configuration method to utilize

wavelength resources more effectively. We compare performances of power-law networks with and without our method by computer simulations. The results show that our method reduces the blocking probabilities by more than one order of magnitude.

3.1 Topology Models

While the current topology of the Internet has been investigated for actual trace data, there are many studies that focus on modeling methods for Internet topology. In this section, we first describe the ER (Erdös-Rényi) model [54] in which links are randomly placed between nodes (Figure 3.1(a)). We then introduce the BA (Barabási-Albert) model [55] in which the topology grows incrementally and links are placed based on the connectivities of the topologies to form power-law networks (Figure 3.1(b)).

3.1.1 ER (Erdös-Rényi) Model

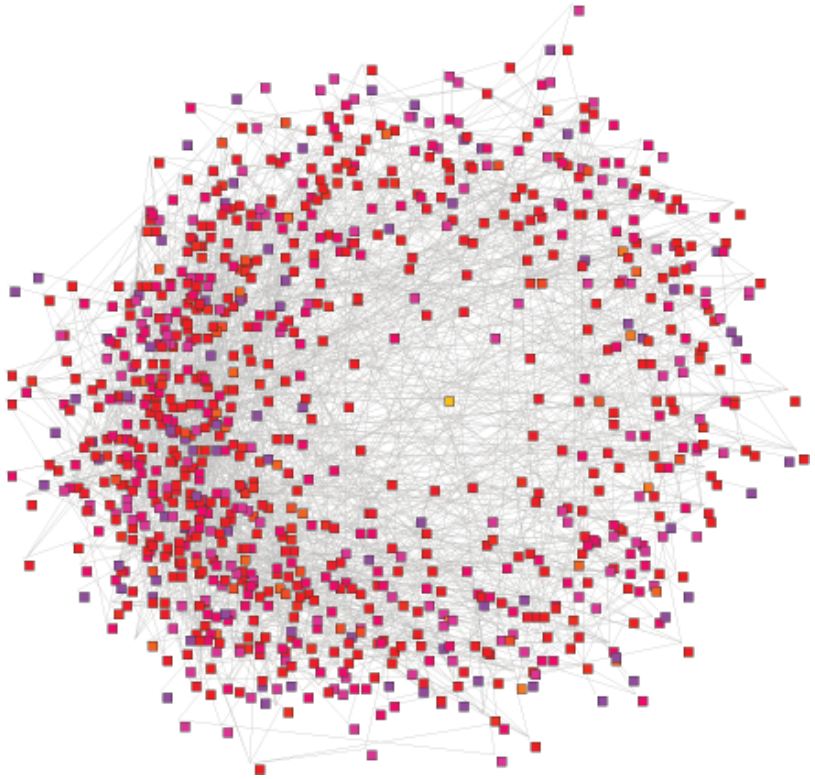
The ER model was designed by Erdös and Rényi to describe communication networks. They assumed that such systems could be modeled with connected nodes of randomly placed links usually called random networks. In this model, the number of nodes N is given at first, and every two nodes are connected with the fixed probability p. Thus, the ER model generates a random network. The probability $P(k)$ that a node has degree (number of links) k is given as

$$P(k) = \binom{N-1}{k} p^k (1-p)^{N-1-k}. \tag{3.1}$$

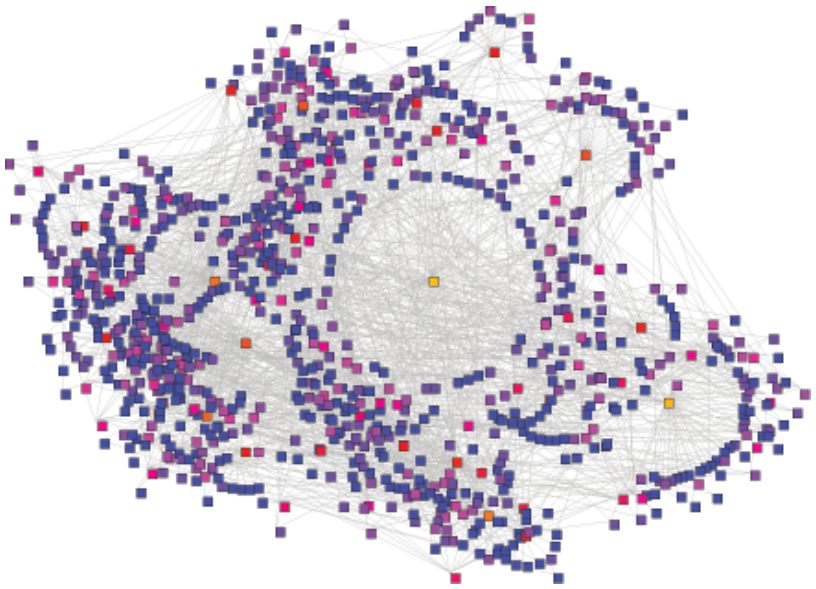
In addition, with large N and small p, Eq. (3.1) becomes

$$P(k) = \frac{\lambda^k e^{-\lambda}}{k!}, \tag{3.2}$$

where $\lambda = pN$. From Eq. (3.2), the distribution of the degrees of the nodes in a random network generated by the ER model follows a Poisson distribution [56].



(a) Random network



(b) Power-law network

Figure 3.1: Topologies of a random network and a power-law network

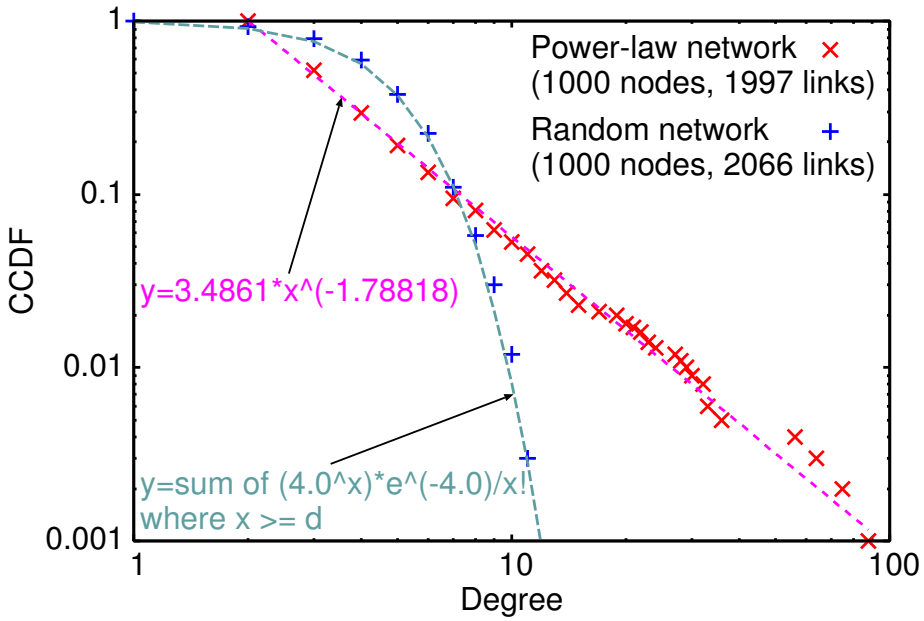


Figure 3.2: Complementary cumulative distribution of node degrees in topologies generated with the ER and BA models

3.1.2 BA (Barabási-Albert) Model

Barabási and Albert designed their model to emulate the growth of such large-scale networks as the Internet. The BA model is characterized by two features that the ER model does not have: *Incremental Growth* and *Preferential Attachment*. Generating a topology is started with a small number of nodes m_0.

1. *Incremental Growth*: Add a new node at each time step.

2. *Preferential Attachment*: Connect the new node with two other different nodes, which are chosen with the probability Π (k_i is the degree of node i).

$$\Pi(k_i) = \frac{k_i}{\sum_j k_j}. \tag{3.3}$$

3.1.3 Properties of Random and Power-Law Networks

Figure 3.2 shows complementary cumulative distribution of node degrees in the topologies generated by the ER and BA models. The number of nodes is 1,000. The connection probability of the ER model is 0.002 and 2,066 links are generated. The number of nodes

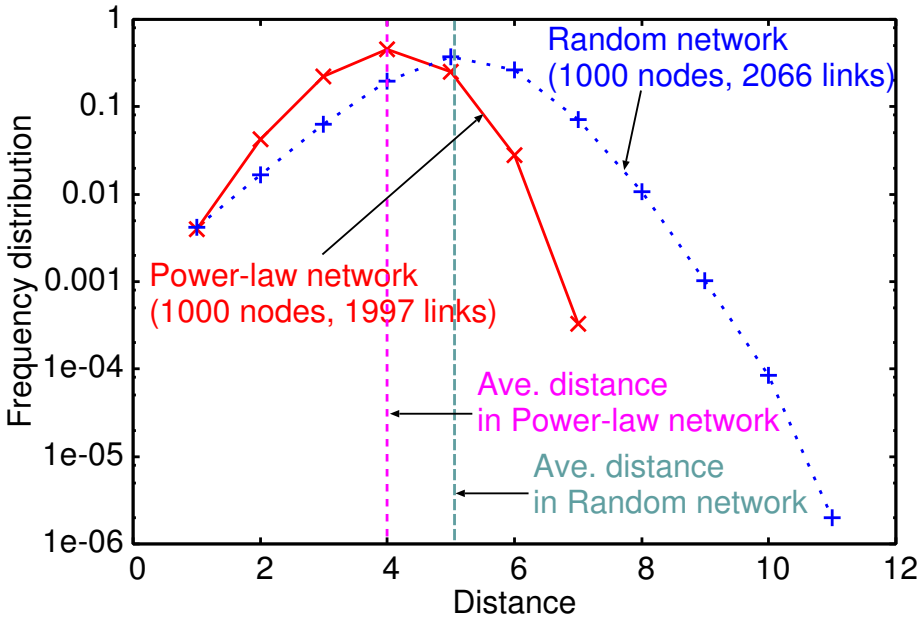
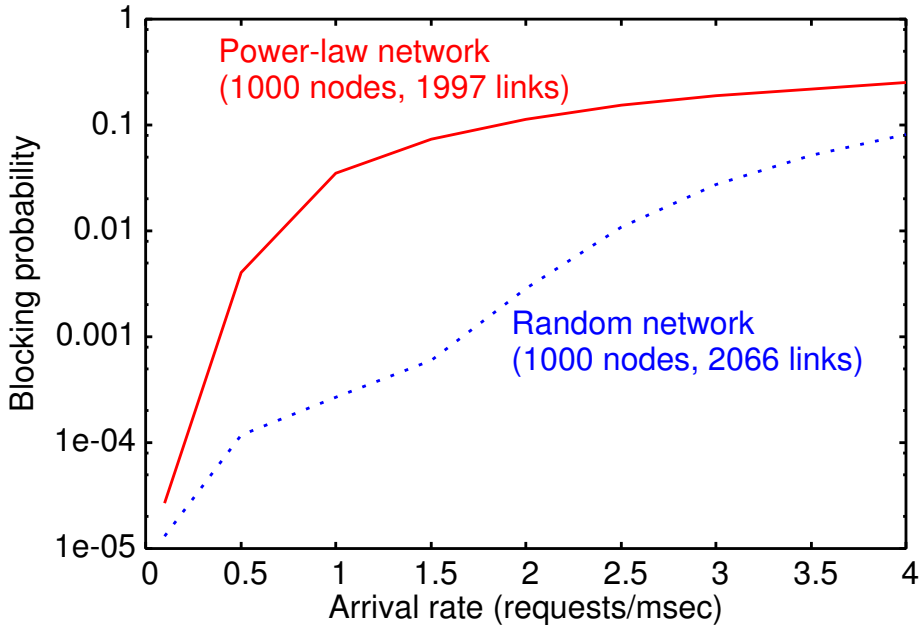


Figure 3.3: Distributions of distances between nodes in topologies generated with the ER and BA models

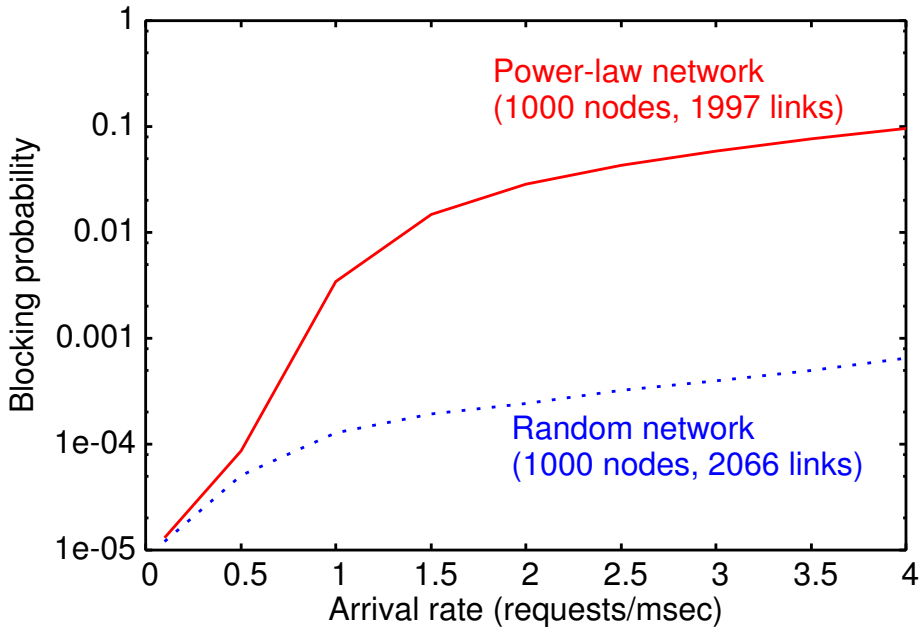
at the initial phase and the number of links added at each time step in the BA model are set as $m_0 = m = 2$ and 1,997 links are generated. This figure shows that the distribution of node degrees of the random network approximately follows a Poisson distribution. On the other hand, distribution of the degrees of the power-law network is approximately aligned on a log-log plot, which indicates the distribution follows the power-law. Distributions of distances between nodes in the random network and the power-law network are shown in Figure 3.3. The horizontal axis represents distance; we mean distance is number of hops between a pair of nodes. The vertical axis represents frequency of node pairs whose distances are h. The variance of the distances in the random network is larger than that in the power-law network. In addition, the average distance of the power-law network is smaller than that of the random network due to the existence of hub nodes.

3.1.4 Performances of Random and Power-Law Networks

If the physical topology of a WDM network is power-law, a large variance of node degrees strongly affects the performance of the network, such as its blocking probability. In this subsection, we investigate the performances of blocking probability in random and power-law WDM networks.



(a) 16 wavelengths



(b) 32 wavelengths

Figure 3.4: Blocking probabilities in random and power-law networks

We measured the blocking probabilities of lightpath establishment by computer simulations with the topologies which we use for the comparisons of properties in the previous

subsection. In addition, we assume the following conditions and restrictions:

- The number of physical links between a pair of two adjacent nodes is one.
- Each link is a bi-directional (i.e., it is composed of an incoming and an outgoing fibers).
- Propagation delays of the fibers are uniformly 0.1 msec.
- Processing delays at the nodes are ignored.
- Arrival of demands between all of the node pairs follows a Poisson process with an average rate λ.
- Holding time of the lightpaths follows an exponential distribution with an average rate of $1/\mu$.
- The shortest-hop routes are used for routes of lightpaths.
- Wavelengths are assigned by the backward reservation protocol [28].
- Wavelength conversion is not available at any node.

Figure 3.4 shows the results of simulations with 16 and 32 multiplexed wavelengths. The horizontal axes represent arrival rate. The vertical axes represent blocking probability. λ is changed from 0.1 requests/msec to 4.0 requests/msec and μ is set to 1.0 per second, i.e., average holding time is 1.0 sec. From these results, it is found that power-law networks cannot accommodate still less traffic demands than random networks when the traffic load is not light. This is because many requests compete for wavelength resources around hub (i.e., high-degree) nodes. To see this more clearly, we measure load L_e on link e ($\in E$). L_e means the number of node pairs whose lightpaths go through link e and given by Eq. (3.4). V is a set of the nodes in a network and E is a set of the links. $\pi_{i,j}$ is a set of the links included in the route of lightpaths from node i to node j. x_e is defined as Eq. (3.5).

$$L_e = \sum_{i,j \in V, i \neq j} x_e(\pi_{i,j}). \tag{3.4}$$

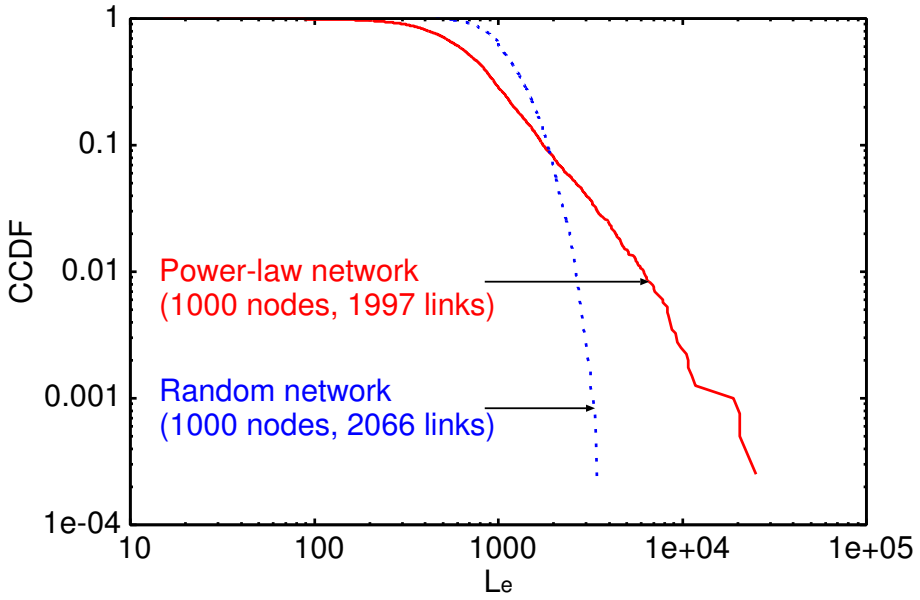


Figure 3.5: Complementary cumulative distributions of link loads in topologies generated with the ER and BA models

$$x_e(\pi) = \begin{cases} 1 & (e \in \pi) \\ 0 & (otherwise) \end{cases} . \tag{3.5}$$

We show the complementary cumulative distributions of L_e for a power-law network and a random network in Figure 3.5. From this figure, the link load distributions show much the same tendency to the node degree distributions; the link load distribution for a power-law network has heavy tail. That is, there are some heavy-loaded links in power-law networks and they increase the blocking probability. Based on the observations, in the following sections, we propose a new method to setup lightpaths more efficiently in power-law networks.

3.2 Virtual Fiber Configuration

In Section 3.2, we showed that the power-law connectivity of physical topologies in WDM networks increases blocking probabilities. The topological property leads most of the shortest path routes between the nodes to pass across hub nodes, and therefore reservation requests conflict at hub nodes. In this section, we consider some approaches to moderate load concentration at hub nodes and introduce a solution using *virtual fiber*.

3.2.1 Approaches to Moderate Load Concentration

There might be some solutions to the load concentration in power-law networks. Here we pick up and discuss two main solutions. After that, we bring on our approach.

Enhancement of Network Equipments

The simplest solution is enhancement of network equipments, i.e., installing more OXCs and fibers into heavily loaded parts in a network or upgrading those equipments. By adopting this, the amount of traffic that a network can accommodate is increased and, as a result, blocking probabilities for the network is improved. However, installing or upgrading network equipments requires much more investment. From a viewpoint of cost, we think it is difficult to moderate the load concentration by only this approach.

Link State Based Routings

Using link state based routings is a second solution. By utilizing link-state based routing, routes of lightpaths are diverted from heavily loaded links. Consequently, we realize load distribution in a network and decrease the blocking probability. But this kind of routing strategy requires newest link state information to perform well. Hence, we must frequently distribute link state information about all of the links and update a routing table at each node. It means that we have to prepare extra capacity for distributing link state information and that each node has overhead to calculate contemporary routes. This penalty is undesirable in particular for large-scale networks.

Changing the Connectivity Logically

To improve blocking probabilities of power-law networks without equipment investment and routing overhead, we consider another approach. In our approach, we logically change a topology having the power-law connectivity into another one that is more similar to a random network. How to change topologies logically is described in the following subsections.

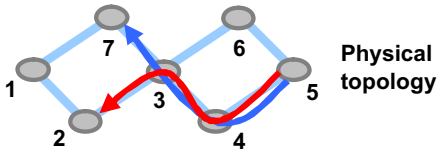
3.2.2 Concept of Quasi-Static Lightpath

In dynamic-wavelength routing networks, lightpaths are established on a demand basis and released after data transmission. However, the more hops (fibers) that lightpaths pass through, the more difficult setup becomes because of the inherent nature of a circuit-switch-based network (i.e., the lightpath with more hops requires more wavelength resources), and this is exacerbated by the wavelength continuity constraint.

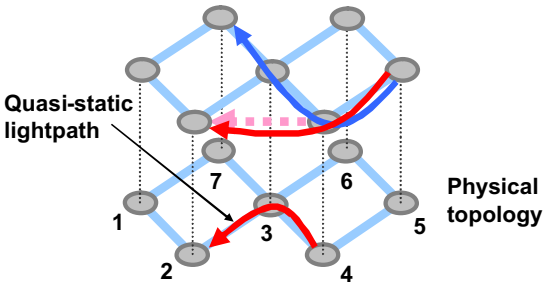
To resolve the inequality of blocking probabilities between short-distance and long-distance node pairs, we prepared some lightpaths beforehand. We refer to such pre-configured lightpaths as quasi-static lightpaths. Quasi-static lightpaths are different from conventional static lightpaths designed for transporting IP packets or communications of other upper layers. Quasi-static lightpaths are reserved as parts of lightpaths. Those lightpaths are released after data transmission, but quasi-static lightpaths keep their configurations. Quasi-static lightpaths may not be reconfigured unless the traffic pattern is substantially changed. In this sense, the pre-configured lightpaths are quasi-static.

Figure 3.6(b) illustrates the concept of quasi-static lightpath. In traditional wavelength routing networks, lightpaths are routed and set up in physical topologies composed of nodes and fibers, as shown in Figure 3.6(a). On the other hand, quasi-static lightpath behaves as a single hop link to upper wavelength routed networks. That is, wavelength routing protocols perceive quasi-static lightpaths as fibers whose available wavelengths are only those reserved for the quasi-lightpaths. Then a virtual optical network is constructed over a physical topology as shown in Figure 3.6(b) (the dotted line is a logical link by a quasi-static lightpath). In the situation of Figure 3.6(a), when a lightpath from node 5 to node 2 is requested, we have to reserve a same wavelength in the three links, $5 \rightarrow 4$, $4 \rightarrow 3$, and $3 \rightarrow 2$ to establish it. However, in the case of Figure 3.6(b), we have to reserve a same wavelength in only two links, $5 \rightarrow 4$ and $4 \rightarrow 2$, due to a logical link by a quasi-static lightpath.

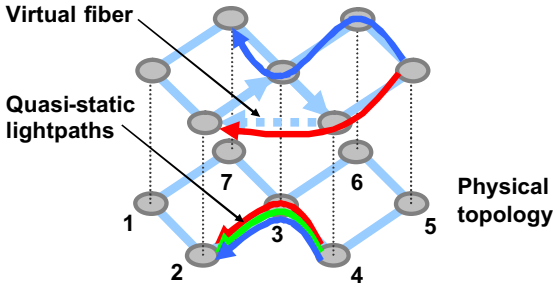
There are two benefits of quasi-static lightpaths. First, the fragmentation of wavelength resources can be avoided by setting up quasi-static lightpaths. When a network is congested, the remaining free wavelength resources are too fragmented to be utilized to establish lightpaths due to the wavelength continuity constraint. However, the constraint



(a) Physical topology



(b) Virtual optical network construction with a quasi-static lightpath



(c) Virtual optical network construction with a virtual fiber

Figure 3.6: Virtual optical network construction (The links without arrows are bi-directional.)

is always satisfied at the parts consisting of quasi-static lightpaths. Therefore, quasi-static lightpaths can promote an effective utilization of resources. Second, quasi-static lightpaths shorten the distance between nodes. Viewing from the upper layer, the source node of a quasi-static lightpath is directly connected to the destination nodes of the quasi-static lightpath, which reduces the number of hop-counts between nodes.

However, quasi-static lightpath configuration has a disadvantage that it makes vulnerable parts to traffic load in a network. Each of logical links by quasi-static lightpaths has only one available wavelength while fibers, which devote their wavelengths to quasi-static lightpaths, lose some of available wavelengths. In the case as in Figure 3.6(b), link $4 \rightarrow 2$ can accommodate only one request at a time. At the same time, links $4 \rightarrow 3$ and $3 \rightarrow 2$ has only $w - 1$ available wavelengths, assuming that the total number of wavelengths is

w; those links can accommodate only $w - 1$ requests at a time. Thus, for effectively utilizing the quasi-static lightpaths, it is necessary to append traffic engineering mechanisms to a routing architecture, i.e., we have to use link state based routings. Therefore, we use *virtual fibers* instead of quasi-static lightpaths in order to construct logical topologies.

3.2.3 Virtual Fiber: Bundle of Quasi-Static Lightpaths

Virtual fiber is equivalent to a bundle of quasi-static lightpaths for all of the wavelengths. Virtual fiber configuration is illustrated in Figure 3.6(c). Quasi-static lightpaths from node 4 to node 2 via node 3 are configured for all of the wavelengths (here we assume $w = 3$). We regard a set of these quasi-static lightpaths as a fiber in a virtual optical network. We call this operation *cut-through* hereafter. Then, node 4 gets a fiber to node 2, which has w available wavelengths. Instead of the virtual fiber, node 4 loses a fiber to node 3 in the virtual optical network and node 2 loses a fiber from node 3. As for node 3, it loses an incoming fiber and an outgoing fiber. That is, the degrees of intermediate nodes of a virtual fiber are reduced by one for each.

Since fibers reserved for virtual fibers vanish in a virtual optical network, some node pairs have to change routes of lightpaths. In Figure 3.6(a) and Figure 3.6(b), the route of a lightpath from node 5 to node 7 is $5 \rightarrow 4 \rightarrow 3 \rightarrow 7$. But, in Figure 3.6(c), fiber $4 \rightarrow 3$ disappears in the virtual optical network. The route of a lightpath from node 5 to node 7 is changed to $5 \rightarrow 6 \rightarrow 3 \rightarrow 7$ in that case. Although this seems a demerit of virtual fiber configuration at first glance, it is useful for load distribution. For example, in Figure 3.6(a) and Figure 3.6(b), fiber $4 \rightarrow 3$ is heavily loaded. However, this load concentration is moderated by a cut-through operation as in Figure 3.6(c).

3.3 Configuration Methods of Virtual Optical Networks

In this section, we propose two types of configuration methods of virtual optical networks, degree based and load based. The basic strategy of our method is reducing degrees of heavily loaded nodes, which would mainly be hub nodes, by cut-through operations and bypassing some of the heavy traffic. In degree based method, we regard high degree nodes

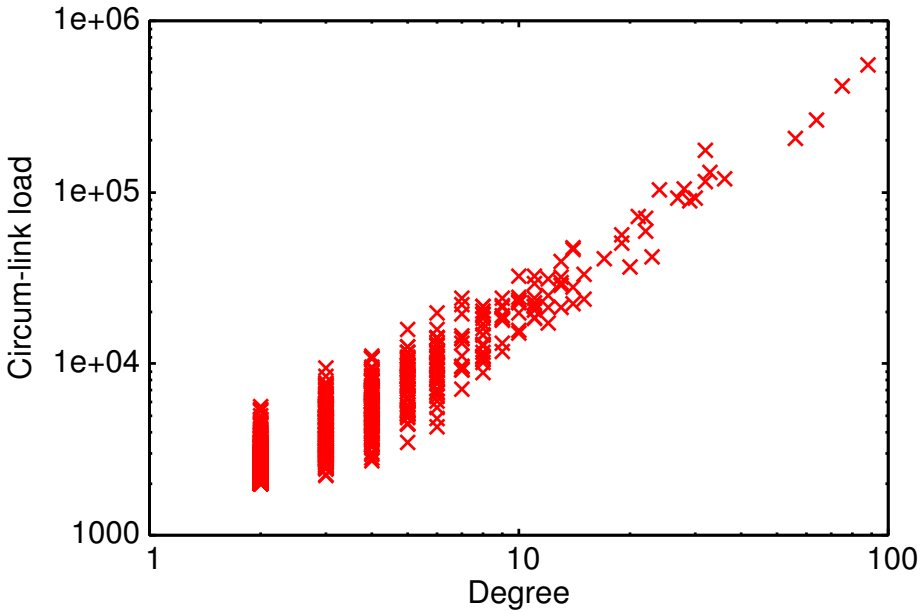


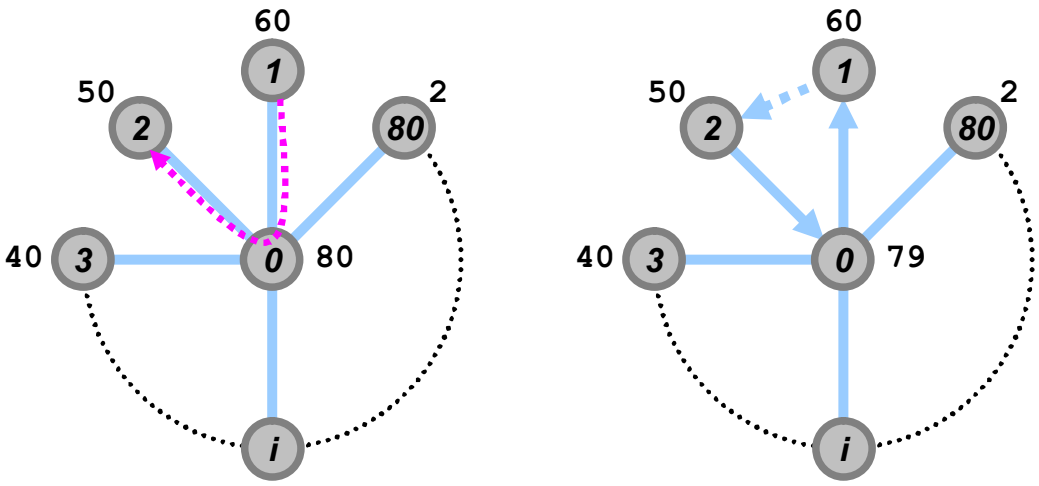
Figure 3.7: Correlation between degree and circum-link load

as heavily loaded nodes since shortest paths between nodes tend to pass through hub nodes in power-law networks. In load based method, we utilize circum-link load as the metric. Circum-link load of a node i, c_i defined by Eq. (3.6), means the sum of loads on the links connected with a node i:

$$c_i = \sum_{(i \to j) \in E} L_{(i \to j)} + \sum_{(j \to i) \in E} L_{(j \to i)}. \tag{3.6}$$

To configure virtual fibers efficiently, circum-link load is more suitable for the metric than node degree because the purpose of virtual fiber configuration is load distribution. But circum-link load is so variable by change of traffic pattern that we have to recalculate it after every cut-through operation. On the other hand, degree is independent of traffic pattern and correlated closely with circum-link load as shown in Figure 3.7. Thus, we consider degree based and load based methods.

We explain the outline of our methods with an instance illustrated in Figure 3.8. Here we use degree as the metric. Figure 3.8(a) shows a hub node 0 and its adjacent nodes 1 to 80 in a power-law network (the other nodes and links are omitted here). The numbers described beside nodes are degree. Supposed that the degree of node 0 is maximum in the network. It is reasonable to expect that larger amount of traffic is transmitted through



(a) Physical connectivity of a hub node

(b) Cut-through a hub node (A dotted arrow is a virtual fiber.)

Figure 3.8: Virtual fiber configuration around a hub node (Numbers described beside nodes are degree.)

higher degree (i.e., more heavily loaded) nodes. If so, for node 0, incoming traffic from node 1 is heavier than those from the other adjacency nodes. In the same way, for node 0, outgoing traffic to node 2 is heaviest among those to the other adjacency nodes except to node 1. Then we prepare a virtual fiber from node 1 to node 2 via node 0 and construct a virtual optical network like Figure 3.8(b). This cut-through operation splits traffic through node 0 into two factions, i.e., traffic from node 1 to node 2 and the other by the virtual fiber. Additionally, this operation diverts traffic from node 1 to the other adjacent nodes and from the other adjacent nodes to node 2 from node 0, as a result. Thus, load on node 0 is reduced and distributed.

Our proposed method repeats the above heuristic process. The details of our method are described below.

3.3.1 Notations

We use the following notations to explain our method.

N: Set of the nodes in a network.

F: Set of the fibers in a network, including the virtual fibers.

F_{n_1,n_2}: Set of the fibers placed from a node n_1 to a node n_2 in a virtual optical network.

A^n_{in}: Set of the adjacent nodes, which are connected to a node n.

A^n_{out}: Set of the adjacent nodes, which are connected from a node n.

$Cut(f_1, f_2)$: Cut-through operation from a fiber f_1 to a fiber f_2.

d_n: Degree of a node $n \in N$.

c_n: Circum-link load of a node $n \in N$.

3.3.2 Degree Based Method

Step 1: Set the value of th such that $th > 2$. Go to Step 2.

Step 2: If $max\ d_n > th$ ($n \in N$), then $n_0 \leftarrow n$ and go to Step 3. Otherwise, go to Step 5.

Step 3: Search such a node pair (n_{in}, n_{out}) that $d_{n_{in}} + d_{n_{out}}$ is maximum where $n_{in} \in A^{n_0}_{in}$, $n_{out} \in A^{n_0}_{out}$, $n_{in} \neq n_{out}$, and $F_{n_{in},n_{out}} = \phi$. If it is found, ($n_1$, n_2) $\leftarrow$ (n_{in}, n_{out}) and go to Step 4. Otherwise, go to Step 5.

Step 4: $Cut(f_1, f_2)$ ($f_1 \in F_{n_1,n_0}, f_2 \in F_{n_0,n_2}$). go back to Step 2.

Step 5: Quit virtual fiber configuration.

In Step 1, we set the threshold th to determine a terminal condition. If the maximum degree is equal to or less than th, this method quit configuring virtual fibers. This evaluation is done in Step 2. The floor of th is two because, if $th = 1$, a generated virtual optical network is an uni-directed cycle graph. In Step 3, we select edge nodes of a virtual fiber via node n_0, n_1 and n_2, by the heuristic approach described above. Note that node 1 and node 2 must not be connected by a physical or virtual fiber yet at this point. This is because, if there is already a direct link between node 1 and node 2, a virtual fiber configured from

node 1 to node 2 would have almost no effect for load distribution and be just waste of wavelength resources. If a node pair (n_1, n_2) satisfying the restriction is found, we operate cut-through from node n_1 to node n_2 via node n_0 and iterate a same process from Step 2. Thus, the maximum degree in a network is decreased by one every iteration.

3.3.3 Load Based Method

The algorithm of this method is almost the same as degree based method. Only one difference is that the metric is replaced with normalized circum-link load by the number of node pairs, $\tilde{c}_i = c_i/|N|(|N|-1)$, where $|N|$ is the number of nodes in a network. The restriction against the threshold th in degree based method is removed.

Step 1: Set the value of th. Go to Step 2.

Step 2: If $max\ \tilde{c}_n > th$ ($n \in N$), then $n_0 \leftarrow n$ and go to Step 3. Otherwise, go to Step 5.

Step 3: Search such a node pair (n_{in}, n_{out}) that $\tilde{c}_{n_{in}} + \tilde{c}_{n_{out}}$ is maximum where $n_{in} \in A_{in}^{n_0}$, $n_{out} \in A_{out}^{n_0}$, $n_{in} \neq n_{out}$, and $F_{n_{in},n_{out}} = \phi$. If it is found, ($n_1$, n_2) $\leftarrow$ (n_{in}, n_{out}) and go to Step 4. Otherwise, go to Step 5.

Step 4: $Cut(f_1, f_2)$ ($f_1 \in F_{n_1,n_0}, f_2 \in F_{n_0,n_2}$). go back to Step 2.

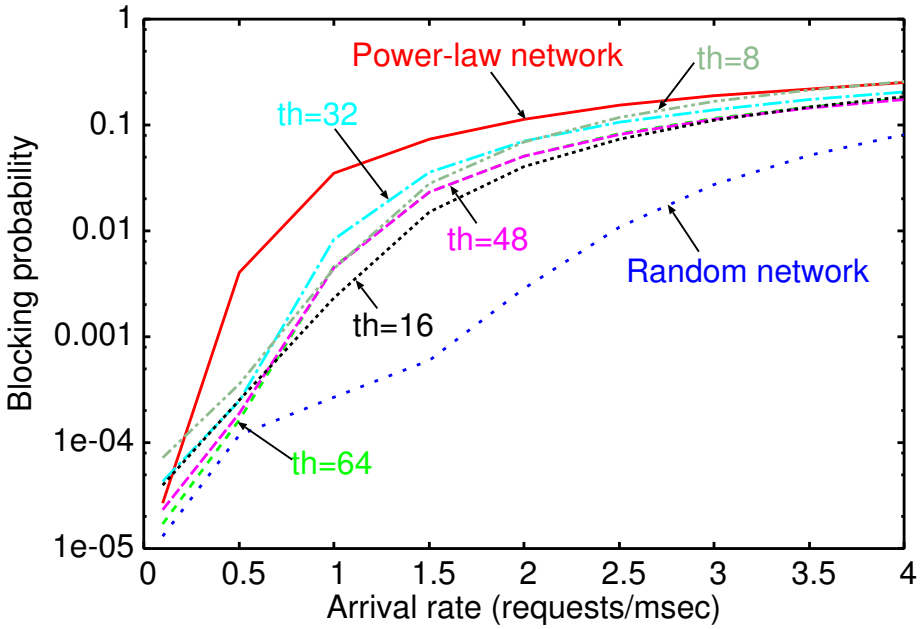
Step 5: Quit virtual fiber configuration.

3.4 Numerical Evaluation

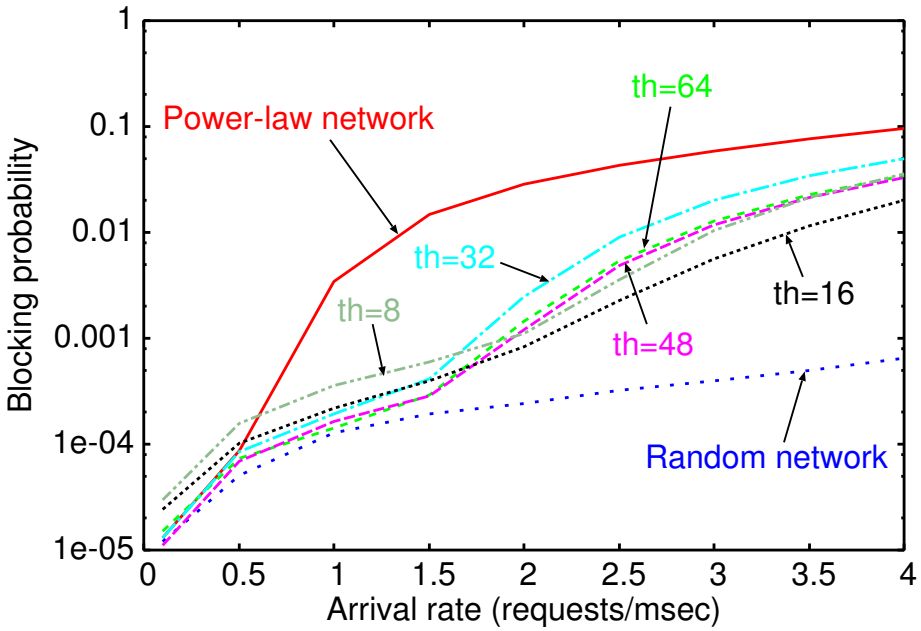
We evaluated the performances of degree based and load based virtual fiber configuration methods with the same simulation model in Section 3.1.

3.4.1 Performance of Degree Based Method

The results of degree based virtual fiber configuration method are illustrated in Figure 3.9. The maximum degree of the power-law network topology is 88. The degree thresholds we examined are 64, 48, 32, 16, and 8. Our proposed method reduces more than one



(a) 16 wavelengths



(b) 32 wavelengths

Figure 3.9: Variation of blocking probabilities for different thresholds th in power-law networks

order of magnitude of the blocking probability when the arrival rate is moderate. For lower arrival rates, our method performs best when th is 48 or 64. The performances for these

Table 3.1: Average distance, average/maximum/minimum link load, and number of links of logical topologies generated by degree based virtual fiber configurations (a bi-directional link is counted as two uni-directional links.)

Topology	Ave. distance	Ave. L_e	Max. L_e	Min. L_e	Number of links
Power-law network	3.99	998.89	25120	15	3994
$th = 64$	4.15	1046.0	12905	48	3959
$th = 48$	4.33	1107.1	11863	62	3903
$th = 32$	4.47	1166.0	11786	55	3834
$th = 16$	5.09	1406.9	9993	117	3613
$th = 8$	5.92	1787.1	8745	325	3314
Random network	5.06	1222.5	3442	414	4132

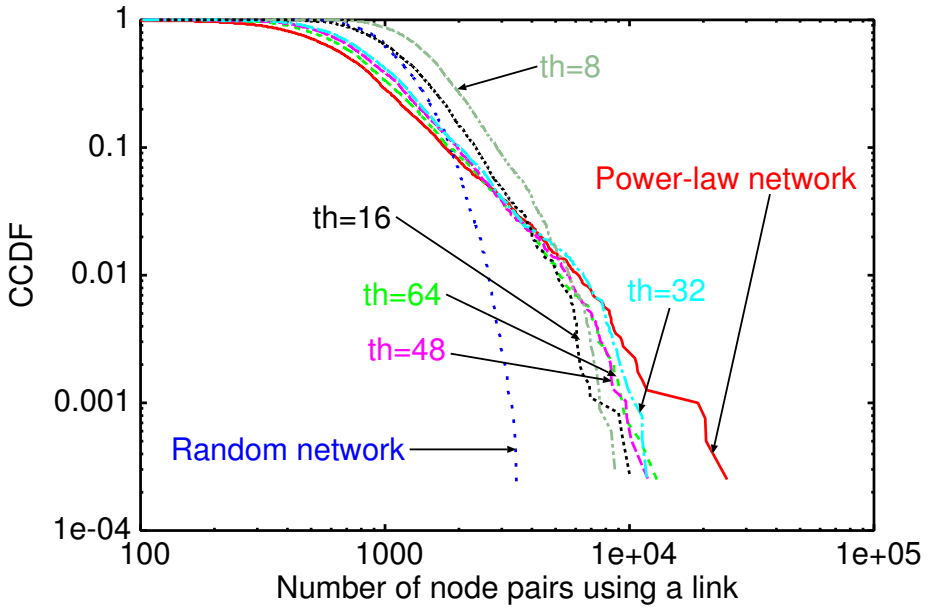


Figure 3.10: Complementary cumulative distributions of link loads distances between nodes on logical topologies

thresholds are almost same all through the arrival rate. For higher arrival rates, $th = 16$ lets the degree based method perform better than the other thresholds. An optimal degree threshold depends on arrival rate.

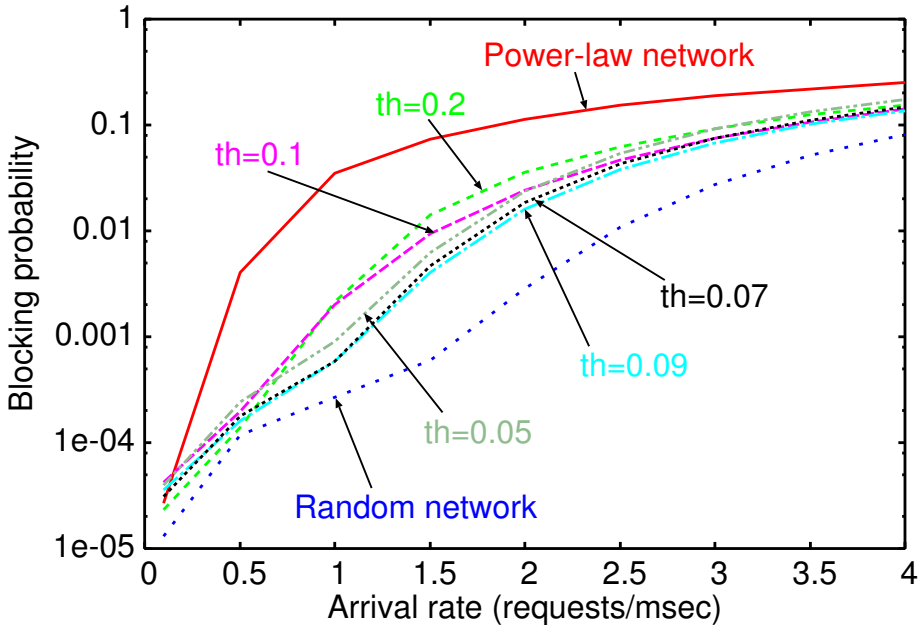
This fact is explained by the changes of distance and link load distribution. Table 3.1 shows average distance, average/maximum/minimum link load L_e, and number of links for each threshold. Note that a bi-directional link is counted as two uni-directional links here. Difference of the number of links in the power-law network, 3994, and a number of links in

a virtual optical network is the number of cut-through operations. Cut-through operations reduce maximum or higher load. However, the reduction requires sacrifices from average load and average distance. This is because those operations logically decrease the number of links and make shortest paths through hub nodes unusable, i.e., link utilization becomes easier to be increased. When arrival rate of requests is low, blocking probability is much affected by distance rather than by link load L_e since offered load is also low. On the other hand, when arrival rate is high, link load, especially for maximum link load, affects the blocking probability: The higher link load L_e is, the more offered load for link e is likely to be increased. Therefore, the degree based method with $th = 16$ shows best performance for the moderate and high arrival rate while it performs better with $th = 64$ or $th = 48$ when the arrival rate is low.

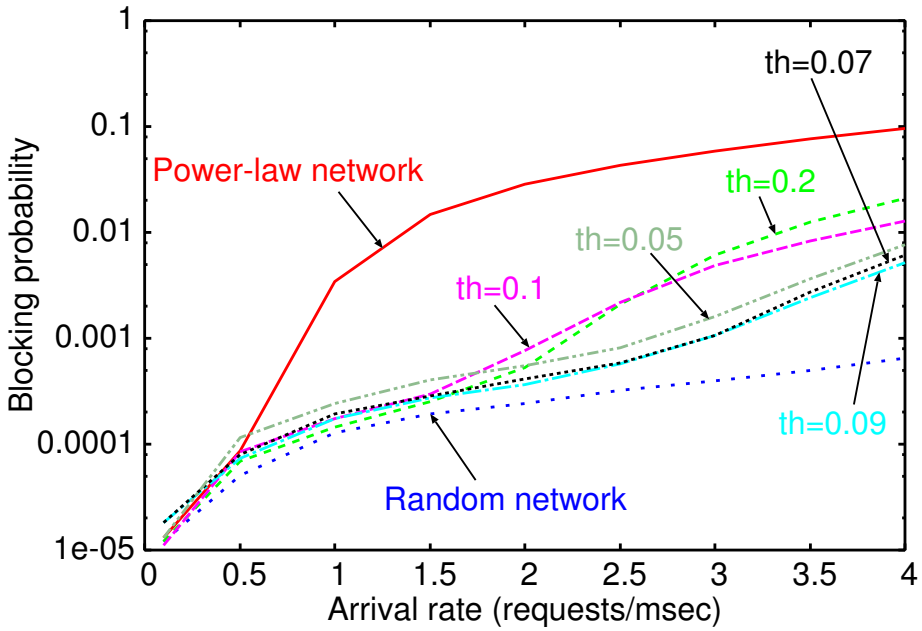
When th is 8, since the distances between the nodes become longer than any other topologies due to excess operations of cut-through, blocking probability with $th = 8$ increases when comparing to the results with $th = 16$. In Table 3.1, it seems that there is little difference between $th = 48$ and $th = 32$. But as in Figure 3.9, $th = 32$ is not a good threshold. Figure 3.10 shows the L_e distribution of each threshold. From this figure, although the maximum link load is decreased by reducing the degree threshold from 64 or 48 to 32, the frequency of heavily loaded nodes is higher. We consider this is because some links connected to nodes around hub nodes are overloaded. By configuring virtual fibers, routes between some node pairs passing through hub nodes are diverted and load originally for links connected to hub nodes is distributed. However, when th is 32, this load distribution does not work well and diverted routes from hub nodes tend to pass through certain links. The degree based virtual fiber configuration method decides where to be cut-through with only node degree information. But, instead of the simplicity and heuristics, it sometimes carries out unprofitable configurations. The other proposed method, load based virtual fiber configuration method revises this defect.

3.4.2 Performance of Load Based Method

We examined the performance of the load based virtual fiber configuration method when the normalized load threshold is 0.2, 0.1, 0.09, 0.07, and 0.05. The maximum normalized



(a) 16 wavelengths



(b) 32 wavelengths

Figure 3.11: Variation of blocking probabilities for different thresholds th in power-law networks

circum-link load of the power-law network is 0.552384. The simulation results are illustrated in Figure 3.11. This method performs better than the degree based method when

Table 3.2: Maximum degree, average distance, average/maximum/minimum link load, and number of links of logical topologies generated by load based virtual fiber configurations (a bi-directional link is counted as two uni-directional links.)

Topology	Max. deg.	Ave. dist.	Ave. L_e	Max. L_e	Min. L_e	Num. of links
$th = 0.2$	59	4.24	1075.7	10182	53	3934
$th = 0.1$	35	4.66	1229.1	10893	86	3784
$th = 0.09$	30	4.75	1264.8	7315	40	3750
$th = 0.07$	24	4.95	1345.3	6491	56	3673
$th = 0.05$	16	5.38	1532.0	6281	104	3510

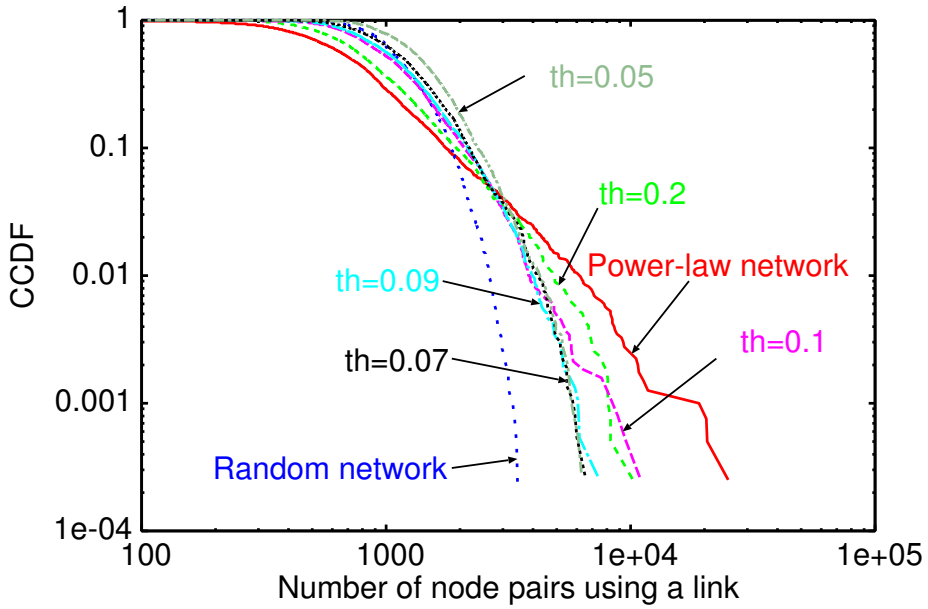


Figure 3.12: Complementary cumulative distributions of link loads distances between nodes on logical topologies

arrival rate is moderate or high. The results of $th = 0.09$ and $th = 0.07$ are similar all through the arrival rate. When the threshold is higher, i.e., 0.2 or 0.1, the blocking probability is worsen at high arrival rates. This is because the number of configured virtual fibers is not enough. Compared to the cases with the other thresholds, heavily loaded links still remain in a virtual optical network as illustrated in Figure 3.12.

The maximum link load of the virtual optical network generated with load threshold 0.1 is higher than that of the logical topology generated with load threshold 0.2. Hence, the former topology is more sensitive to increase of arrival rate and its blocking probability gets higher than the blocking probability of the latter topology at moderate arrival rate. But this

relation turns back at high arrival rate since link load is totally more well-balanced when th is 0.1. Relatively heavily loaded links (the middle of L_e distribution in Figure 3.12) also affect increase of blocking probability at high arrival rate because offered load for those links becomes high. On the other hand, when the threshold has a lower value, 0.5, too many virtual fibers are configured and the blocking probability is also slightly increased. The performance of load based method is stable when th is about 0.09. It means that we do not need to reconfigure a virtual optical network so much according to the change of arrival rate of lightpath setup requests.

To compare the efficiency of our two methods, we list maximum degree, average distance, average/maximum/minimum link load, and number of links for each load threshold in Table 3.2. Focusing on the load threshold is 0.09, the load based method reduces the maximum link load lower than the degree based method with less number of cut-through operations. In addition, the load based method keeps average distance and average link load lower not only than the degree based method with $th = 8$ but with $th = 16$. Thus, the load based method achieves better performance when arrival rate is high.

3.5 Summary

According to the trend of technological development of optical networks, large-scale optical networks will be constructed by interconnecting a number of local optical networks in the future. There is a possibility that topologies of such large-scale optical networks exhibit the power-law attributes rather than the properties of random networks. However, in traditional studies on WDM-based networks, the objective physical topologies are not large and rely on random networks. We investigated the performance of large-scaled WDM networks whose topologies follows the power-law. The results show that high-degree nodes in the power-law networks are easy to be congested and that the congestion at those nodes causes the decline of performance of blocking probability. To resolve this problem, we proposed two virtual fiber configuration methods to accelerate the performance of WDM networks with physical topologies following the power-law. We evaluated our method by simulation and confirmed that our proposed method is efficient for power-law networks to improve the blocking probability.

For future research work, we plan to consider the way to determine thresholds of maximum degree or maximum link load L_e. One possible candidate is to use the results of analyzing the structural properties of the topologies exhibiting the power-low attributes.

Chapter 4

Performance Analysis of Signaling State Managements

RSVP-TE is a signaling protocol to setup and teardown lightpaths in wavelength-routed GMPLS networks. RSVP-TE uses the soft-state control mechanism to manage lightpaths. In the soft-state control mechanism, each node sets a timer for each control state and resets the timer with refresh messages to maintain the state. When the timer expires due to losses of refresh messages, the control state is initialized and a reserved resource managed with the state is released. It has been considered that resource utilization of soft-state protocols is inferior to that of hard-state protocols since soft-state protocols may reserve resources until control states are deleted due to timeout. Therefore, some extensions to promote the performance of soft-state protocols, such as message retransmission, have been considered. In this chapter, we analyze the behavior of GMPLS RSVP-TE and its variants with a Markov model and analyze the performance of RSVP-TE. From the results, we demonstrate that resource utilization of RSVP-TE can be equivalent to that of a hard-state protocol when the loss probability of signaling messages is low. We also investigate the effectiveness of message retransmission and show that using message retransmission leads to poor resource utilization in some cases.

Table 4.1: Types of RSVP-TE control messages

Type	Role of message
Path	request for a LSP session
Resv	reserves a label
PathErr	notifies an error relating to Path state
ResvErr	notifies an error relating to Resv state
PathTear	removes a Path state
ResvTear	removes a Resv state
ResvConf	confirms the LSP establishment

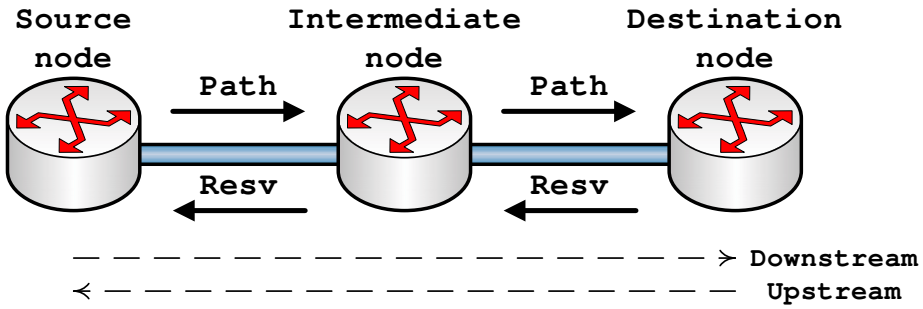


Figure 4.1: LSP establishment by RSVP-TE

4.1 GMPLS RSVP-TE

GMPLS is the standard technology to configure lightpaths in wavelength-routed networks. In GMPLS, wavelengths are regarded as labels and lightpaths are called LSPs (Label Switched Paths). RSVP-TE is a signaling protocol for managing LSPs. In this section, we briefly review RSVP-TE.

4.1.1 Signaling Process of GMPLS RSVP-TE

RSVP-TE has seven types of signaling messages: Path, Resv, PathErr, ResvErr, PathTear, ResvTear, and ResvConf as listed in Table 4.1. Figure 4.1 illustrates LSP establishment by RSVP-TE, where each control signal is sent hop-by-hop. When an LSP request arrives at a source node, the source node creates a Path trigger message and sends it downstream. Each intermediate node that receives the Path trigger message makes a Path state in itself and it also checks information about available labels in the Path trigger message. If there is an available label on the outgoing link, the node forwards the message downstream. Otherwise, a PathErr message is created and sent back toward the source node. When the Path

trigger message arrives at a destination node, the node makes a Path state. If there is one or more available labels, the destination node selects a label from available labels listed in the received Path trigger message and reserves the label. Then, a Resv trigger message that includes the selected label is created and sent upstream. If there is no available label, the destination node sends a PathErr message upstream. Each intermediate node that receives the Resv trigger message reserves the label specified in the message and makes a Resv state. After that, the node selects a label to be reserved by its upstream node[1] and forwards the Resv trigger message upstream. If an intermediate node fails to reserve a label due to a lack of available labels, the node creates a ResvErr message and sends it downstream. If the source node successfully receives the Resv trigger message, it means that an LSP is established. If the destination node requests confirmation of LSP establishment, the source node sends a ResvConf message toward the destination node. After data transmission is completed, the source node sends a PathTear message downstream. Intermediate nodes that receive the PathTear message delete their Path and Resv states and forward the message downstream.

4.1.2 State Control at Nodes

As mentioned above, nodes create a Path and a Resv state for each LSP. In soft-state control, these states are maintained by refreshing them during data transmission. Furthermore, when nodes create control states, they also set state timeout timers to manage lifetimes of control states. If a state timeout timer expires, a corresponding control state is removed and a reserved label is released. Lifetimes of control states are prolonged and state timeout timers are reset if refresh messages arrive before state timeouts. When a node sends a Path or a Resv trigger message, it also sets a refresh timer, and every time a refresh timer expires, a refresh message is sent and the timer is reset. In RSVP-TE, signaling messages are sent in best-effort unless the message retransmission extension [57] is used. Lifetimes of states are typically longer than refresh intervals so as to send some refresh messages by state timeouts. On the other hand, since hard-state signaling does not have the refresh mechanism, message retransmission is necessary to deliver signaling messages to receiver

[1] If the wavelength selection is subject to the wavelength continuity constraint, the same label is selected.

nodes.

Loss of a PathTear message in the standard RSVP-TE requires so much as a state lifetime in order to release a reserved label. Therefore, RSVP-TE would make the resource utilization lower than by hard-state signaling. Although short lifetimes of control states may improve the resource utilization of RSVP-TE, refresh intervals also become short at the same time, which increases the number of signaling messages. If several losses of refresh messages occur, corresponding control states are removed incorrectly (false removal). Although frequent refreshing suppresses false removals, the number of signaling messages also increases.

However, RSVP-TE is tolerant to failures on the control plane. Control states would, therefore, be initialized by state timeout while control channels are down due to network failures. Hard-state signaling cannot update or delete control states during such failures on the control plane.

4.2 Modeling and Analysis of GMPLS RSVP-TE for Single-Hop LSP

In this section, we investigate the steady-state performance of GMPLS RSVP-TE for single-hop LSP. We develop a model of GMPLS RSVP-TE based on the Markov model in [51] and use it to analyze the performance of GMPLS RSVP-TE. We consider two types of RSVP-TE: the standard RSVP-TE (we call this RSVP-TE hereafter) and RSVP-TE with the extension of the message retransmission (RSVP-TE/Ack). As opposed to the model in [51], our model incorporates RSVP-TE that has the control state for backward direction, i.e., Resv state. We also extend the state transition of the control plane failure and recovery into the model to show how GMPLS RSVP-TE is stable during disruption of the communications on the control plane.

4.2.1 Model of GMPLS RSVP-TE for Single-Hop LSP

First, we consider the model of GMPLS RSVP-TE without control plane failure. We assume the following in order to develop our models with the Markov chain.

- Arrivals of LSP setup requests follow a Poisson process with rate λ_r.
- Connection time of LSPs follows an exponential distribution with rate μ.
- Message processing delay at nodes is 0.
- Propagation delay per hop of signaling messages follows an exponential distribution with rate $1/D$.
- Blocking probability of label reservation per hop, p_b, is constant.
- Signaling message loss probability per hop, p_l, is constant for an LSP.
- Any incoming wavelength can be converted to any outgoing wavelength.

We also assume the items below for the control parameters and the message processing of RSVP-TE.

- Refresh intervals follow an exponential distribution with rate $1/T$ regardless of sender nodes and message types.
- Lifetimes of control states X are given as T multiplied by k, i.e., $X = kT$, where k is a constant number of refresh events.
- Retransmission intervals follow an exponential distribution with rate $1/R$ regardless of the sender node and message type.
- The maximum number of retransmission times m is constant.
- Error messages are not lost.
- Acknowledgments of message receipt are not lost.

Now we focus on the steady-state behavior of GMPLS RSVP-TE for an LSP. Although we assume that the time parameters, propagation delay, refresh interval, state lifetime, and retransmission interval follow exponential distributions, the average performance of GMPLS RSVP-TE is decided from the average values of those parameters, i.e., D, T, X, and R. Hence, these assumptions do not affect to the results we want. Constant blocking

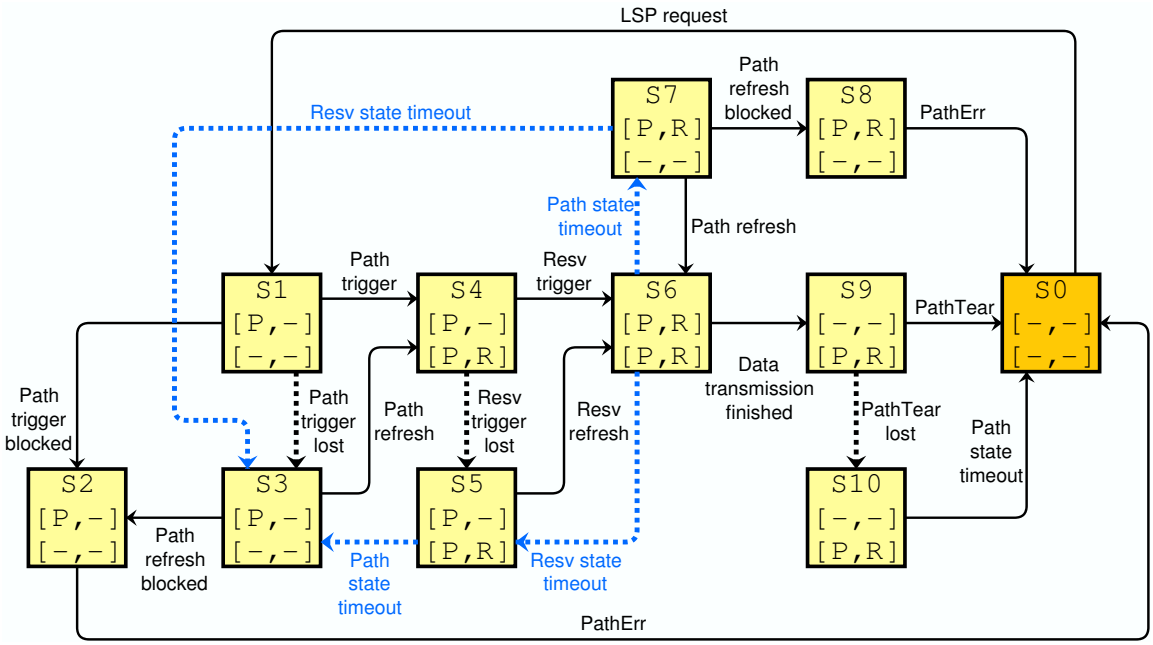


Figure 4.2: State transition of RSVP-TE for a single-hop LSP

probability and the random loss model of signaling messages are also reasonable for the same reason that we are paying attention to the steady state. Note that it is assumed that losses of signaling messages occur only due to the buffer overflow in the receive buffer at nodes where multiple LSP sessions traverses. We therefore assume here that the signaling messages for an LSP are randomly dropped. The case for the buffer overflow will be considered in Section 4.4.

Figure 4.2 shows the state transition of RSVP-TE for a single-hop LSP. This state transition consists of 11 states: S_i $(i = 0, 1, \cdots, 10)$. Each square represents a state of the state transition and has a 2×2 matrix. The first row of the matrix has the status of a source node, and the second row has the status of a destination node. A "P" in the left column of a state indicates that there is a Path state. Similarly, "R" in the right column indicates that there is a Resv state. If there is no control state (i.e., a default state), it is indicated as "$-$." We explain the operations of RSVP–TE at S_i below.

S_0: The initial state. When an LSP setup request arrives at a source node, the Markov chain transits to S_1.

S_1: The source node creates a Path state and sends a Path trigger message. If the message is lost on the way from the source node to a destination node, the Markov chain

transits to S_3. If the destination node successfully receives the message and if there is an available label, the Markov chain transits to S_4. If a destination node receives the message but there is no available label, the Markov chain transits to S_2.

S_2: The destination node sends a PathErr message. The Markov chain transits to S_0.

S_3: The source node sends a Path refresh message. If the destination node receives the message and there is an available label, the Markov chain transits to S_4. If the destination node receives the message and there is no available label, the Markov chain transits to S_2.

S_4: The destination node creates a Path state. The destination node also makes a Resv state and sends a Resv trigger message. If the source node receives the Resv trigger message, the Markov chain transits to S_6. Otherwise, the Markov chain transits to S_5.

S_5: The destination node sends a Resv refresh message. If the source node receives the Resv refresh message, the Markov chain transits to S_6. If a false removal occurs at the destination node because of the successive loss of refresh messages, the Markov chain transits to S_3.

S_6: In this state, the source node is transmitting data by the established LSP. If the data transmission is successfully completed, the Markov chain transits to S_9. If a false removal of either the Resv state at the source node or the Path state at the destination occurs, the Markov chain transits to S_5 or S_7, respectively.

S_7: If the destination node receives a Path refresh message and there is an available label, the Markov chain transits to S_6. If the destination node receives a Path refresh message and there is no available label, the Markov chain transits to S_8. If a false removal occurs at the source node, the Markov chain transits to S_3.

S_8: The destination node sends a PathErr message. The Markov chain transits to S_0.

S_9: The source node sends a PathTear message. If the destination node receives the message, the Markov chain transits to S_0. Otherwise, the Markov chain transits to S_{10}.

Table 4.2: Transition rates of the state transition

Transition	Rate	
	RSVP-TE	RSVP-TE/Ack
$S_0 \rightarrow S_1$	λ_r	
$S_1 \rightarrow S_2$	$\frac{p_b(1-p_l)}{D}$	
$S_1 \rightarrow S_3$, $S_2 \rightarrow S_3$, $S_4 \rightarrow S_5$, $S_9 \rightarrow S_{10}$	$\frac{p_l}{D}$	
$S_1 \rightarrow S_4$	$\frac{(1-p_b)(1-p_l)}{D}$	
$S_2 \rightarrow S_0$, $S_4 \rightarrow S_6$, $S_8 \rightarrow S_0$, $S_9 \rightarrow S_0$	$\frac{1-p_l}{D}$	
$S_3 \rightarrow S_2$, $S_7 \rightarrow S_8$	$\frac{p_b(1-p_l)}{T}$	$p_b(1-p_l)\left(\frac{1}{T}+\frac{1}{R}\right)$
$S_3 \rightarrow S_4$, $S_7 \rightarrow S_6$	$\frac{(1-p_b)(1-p_l)}{T}$	$(1-p_b)(1-p_l)\left(\frac{1}{T}+\frac{1}{R}\right)$
$S_6 \rightarrow S_9$	μ	
$S_5 \rightarrow S_3$, $S_6 \rightarrow S_5$, $S_6 \rightarrow S_7$, $S_7 \rightarrow S_3$	$\frac{p_l^k}{X}$	$\frac{p_l^{(k-1)(m+1)+1}}{X}$
$S_{10} \rightarrow S_0$	$\frac{1}{X}$	$\frac{1-p_l}{R}+\frac{1}{X}$

S_{10}: If a Path state at the destination node is deleted by a state timeout, the Markov chain transits to S_0.

The state transition of RSVP-TE/Ack is obtained by some replacements of the transition rates of RSVP-TE as in Table 4.2. The retransmission rate in RSVP-TE/Ack is given as $1/R$; therefore, the rate that refresh messages are sent in RSVP-TE/Ack is $1/T + 1/R$. RSVP-TE/Ack can also retransmit teardown messages. The rate of $S_{10} \rightarrow S_0$ in RSVP-TE/Ack is $1/X + (1-p_l)/R$ since the probability that a retransmitted message reaches the receiver node is $(1-p_l)$.

The hard-state BR does not use timers or refresh messages; and the rate that signaling messages are retransmitted in the hard-state BR is $1/R$. The state transition of the hard-state BR is obtained by replacing the transition rates of RSVP-TE/Ack, that is, replacing $1/T$ and $1/X$ with 0. Then, states S_7 and S_8 become unreachable and can be removed.

4.2.2 Model of GMPLS RSVP-TE for Single-Hop LSP with Control Plane Failure

Here we consider the model of GMPLS RSVP-TE with control plane failure. To develop this model, we add the following assumptions.

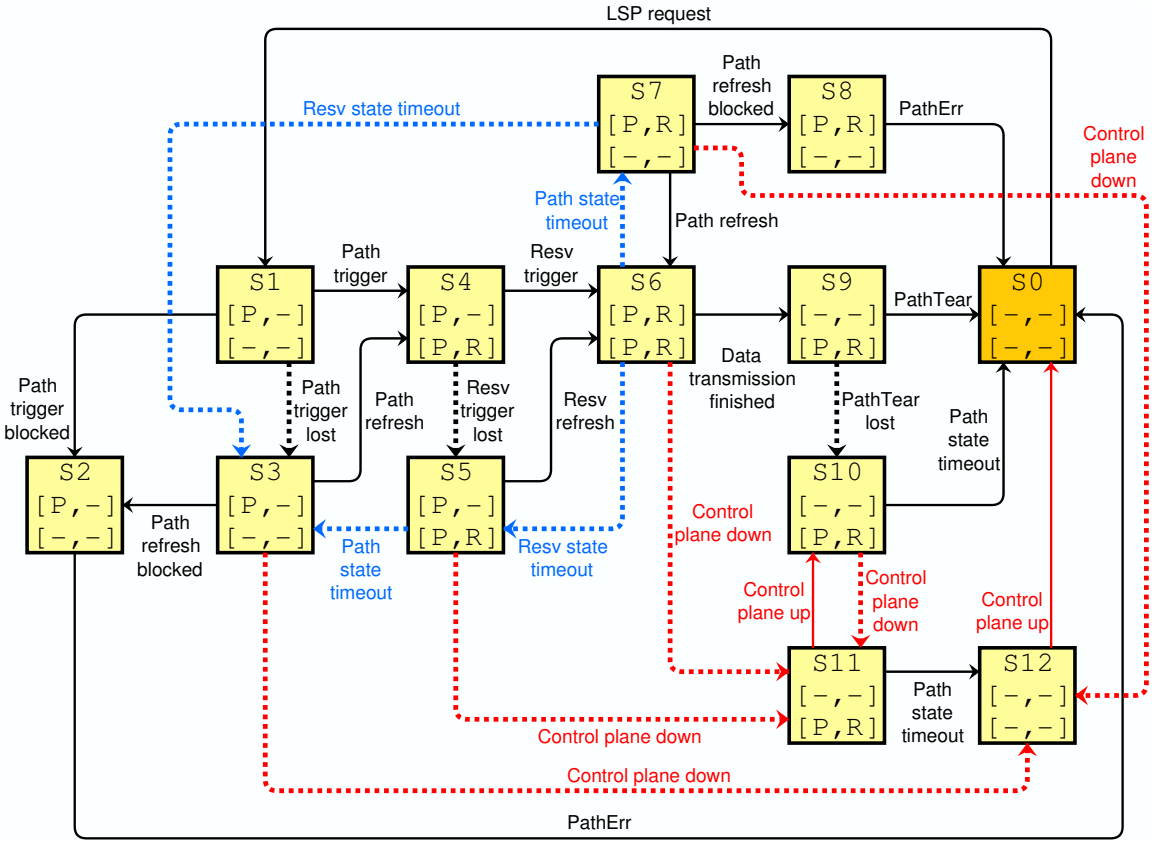


Figure 4.3: State transition of GMPLS RSVP-TE for single-hop LSPs with control plane failure

- When a failure occurs on a control plane, all the communications of signaling messages among the nodes become impossible. This is the worst case of control plane failure.

- When a source node finds that a failure has occurred in a control plane, the source node deletes its Path state immediately.

- In our analysis, we set control plane failures to occur in accordance with a Poisson process with rate ϕ, and the delays to recover from control plane failures follow an exponential distribution with rate γ.

Figure 4.3 shows the state transition of RSVP-TE for a single-hop LSP with control plane failure. Two new states, S_{11} and S_{12}, and their associated transitions are added to the state transition in Figure 4.2. Control plane failures would occur at S_3, S_5, S_6, and S_{10}. At S_3, if a control plane failure occurs, the Markov chain transits to S_{12}. While at the other

Table 4.3: Rates of the additional transitions for control plane failure

Transition	Rate	
	RSVP-TE	RSVP-TE/Ack
$S_3 \rightarrow S_{12}$, $S_5 \rightarrow S_{11}$, $S_6 \rightarrow S_{11}$, $S_7 \rightarrow S_{12}$, $S_{10} \rightarrow S_{11}$	ϕ	
$S_{11} \rightarrow S_{10}$, $S_{12} \rightarrow S_0$	γ	
$S_{11} \rightarrow S_{12}$	$\frac{1}{X}$	

states, if a control plane failure occurs, the Markov chain transits to S_{11}. RSVP-TE works at S_{11} and S_{12} as follows.

S_{11}: If a control plane recovers from a failure, the Markov chain transits to S_{10}. If the Path state at the destination node is deleted by a state timeout, the Markov chain transits to S_{12}.

S_{12}: If a control plane recovers from a failure, the Markov chain transits to S_0.

The rates of the added transitions are listed in Table 4.3. The state transitions of RSVP-TE/Ack and the hard-state BR are obtained in the same way as in Section 4.2.1.

4.2.3 Analysis of GMPLS RSVP-TE for Single-Hop LSP

We analyze the performance of GMPLS RSVP-TE with our models presented in Section 4.2.1 and 4.2.2. In this analysis, we quantitatively demonstrate how soft-state protocols are affected by control parameter settings.

As the performance metric for this analysis, we use unoccupied time, which is defined as the time that a label is reserved but not used for data transmission. The unoccupied time is caused by the inconsistency of signaling states at nodes along an LSP. The longer the unoccupied time is, the lower the resource utilization becomes. Therefore, it is essential for signaling protocols to shorten this inconsistency period. Note that the minimum unoccupied time is the round-trip time of an LSP.

The unoccupied time is obtained by using the steady-state probabilities. Supposing that the state transition of GMPLS RSVP-TE is composed of N states, π_i is the steady-state probability for S_i ($i = 0, 1, \cdots, N-1$), and t_i is the average total time that the process of

GMPLS RSVP-TE is at S_i. Let τ be the average duration from the beginning to the end of GMPLS RSVP-TE sessions. A GMPLS RSVP-TE session starts when a source node sends a Path trigger message to establish an LSP and finishes when the LSP is removed after the data transmission. Here, t_i is expressed as

$$t_i = \pi_i \tau.$$

From this equation, the relation between any two steady-state probabilities can be described as

$$\frac{\pi_i}{\pi_j} = \frac{t_i}{t_j} \quad (i, j = 0, 1, \cdots, N-1).$$

Since the average time of data transmission is $1/\mu$,

$$t_i = \frac{\pi_i}{\mu \pi_d},$$

where S_d is the state that a source node transmits data on an established LSP. The steady-state probabilities can be obtained by solving the state transition equation. Let $S^{'}$ be a set of the states for which a label is reserved but unoccupied for data transmission. The unoccupied time $\tau^{'}$ is defined as follows:

$$\tau^{'} = \sum_{i \in I^{'}} t_i = \sum_{i \in I^{'}} \frac{\pi_i}{\mu \pi_d} \quad (I^{'} = \{i \mid S_i \in S^{'}\}).$$

In the state transition in Figure 4.2, the states having a Resv state are $S_4, S_5, \cdots, S_{10}$. Since the state that a source node transmits data to the destination node is S_6, $\tau^{'}$ is,

$$\tau^{'} = \frac{\pi_4 + \pi_5 + \pi_7 + \pi_8 + \pi_9 + \pi_{10}}{\mu \pi_6}. \tag{4.1}$$

For the state transition of Figure 4.3, $\tau^{'}$ is given by

$$\tau^{'} = \frac{\pi_4 + \pi_5 + \pi_7 + \pi_8 + \pi_9 + \pi_{10} + \pi_{11}}{\mu \pi_6}. \tag{4.2}$$

Table 4.4: Definitions of protocols and their parameter settings

Protocol	T	k	R	m
RSVP-TE	30	3	–	–
RSVP-TE(SL)	0.5	3	–	–
RSVP-TE(FR)	0.5	180	–	–
RSVP-TE/Ack	30	3	0.5	3
HS-BR	–	–	0.5	∞

The arrival rate of LSP requests has no impact on the unoccupied time since τ is the average duration from the beginning to the end of the GMPLS RSVP-TE sessions. Hence, we merged S_0 and S_1 into a state and solved the state transition equation. We compare the unoccupied times of five signaling protocols in Table 4.4. RSVP-TE(SL) is a variant of RSVP-TE, whose refresh interval is as short as the retransmission interval of RSVP-TE/Ack. Note that the state lifetime of RSVP-TE(SL) is also shortened to 1.5 sec from 90 sec. RSVP-TE(FR) has the same refresh interval as RSVP-TE(SL) and the same state lifetime as RSVP-TE. HS-BR is BR with hard-state control that has the same retransmission interval as RSVP-TE/Ack. Since the message retransmission continues until a sender node confirms that the signaling message has been received by the receiver node in HS-BR, the maximum number of retransmission times is unlimited. In what follows, we use these parameter values unless otherwise specified: $D = 0.001$, $T = 30$, $k = 3$, $\mu = 0.00001$, $p_l = 0.00001$, $p_b = 0.001$, $R = 0.5$, and $m = 3$. D does not affect the increase of LSP setup and teardown delays but just decides the minimum of those delays. The default values of T, k, R, and m are described as standard or reference values in [57, 58].

There are three factors that control whether reserved labels remain unoccupied in RSVP-TE: propagation delay, signaling message loss, and false removal. Propagation delay, D, is unavoidable and thus determines the minimum unoccupied time. Signaling message loss occurs with the probability p_l. If p_l is not small enough, the unoccupied time is increased by signaling message loss. The probability that a false removal occurs is proportional to the message loss probability to the power of n, p_l^n ($n = k$ for RSVP-TE; $n = (k-1)(m+1)+1$ for RSVP-TE/Ack). Meanwhile, the unoccupied time of HS-BR has nothing to do with false removal because HS-BR does not use any timers.

Figure 4.4 shows the unoccupied time, which is dependent on the signaling message

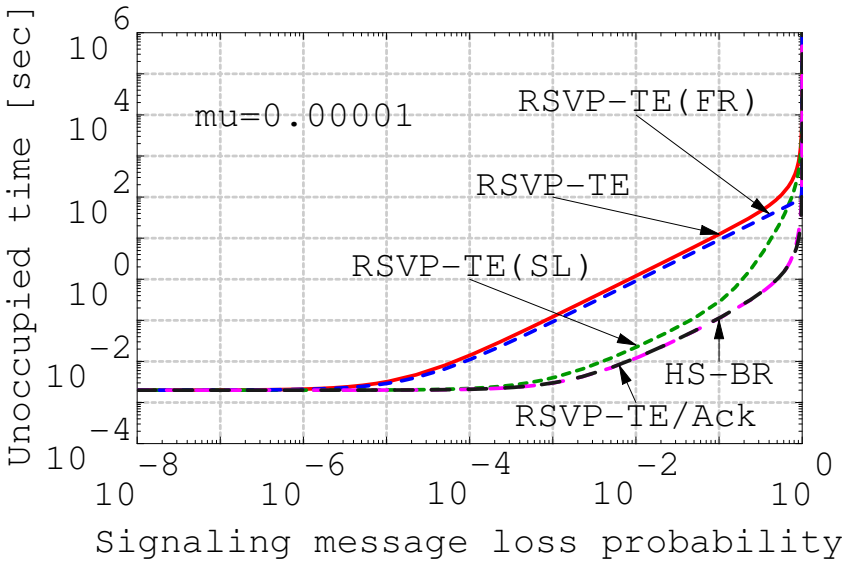
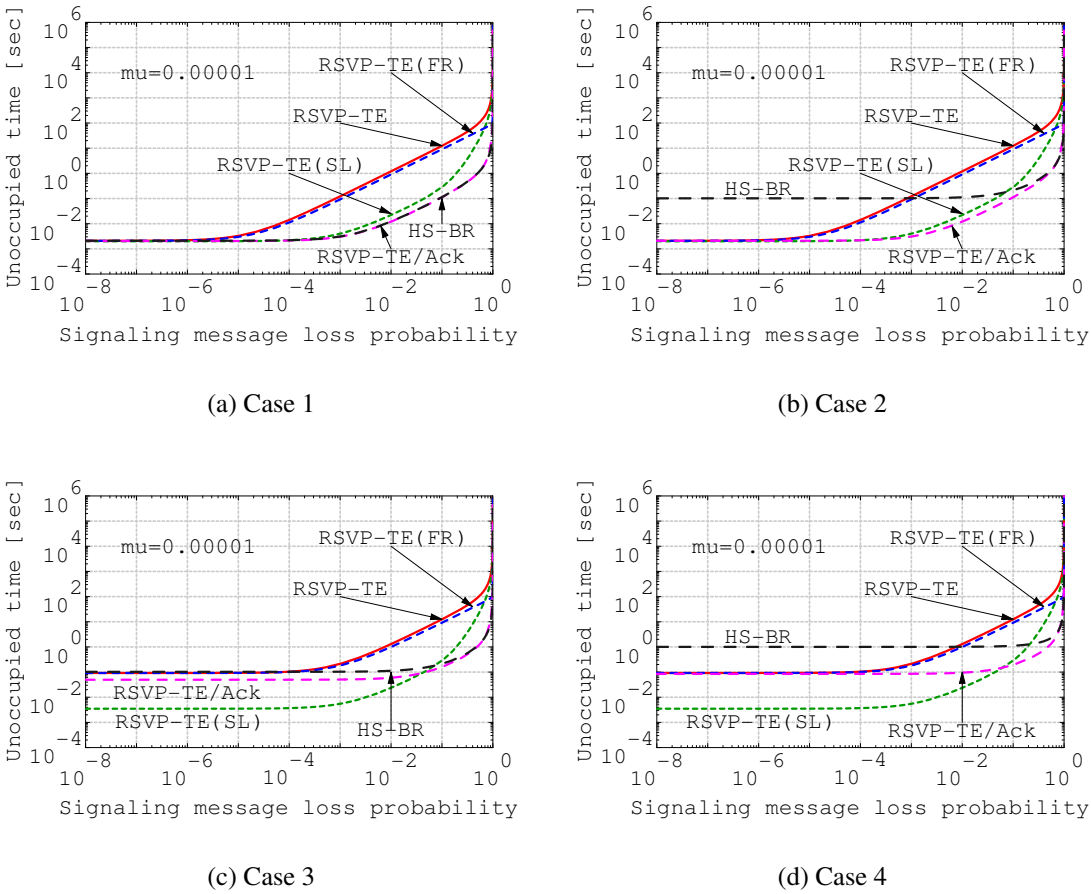


Figure 4.4: Unoccupied time versus message loss probability for a single-hop LSP without control plane failure

loss probability for a single-hop LSP without control plane failure. The time unit is seconds. When the signaling message loss probability is smaller than 10^{-6}, there is no difference in the unoccupied time among the five protocols since message losses seldom occur. When the message loss probability is greater than 10^{-6}, the increase of unoccupied time in RSVP-TE is mainly due to losses of PathTear messages. In RSVP-TE, since PathTear messages are not retransmitted, if a PathTear message is lost, control states at a destination node are not deleted until the state timeout timer expires. RSVP-TE(SL) and RSVP-TE(FR) do not retransmit signaling messages, though the performance degradation of RSVP-TE(SL) is less than those of RSVP-TE and RSVP-TE(FR) since the state lifetime of RSVP-TE(SL) is quite short. The difference in unoccupied time between RSVP-TE and RSVP-TE(FR) comes from occurrences of false removals. False removals are likely to occur when the message loss probability is high. According to Figure 4.4, the influence of false removal does not appear if the message loss probability is lower than 0.1.

The results of RSVP-TE/Ack exhibit a similar tendency as HS-BR, where the unoccupied time of RSVP-TE/Ack is shorter than that of RSVP-TE(SL) since RSVP-TE/Ack can retransmit PathTear messages. In addition, the retransmission of refresh messages enables RSVP-TE/Ack to avoid false removals even when the message loss probability is high.

At this point we investigate the performance of GMPLS RSVP-TE for a single-hop



(a) Case 1

(b) Case 2

(c) Case 3

(d) Case 4

Figure 4.5: Unoccupied time versus message loss probability for a single-hop LSP with control plane failure

LSP with control plane failure. We analyzed the unoccupied time in these four cases.[2]

Case 1: Control plane failures rarely occur and it does not take a long time for the control plane to recover from a failure ($\phi = 10^{-8}$ and $\gamma = 10^{-2}$).

Case 2: Control plane failures rarely occur and it takes a long time for the control plane to recover from a failure ($\phi = 10^{-8}$ and $\gamma = 10^{-5}$).

Case 3: Control plane failures frequently occur and it does not take a long time for the control plane to recover from a failure ($\phi = 10^{-5}$ and $\gamma = 10^{-2}$).

Case 4: Control plane failures frequently occur and it takes a longer time for the control plane to recover from a failure than in case 3 ($\phi = 10^{-5}$ and $\gamma = 10^{-3}$).

[2] 1 day $= 86,400$ sec $< 10^5$ sec. 3 year $= 93,312,000$ sec $< 10^8$ sec.

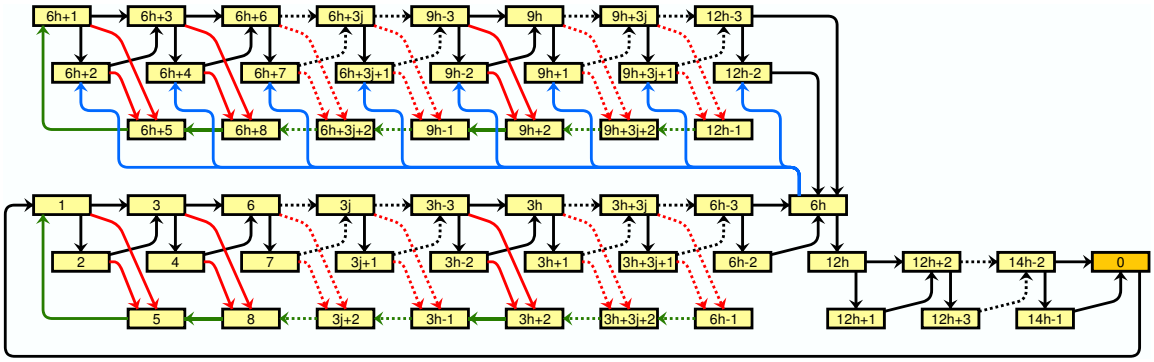


Figure 4.6: State transition of RSVP-TE for an h-hop LSP

Figure 4.5 shows the unoccupied times in these four cases. As can be seen from the comparison between Figure 4.4 and Figure 4.5(a), the influence of control plane failure does not appear in Case 1. However, Figure 4.5(b) shows that the performance of HS-BR decreases even when the message loss probability is low. This is because HS-BR does not have the state timeout mechanism and must wait until the control plane recovers in order to release the reserved resources. This tendency can also be seen in Case 3 (Figure 4.5(c)) and Case 4 (Figure 4.5(d)), where control plane failures occur frequently. On the other hand, the unoccupied time of RSVP-TE is independent of the recovery time. The unoccupied times of RSVP-TE in Cases 1 and 2 are almost the same, and there is no difference between the unoccupied times of RSVP-TE in Cases 3 and 4, too. These results indicate that the soft-state protocols are stable in terms of control plane failures.

4.3 Model and Analysis of GMPLS RSVP-TE for Multi-Hop LSP

In this section, we develop the model of GMPLS RSVP-TE for multi-hop LSPs and analyze LSP setup delay, recovery delay, and teardown delay. LSP setup delay is the time from when a source node sends a Path trigger message till when an LSP is established. Recovery delay is the time from when an LSP is disrupted by a false removal till when the disrupted LSP recovers. Teardown delay is the time from when a source node sends a PathTear message till when an LSP is completely deleted. We do not discuss the control plane failure here but it can be extended to our model, as in Section 4.2.2.

4.3.1 Model of GMPLS RSVP-TE for Multi-Hop LSP

To analyze the performance of GMPLS RSVP-TE for multi-hop LSPs, we assume that false removals never occur during the LSP setup and recovery phase. That is, we consider false removals only when the LSP is established. Although we can develop the Markov model without this assumption, the number of states rapidly increases with an increasing number of hops. This is because states have to be prepared based on where and when false removals occur. Furthermore, since the LSP holding time (an order of seconds or more) is longer than the LSP setup delay (in the order of ms), the impact of false removals during the LSP setup phase would be small. Actually, the probability that a false removal occurs is quite low in the single-hop case (see the difference between RSVP-TE and RSVP-TE(B) in Figure 4.4). Therefore, we assume here that false removals occur after a LSP is successfully established. To enable our model to analyze the recovery time, we also assume that a disrupted LSP is recovered on the same route after a false removal occurs.

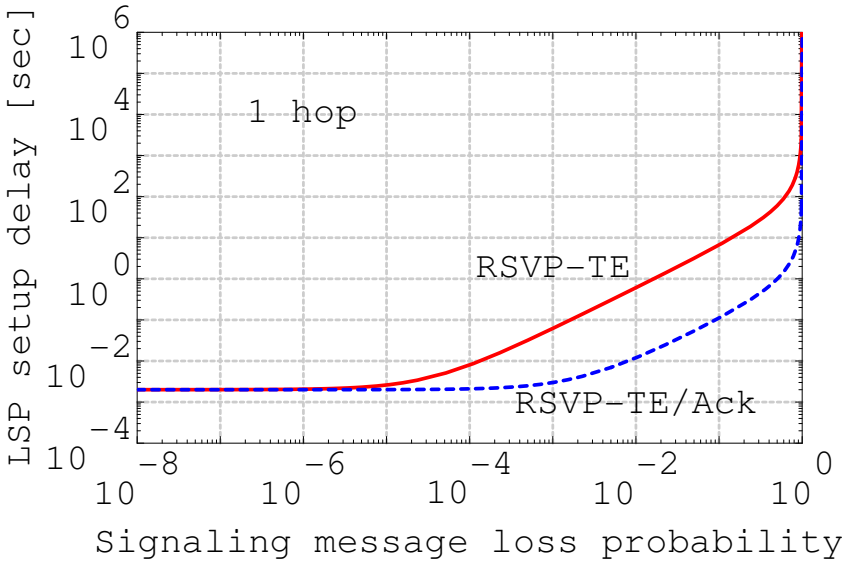
Figure 4.6 illustrates the state transition of RSVP-TE for an h-hop LSP, where rectangles represent the states and the number of states is $14h$. The index of state S_i, i, is denoted inside each rectangle. The process of setting up an LSP setup is modeled with the states S_1 to S_{6h-1}, while the process of recovery from a false removal is modeled with the states S_{6h+1} to S_{12h-1}, and LSP teardown is modeled with the states S_{12h} to S_{14h-1}. Refer to Appendix A for the detail description of these state transitions.

4.3.2 Analysis of GMPLS RSVP-TE for Multi-Hop LSP

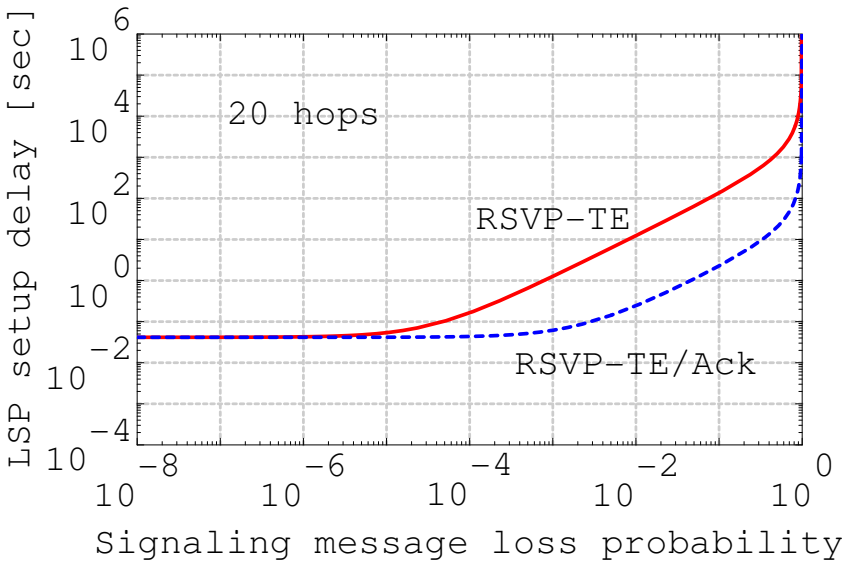
We can analyze the setup delay, the recovery delay, and the teardown delay for an LSP, T_S, T_R, and T_D, by the model described above. As we discussed in Section 4.2, these delays are obtained with fractions of the steady-state probabilities:

$$T_S = \frac{\sum_{j=1}^{6h-1} \pi_j}{\mu \pi_{6h}}, \quad T_R = \frac{\sum_{j=6h+1}^{12h-1} \pi_j}{\mu \pi_{6h}}, \quad T_D = \frac{\sum_{j=12h}^{14h-1} \pi_j}{\mu \pi_{6h}}. \tag{4.3}$$

Figure 4.7 compares the LSP setup delay between a single-hop LSP and a 20-hop LSP. The horizontal axes represent the loss probability of signaling messages, and the vertical



(a) 1 hop



(b) 20 hops

Figure 4.7: Comparison of setup time between different lengths of LSP

axes represent the LSP setup delay. Although setup delays are different due to the propagation delay, the points at which the setup delays of RSVP-TE and RSVP-TE/Ack start to rise are almost the same (10^{-6} for RSVP-TE and 10^{-4} for RSVP-TE/Ack). That is, the properties of RSVP-TE and RSVP-TE/Ack in regard to the signaling message loss probability are independent of LSP length. This means that the results of our analysis in Section 4.2 are

applicable for discussing the effectiveness of RSVP-TE and RSVP-TE/Ack for multi-hop LSPs.

4.4 Effectiveness of Message Retransmission

In previous sections, we compared RSVP-TE with RSVP-TE/Ack in instances where the signaling message loss probabilities are the same. However, the number of signaling messages in RSVP-TE/Ack is greater than that in RSVP-TE since signaling messages would be retransmitted in RSVP-TE/Ack. Since the size of the receive buffer is finite, if the number of LSP sessions increases, the signaling message loss probability also increases. In this section, we reconsider the effectiveness of message retransmission in RSVP-TE/Ack taking into account the increment of message loss probability by message retransmission. We apply the results of our analysis for a single LSP in Section 4.2 to show when message retransmission is efficient and when it is inefficient.

4.4.1 Model of Signaling Message Loss

It is assumed that losses of signaling messages occur only due to the buffer overflow in the receive buffer. We also assume that the signaling messages in RSVP-TE arrive according to the Poisson process with rate λ_1 and that the processing time of a signaling message follows the exponential distribution with rate μ_p. When there are w LSP sessions, the total message transmission rate is $w\lambda_1$. Therefore, the message loss probability of RSVP-TE, P_{b_1}, is described with the $M/M/1/K$ queuing model:

$$P_{b_1} = \frac{(w\rho_1)^K}{\sum_{i=0}^{K}(w\rho_1)^i} = \frac{(1-w\rho_1)(w\rho_1)^K}{1-(w\rho_1)^{K+1}}, \tag{4.4}$$

where ρ_1 is defined as λ_1/μ_p. For RSVP-TE/Ack, the message loss probability, P_{b_2}, is given in the same manner. That is:

$$P_{b_2} = \frac{(w\rho_2)^K}{\sum_{i=0}^{K}(w\rho_2)^i} = \frac{(1-w\rho_2)(w\rho_2)^K}{1-(w\rho_2)^{K+1}}, \tag{4.5}$$

where $\rho_2 = \lambda_2/\mu_p$, and λ_2 is the arrival rate of signaling messages in RSVP-TE/Ack. Solving Eq. (4.4) for K,

$$K = \frac{\log\left[\dfrac{P_{b_1}}{1-(1-P_{b_1})w\rho_1}\right]}{\log\left[w\rho_1\right]} \tag{4.6}$$

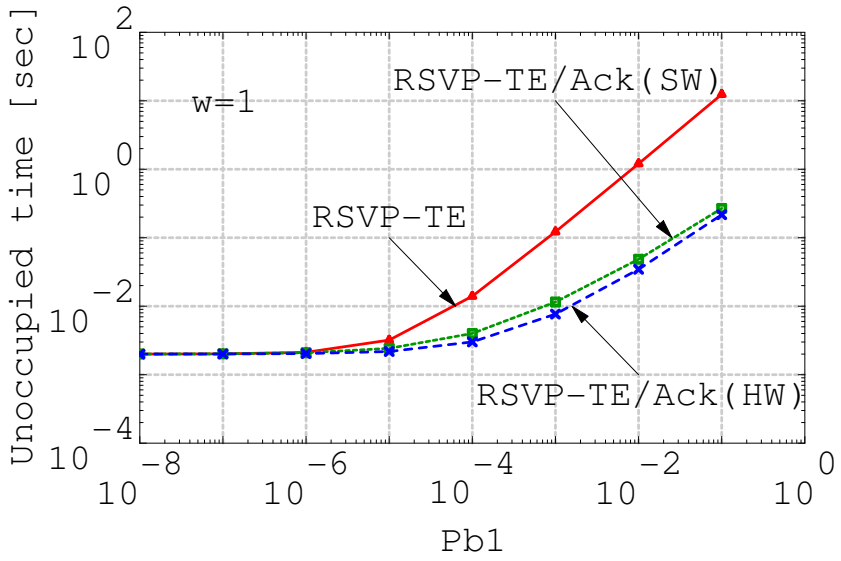
is obtained. Then, P_{b_2} is expressed as a function of P_{b_1} by substituting Eq. (4.6) into Eq. (4.5).

In RSVP-TE protocols, signaling messages are sent in the forward (from a source node to a destination) and backward directions. Here we focus only on the signaling messages sent in the forward direction. In the state transition of Figure 4.2, Path and PathTear fall into such messages. Path trigger messages are sent at state S_1 in Figure 4.2 at a rate of $1/D$, while Path refresh messages are sent at states S_3, S_5, S_6, and S_7. PathTear messages are sent at state S_9. Hence, λ_1 is given as

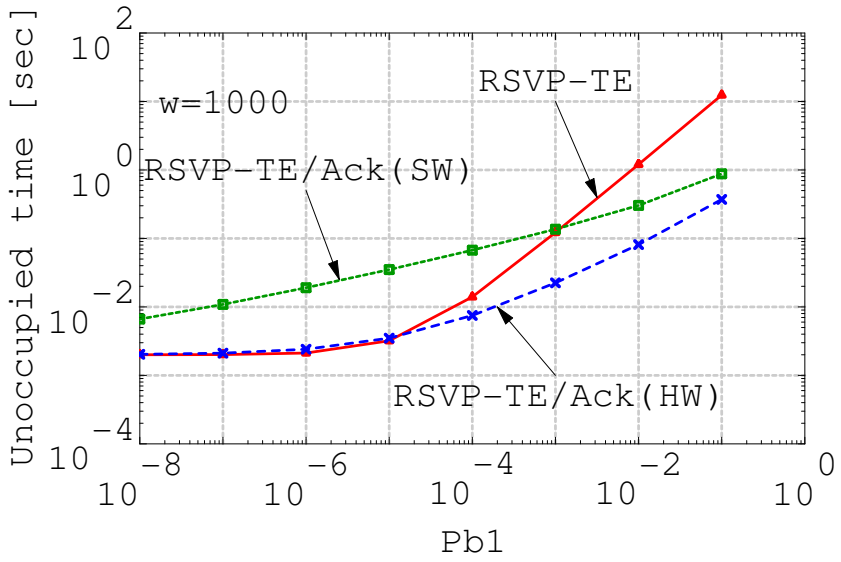
$$\lambda_1 = \frac{1}{D}(\pi_1 + \pi_9) + \frac{1}{T}(\pi_3 + \pi_5 + \pi_6 + \pi_7).$$

In RSVP-TE/Ack, Path messages would be retransmitted at the rate of $1/R$ at states S_3, S_5, S_6, and S_7, and PathTear messages would also be retransmitted at $1/R$ at state S_{10}. Thus, λ_2 is given as

$$\lambda_2 = \lambda_1 + \frac{1}{R}(\pi_3 + \pi_5 + \pi_6 + \pi_7 + \pi_{10}).$$



(a) $w = 1$



(b) $w = 1000$

Figure 4.8: Effectiveness of message retransmission of RSVP-TE/Ack

4.4.2 Numerical Examples

The average connection time of LSP is 100,000 sec since $\mu = 0.00001$. This is sufficiently large that $\pi_i/\pi_6 \approx 0$ $(i = 1, 2, \cdots, 10, i \neq 6)$. Therefore,

$$\lambda_1 \approx \frac{1}{T}, \tag{4.7}$$

$$\lambda_2 \approx \frac{1}{T} + \frac{1}{R}. \tag{4.8}$$

In [59], an RSVP-TE software module is implemented and takes about 0.1 msec to process a signaling message. On the other hand, an RSVP-TE hardware module is implemented in [60] and it requires about 2.4 μsec to process a signaling message. We use these values for μ_p. Figure 4.8 illustrates the effectiveness of message retransmission, with the horizontal axes representing P_{b_1}, and the vertical axes representing the unoccupied time for a single-hop LSP. The unoccupied times of RSVP-TE/Ack are obtained with the model in Section 4.1 and B_2 that is calculated using Eqs. (4.5), (4.6), (4.7), and (4.8). The plots of RSVP-TE/Ack (SW) are the unoccupied times where the RSVP-TE module is implemented with software. RSVP-TE/Ack (HD) represents that the RSVP-TE module is implemented with hardware. RSVP-TE/Ack outperforms RSVP-TE regardless of the type of implementation when the number of sessions is one. However, when the number of sessions is 1000, the unoccupied time of RSVP-TE is shorter than that of RSVP-TE/Ack (SW) when the message loss probability in RSVP-TE is lower than 10^{-3}. This implies that message retransmission can result in poor resource utilization if the message loss probability is low.

4.5 Summary

In this chapter, we developed a Markov model of GMPLS RSVP-TE for single-hop and multi-hop LSPs and analyzed the performance of variants of GMPLS RSVP-TE. From the results, we demonstrated that the performance of RSVP-TE is close to the performance of a hard-state protocol when the loss probability of signaling messages is relatively low. In contrast to soft-state protocols, hard-state protocols do not have a way to manage signaling states under the control plane failure. The results regarding the control plane failure also show that the unoccupied time of hard-state signaling become worse than the performance of soft-state signaling.

Message retransmission improves the responsiveness of GMPLS RSVP-TE when signaling messages are lost. However, it also increases the number of signaling messages and raises the probability of signaling message loss. We used the numerical results of our analysis to investigate the effectiveness of message retransmission, and found that the use of message retransmission can result in poor resource utilization. Specifically, when the signaling message loss probability is lower than 0.001 and when there are more than 1,000 LSP sessions, using message retransmission decreases the resource utilization of RSVP-TE if the RSVP-TE modules are implemented with software. Even if the RSVP-TE modules are implemented with hardware, this can be observed when there are more LSP sessions.

As for future research, we plan to analyze the performance of other signaling protocols for wavelength-routed networks, such as Parallel Reservation [27], and to compare the performance of soft-state and hard-state signaling protocols in the transient state.

Chapter 5

Local Recovery Scheme for Massive Failures

As the number of nodes and links increase, the probability that failures occur increases. Therefore, we should consider not only single node or link failures but also multiple node or link failures. Multiple failures may occur by independent single failures in various places of networks and by failures in a certain region due to earthquakes or accidental power cut. In this thesis, we call the failures in the latter case *massive failures*.

There are lots of schemes to recover lightpath connections and they are categorized into two groups; *protection* and *restoration*. In protection schemes, extra wavelength resources are provisioned for backup of the working lightpaths. Although protection schemes guarantee 100% recovery against only the specified failure scenarios, they cannot deal with the other scenarios that are not taken into account. Therefore, it is difficult to deal with many kinds of failure scenarios with only protection schemes.

On the other hand, restoration schemes reactively search a new path and reserve wavelength after the failure of a working lightpath. Restoration schemes provide more flexible recovery from failures. However, restoration schemes take time for their signaling and the time increases proportionally to the distance between end nodes. The increase of the hop-length of lightpaths results in the high blocking probability of wavelength reservation during the restoration. Although there are link restoration schemes, they are not available in the cases that all the divert routes between the edge nodes of a failed link due to a massive

failure. In addition, restoration schemes distribute lots of control messages into the control plane for failure notifications, link-state advertisement, and wavelength reservation. The increase of control messages results in the increase of the queuing delay and the message loss probability and would influence the control sessions for lightpaths not disrupted by the failures.

In this chapter, we propose a restoration scheme that is applicable to any kinds of failures and reduces the number of control messages for the restoration. Our scheme calculates a cycle enclosing the failed part of the network, called a *diverting cycle*, in a distributed way and diverts disrupted lightpaths along the cycle. Our scheme also reduces the number of control messages during the recovery and avoids the congestion in the control plane. We evaluate the performance of our scheme by computer simulations. The results show that our scheme recovers the lightpath connectivity to almost 100% more quickly than the path restoration scheme when the scale of massive failures is not large. When the scale of massive failures is large, our scheme reduces the number of control messages to about the half comparing to the path restoration scheme.

5.1 Local Recovery of Lightpath Connections with Diverting Cycles

Usually, nodes that detect failures of network components are likely to inform other nodes immediately since those failures cause disconnections of lightpaths in the network. However, when a massive failure occurs in a large-scaled network, a lot of notification messages are flooded and notified nodes send control messages to recover failed lightpaths. This rush of control messages increases the queuing delay in the control plane, causes control message losses, and would influence the management of other lightpath sessions having nothing to do with the failure. As a result, despite the emergency condition, the recovery from the massive failure is delayed and the influence of the failure is spread in the network.

In this section, we propose a restoration scheme that restores lightpath connections locally and restricts the number of control messages during the restoration. This scheme is applicable not only to multiple single failures in various places on networks but also to

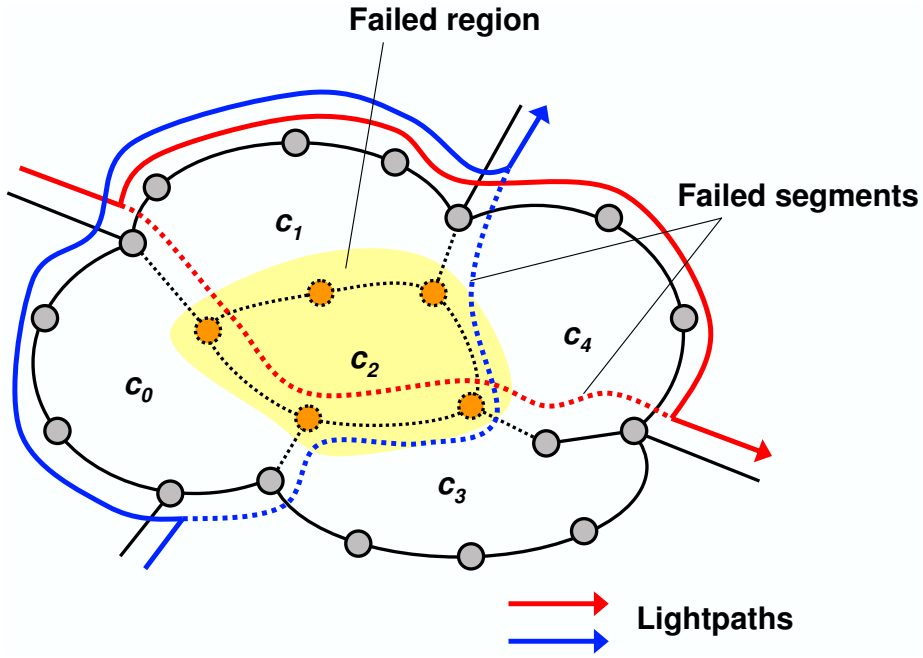


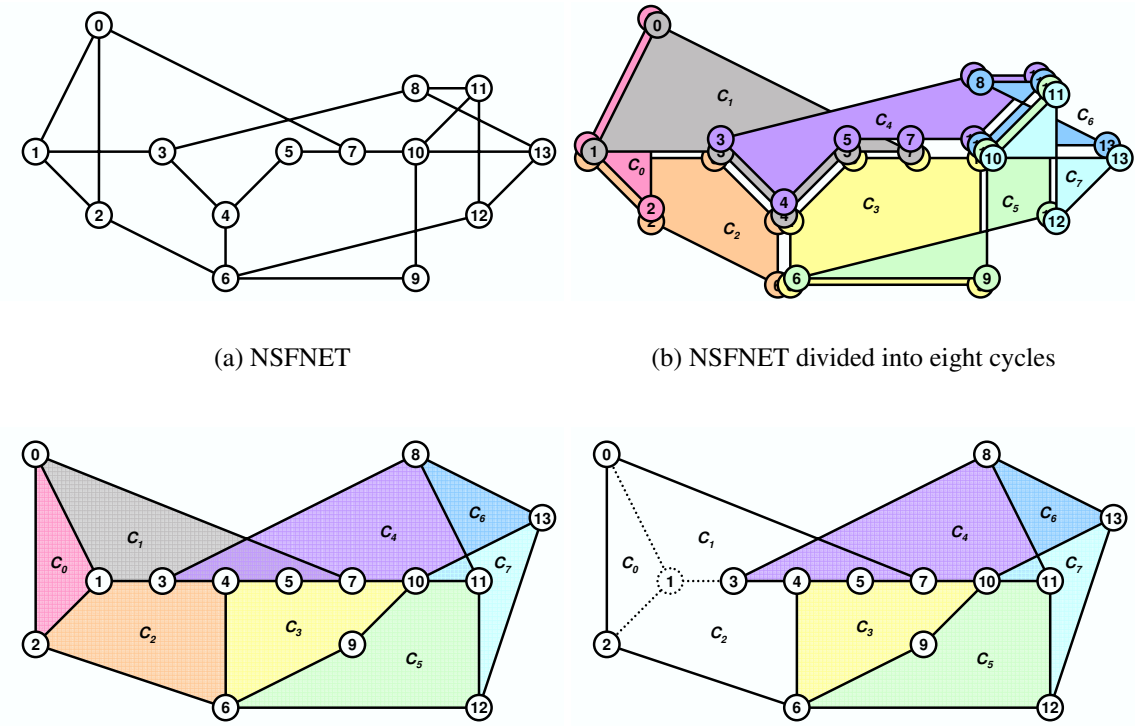
Figure 5.1: Rerouting along a diverting cycle

massive failures.

5.1.1 Outline of Local Recovery with Diverting Cycles

In our scheme, we divide the network topology into cycles (Figure 5.1). The OXCs on each cycle are managed by a controller chosen from the controllers of the nodes on the cycle. In Figure 5.1, when a massive failure occurs at the nodes along cycle c_2, the other cycles c_0, c_1, c_3, and c_4, which are sharing the failed nodes with c_2, are merged and a diverting cycle that encloses the failed region is constructed. Lightpaths torn down by the failure are locally recovered by being rerouted along the diverting cycle. This recovery process is done by a controller. That controller is chosen from the controllers of the cycles merged into the diverting cycle in a distributed way.

In Figure 5.2, we give an illustrative example using the NSFNET topology. At first, the NSFNET topology given in Figure 5.2(a) is divided into cycles as shown in Figure 5.2(b). Figure 5.2(c) illustrates the topological adjacency of the cycles. Assuming that node 1 is failed under this circumstance, the cycles including node 1, c_0, c_1, and c_2, are dynamically merged into a cycle enclosing node 1 (Figure 5.2(d)). Then, the disrupted lightpaths passing through node 1 can be diverted along the cycle enclosing node 1.



(a) NSFNET

(b) NSFNET divided into eight cycles

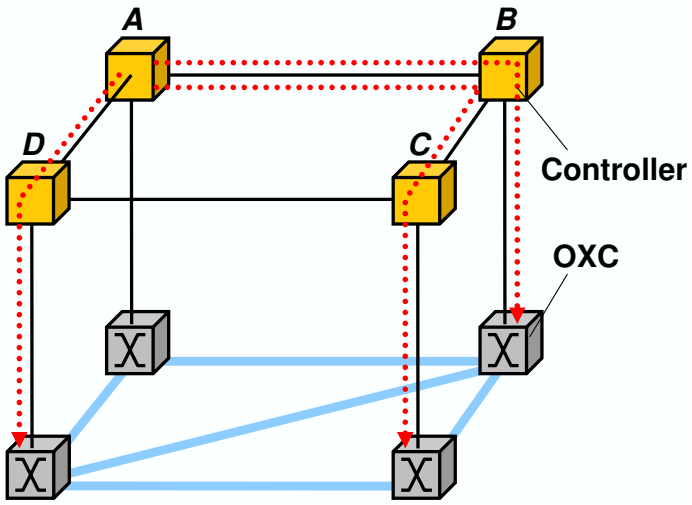
(c) Topological adjacency of the cycles

(d) Enclosure of a node failure by the cycles sharing the node

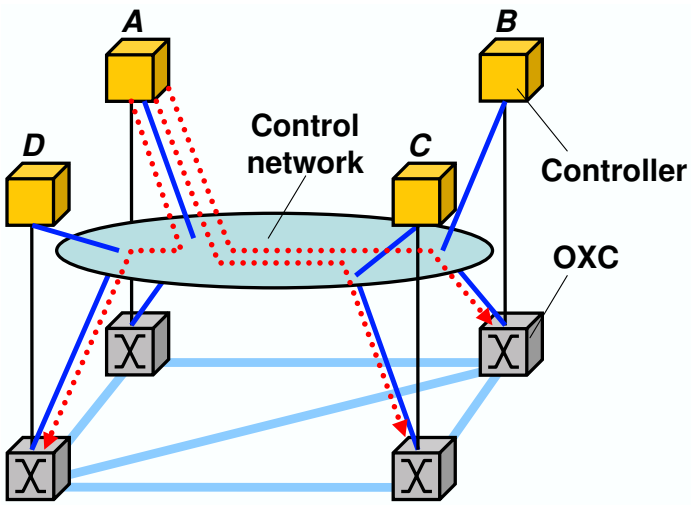
Figure 5.2: Example of cycle division and enclosing a failure

A controller assigned to the diverting cycle manages the wavelength resources of the links along the cycle in a centralized way and can optimize the wavelength assignment for lightpaths. Therefore, our recovery scheme fulfills the wavelength continuity constraint more easily than other distributed recovery scheme. In addition, the locally centralized management restricts the increase of the number of control messages and the distribution of the messages over the control plane.

By dividing network topologies into cycles and assigning a controller to each cycle, the links and OXCs shared by cycles have multiple redundant controllers. This lets the control plane be more reliable against failures of controllers.



(a) Typical linkages among controllers and OXCs



(b) Remote control channels via a control network

Figure 5.3: Association among OXCs and controllers

5.1.2 Node Architecture for Cycle Management

In typical wavelength-routed networks, each node is composed of a controller and an OXC. The controller directly connected by a control channel as in Figure 5.3(a). In the architecture of Figure 5.3(a), controllers cannot configure OXCs at remote nodes. In order to configure remote OXCs, controllers must send other controllers control messages. For example, controller A in Figure 5.3(a) must send control messages to controllers B, C, and D to configure their OXCs. Also, controller A must receive control messages from

controllers B, C, and D to monitor their OXCs. Therefore, if a controller is crushed, the corresponding OXC becomes uncontrollable.

GMPLS networks keep the lightpaths if controllers are failed but OXCs are alive [6, 8]. However, in such architectures as in Figure 5.3(a), OXCs are uncontrollable until their controllers are recovered. Failed controllers must recover the control states of their OXCs by signaling with adjacent controllers after they restart. That is, control interfaces and control states of OXCs are lost by controller failures in traditional wavelength-routed networks. One of the simplest solutions against failures of controllers is to install multiple controllers to each node. However, the equipment cost certainly increases.

Hence, we consider another architecture shown in Figure 5.3(b) to enhance the reliability of the controllers without installing extra controllers. In this architecture, while each node is composed of an OXC and a controller directly connected, each of the OXCs and controllers has an interface connected to the common control network. Controllers can configure remote OXCs by establishing control channels via the control network. Thus, our architecture realizes the redundant management of OXCs by multiple controllers and dynamic assignment of controllers to OXCs without increasing the number of controllers.

5.2 A Scheme for Local Recovery from Massive Failures

In this section, we propose a scheme for local recovery from massive failures. This scheme consists of four parts; *a*) dividing a network topology into cycles, *b*) assigning controllers to the cycles *c*) configuring a diverting cycle enclosing failed parts, and *d*) rerouting lightpaths along a diverting cycle.

5.2.1 Dividing a Network Topology into Cycles

Our scheme divides a topology into cycles at first. It is desirable that the sizes of those cycles are as small as possible in order to suppress the expansion of the influence by network failures. Since this division is carried out at the initial phase, it is possible to calculate the set of the cycles at a certain node. Here we give an algorithm for a controller to calculate the initial cycle division. This algorithm works in distributed way. It is supposed that the

network topologies as inputs for this algorithm are mesh and contain only nodes whose degree is more than one. Multiple fibers between a pair of nodes can be taken as a single link in this algorithm.

Notations

V: Set of the nodes in an input topology. $V = \{v_i \mid 0 \leq i \leq |V| - 1\}$.

E: Set of the links in an input topology. $E = \{e_i \mid 0 \leq i \leq |E| - 1\}$.

$e(v_i, v_j)$: Link whose edges are v_i and v_j.

d_i: Degree of node v_i.

A_i: Set of the adjacent nodes of node v_i. $A_i = \{a_k \mid 0 \leq k \leq d_i,\ a_k \in V,\ e(v_i, a_k) \in E\}$.

$C_{i,p,q}$: Set of the cycles containing nodes v_i, a_p, and a_q ($a_p, a_q \in A_i$, $p \neq q$).

C: Set of the cycles for the initial division. The output of this algorithm.

$E_w(C)$: Set of the edges contained in the cycles in C.

$E(c)$: Set of the edges contained in cycle c. $E(c) = \{e(\hat{v}_k, \hat{v}_{k+1}) \mid \hat{v}_k, \hat{v}_{k+1} \in c,\ 0 \leq k \leq |c| - 1\}$, where cycle c, whose length is l, is given by a node sequence as $\{\hat{v}_0, \hat{v}_1, \cdots, \hat{v}_{l-1}, \hat{v}_0\}$.

Algorithm for Node v_i to Determine the Initial Division

Step 1: $C, C_i, E_w(C) \leftarrow \phi$. If $d_i > 2$, go to Step 2. Otherwise go to Step 4.

Step 2: Repeat Step 2.1 for each pair of adjacent nodes ($a_p, a_q \mid a_p, a_q \in A_i,\ p \neq q$). After that, go to Step 3.

Step 2.1: Calculate a cycle $c_{i,p,q} \in C_{i,p,q}$ such that $|c_{i,p,q} < |c|$ against any cycle $c \in C_{i,p,q}$. If $\{a_u\} \cap c_{i,p,q} = \phi$ for adjacent node $a_u \in A_i$ ($u \neq p, q$), $C_i \leftarrow C_i \cup c_{i,p,q}$.

Step 3: For each cycle $c \in C_i$, inform c of all the nodes $\hat{v} \in c$. Go to Step 4.

Step 4: For each cycle c_j informed by node v_j ($i \neq j$, $v_j \in V$), do one of Step 4.1, 4.2, and 4.3. After that, go to Step 5.

Step 4.1: If $d_i = 2$, record the cycle and return ACK to node v_j. Otherwise go to Step 4.2.

Step 4.2: If $c_j \in C_i$, return ACK to node v_j. Otherwise, go to Step 4.3.

Step 4.3: Otherwise, return NACK to node v_j.

Step 5: For each $c \in C_i$, count up the number of replied ACKs (say $vote(c)$, hereafter). Go to Step 6.

Step 6: Repeat Step 6.1 and 6.2 for each cycle $c \in C_i$. After that, go to Step 7.

Step 6.1: If $vote(c) = |c| - 2$, $C \leftarrow C \cup c$, $C_i \leftarrow C_i - c$, $E_w(C) \leftarrow E_w(C) \cup E(c)$. Otherwise go to Step 6.2.

Step 6.2: If $vote(c) < |c|/2 - 1$, $C_i \leftarrow C_i - c$.

Step 7: While $E_w(C) \neq E$, repeat Step 7.1, 7.2, and 7.3. Otherwise go to Step 8.

Step 7.1: Inform cycle $c_i \in C_i$ such that $vote(c_i) \geq vote(c^{'})$ and $|E(c_i)| - |E_w(C) \cap E(c_i)| \geq |E(c^{'})| - |E_w(C) \cap E(c^{'})|$ against any $c^{'} \in C_i$. Go to Step 7.2.

Step 7.2: Store the informed cycles in the previous step into $C^{'}$. Choose a cycle $c \in C^{'}$ such that $vote(c) \geq vote(c^{'})$ and $|E(c)| - |E_w(C) \cap E(c)| \geq |E(c^{'})| - |E_w(C) \cap E(c^{'})|$ against any $c^{'} \in C^{'}$. $C \leftarrow C \cup c$ and $E_w(C) \leftarrow E_w(C) \cup E(c)$. Go to Step 7.3.

Step 7.3: If $c \in C_i$, $C_i \leftarrow C_i - c$.

Step 8: Quit this algorithm.

This algorithm consists of three parts. From Step 2 to Step 3, each node calculates a set of cycles containing that node and just two of the adjacent nodes of the node. The nodes

whose degree is two do not need to calculate cycles by themselves since those nodes can be contracted to a link with the connected two links. From Step 4 to Step 6, cycles certified by all the nodes that those cycles include, are chosen for the initial division. If all the links in a given topology are not covered by the chosen cycles, additional cycles are selected at Step 7.

When the initial set of cycles is calculated with this algorithm by a centralized node, Steps 3, 4, and 7.1 are skipped.

5.2.2 Assigning Controllers to the Cycles

For each cycle in the set of the initial division, one or more controllers belonging to the cycle are assigned to it. In this chapter, we assign controllers so that the numbers of assigned controllers to the cycles become uniform as far as possible. Controllers assigned to a cycle establish control channels to the OXCs on the cycle via the common control network. Each controller holds the information about the association with controllers and cycles. We select the controller having the minimum controller ID among the controllers assigned to each cycle as the master controller for that cycle and the other controllers become backup controllers.

If the number of links is too large comparing to the number of nodes, the number of cycles included in the initial division is greater than the number of controllers. In such cases, some cycle consolidations are needed before assigning controllers.

5.2.3 Configuring a Diverting Cycle Enclosing Failed Parts

When a massive failure occurs in a network, the master controllers of the cycles sharing the failed nodes negotiate with each other and then merge their cycles into a cycle enclosing the failed region. For this cycle consolidation, we consider a distributed algorithm since it is unrealistic to manage the whole of a large-scaled network in a centralized way. We assume that a massive failure is a set of node failures in a certain region and that at least a controller is working for each of the cycles whose nodes are failed by the massive failure. This assumption is reasonable since multiple controllers are assigned to each cycle.

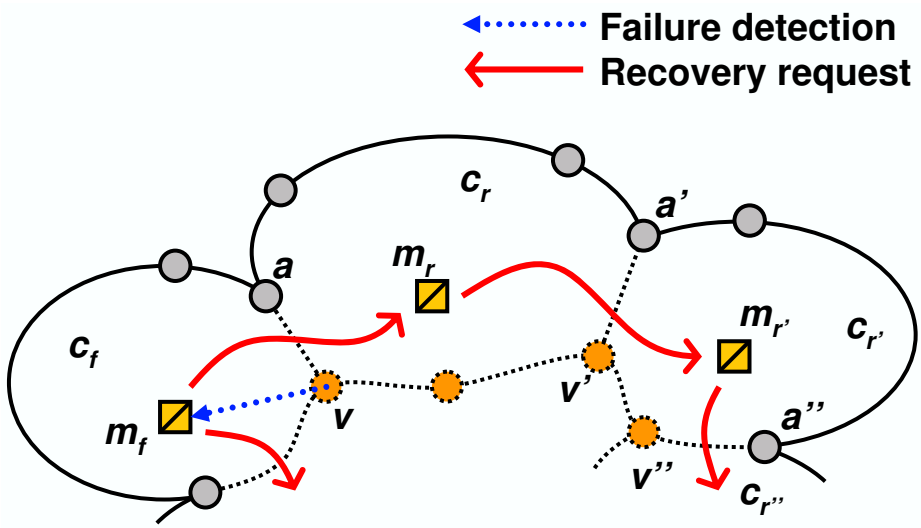


Figure 5.4: Forwarding of a merge request

Notations

In addition to the notations listed in Section 5.2.1, we use following notations.

m_i: Master controller of cycle c_i ($0 \leq i \leq |C| - 1$).

$A_i(v)$: Set of the nodes adjacent to $v \in c_i$ and also included in cycle c_i.

Algorithm for Master Controller m_f to Merge Cycles

When master controller m_f of cycle c_f detects that node $v \in c_f$ fails, m_f sends a merge request message R as follows.

Step 1: Repeat Step 1.1 for each node $a \in A_f(v)$. After that, go to Step 2.

Step 1.1: If node a is not failed and if there is a cycle c_r sharing nodes v and a with cycle c_f, that is, $v, a \in c_r$ ($0 \leq r \leq |C| - 1, r \neq f$), go to Step 1.2.

Step 1.2: Make a merge request message R and store IDs of c_f and the border link $e(v, a)$ in R. Send it to controller m_r.

Step 2: Quit this process.

When controller m_f of cycle c_f receives a merge request message R from controller m_r, m_f goes through these steps.

Step 1: If the generator of R is m_f, go to Step 2. Otherwise, go to Step 1.1.

Step 1.1: Get the last border link, say $e(v, a)$, recorded in R. Go to Step 1.2.

Step 1.2: From node a, look for a failed node $v^{'} \in c_f$ along cycle c_f in the direction opposite to node v. Let $a^{'}$ be the previous hop node of $v^{'}$. Store IDs of c_f and the border link $e(v^{'}, a^{'})$ in R and go to Step 1.3.

Step 1.3: If $\{c \mid c \in C, v^{'}, a^{'} \in c\} \neq \{c_f\}$, $c_{r'} \leftarrow c \in C$, where $v^{'}, a^{'} \in c$ and $c \neq c_f$. Otherwise $c_{r'} \leftarrow c_r$. Go to Step 1.4.

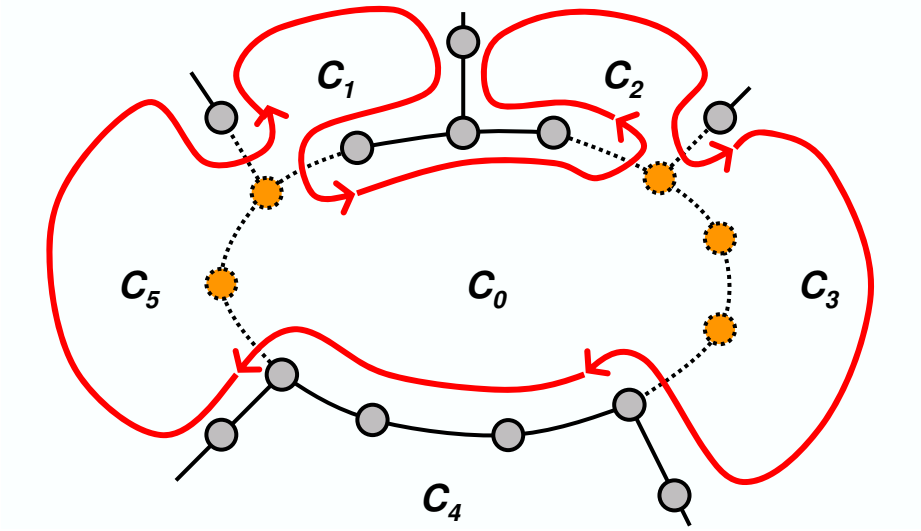
Step 1.4: Send message R to controller $m_{r'}$. Go to Step 3.

Step 2: Merge the cycles listed in message R. Go to Step 3.

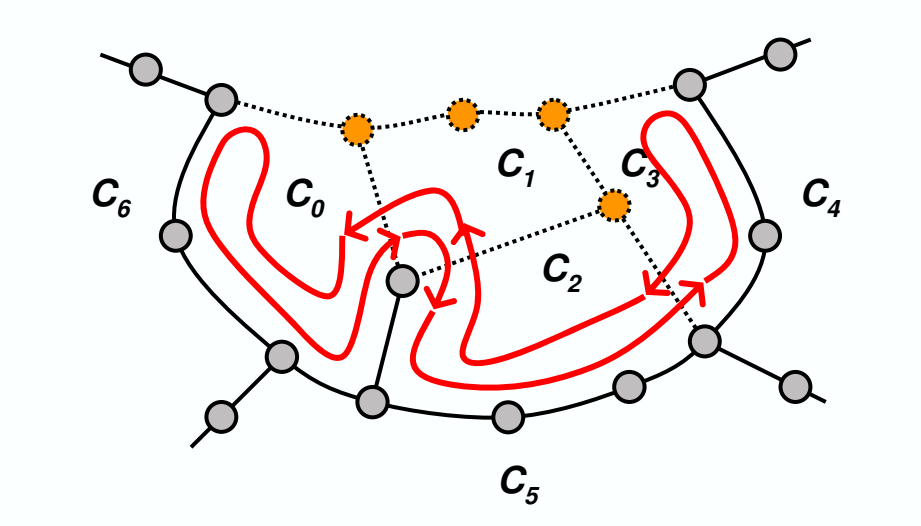
Step 3: Quit this process.

Each controller that detects a failure starts configuring a diverting cycle enclosing the failed part: The controller sends merge requests to the master controllers of adjacent cycles sharing the failed parts. Those requests are forwarded around the failed part as depicted by the solid arrows in Figure 5.4. On forwarding the merge requests, each master controller records the identifiers of its cycle and the border link of the failed part and the working part in the requests. The forwarded requests finally return to their generators. When generators receive their merge requests, they can figure out both the cycles to be merged and the outline of the diverting cycle, as in Figure 5.5.

After a cycle consolidation, a master controller for the diverting cycle should be selected from controllers of the merged cycles in a certain way, such as selecting one of the smallest ID. There are mainly two strategies for a master controller to control a diverting cycle. One is direct control: A certain controller chosen from those of the merged cycles establishes control channels to the OXCs included in the diverting cycle (Figure 5.6(a)). This strategy is simple but not scalable. It is difficult for only a controller to control the OXCs when the diverting cycle is large. In the other strategy, the diverting cycle are managed in hierarchical way as illustrated in Figure 5.6(b). Each cycle merged into the diverting cycle is managed by a controller. The controllers for the element cycles of the diverting cycle are managed by a master controller.



(a) Case of an inner massive failure

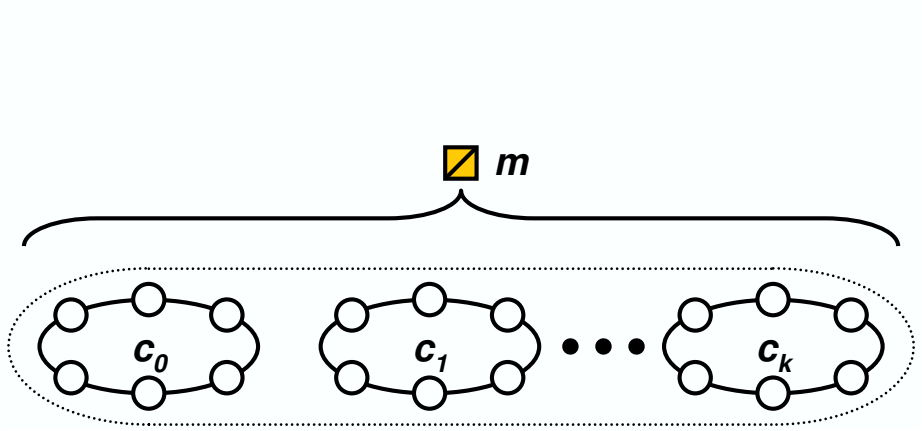


(b) Case of a border massive failure

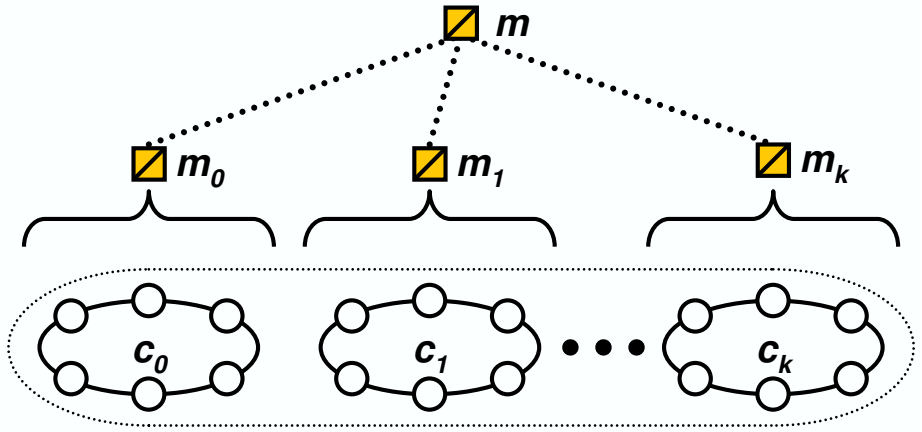
Figure 5.5: Outline of the merged cycle enclosing a failed region

5.2.4 Rerouting Lightpaths along a Diverting Cycle

A master controller of a diverting cycle obtains the information about failed sessions by the network failure by stacking the information written in merge request messages at each controller. The routes of failed sessions are straddling the diverting cycle. In Figure 5.1, the failed segments are illustrated with dotted lines. The master controller diverts these failed segments to the outline of the diverting cycle by assigning wavelengths. Note that



(a) Direct control



(b) Hierarchical control

Figure 5.6: Strategy of controlling a diverting cycle

the OXCs on the diverting cycle are managed by the master controller, not only distributed wavelength routing algorithms but also centralized algorithms are applicable.

If some of the failed sessions cannot be recovered by the local recovery with the diverting cycle due to the deficiency of wavelengths. In this case, the master controller sends notifications to controllers of upstream nodes of the failed segments or the source nodes for crank-back recovery or for end-to-end recovery.

5.3 Evaluation

In this section, we evaluate the performance of our proposed recovery scheme by computer simulations.

5.3.1 Simulation Model

We use a $n \times n$ lattice topology. Each link consists of a bi-directional fiber and has the same length. Each node consists of an OXC and a controller. The number of wavelengths w of a link is identical in the network and wavelength converters are not deployed. Lightpath requests are generated at random among the $4(n-1)$ edge nodes on the outline of the network topology. The number of lightpaths is given by the product of the number of the edge nodes and the load parameter l. We suppose that the edge nodes know the latest wavelength utilization at each link when they calculate routes. For these requested lightpaths, edge nodes choose shortest routes that at least a wavelength is available. Wavelengths are reserved by the backward reservation [28] (wavelengths to be reserved are chosen randomly from available ones). The holding time of the lightpaths is infinity. We consider only the propagation delay of control messages and ignore the message processing delay and switching delay. Therefore, we also control message losses never occurs. One unit time is equivalent to the propagation delay of a control message between adjacent controllers.

Under the circumstance, we make a $m \times m$-range massive OXC failure occur in the center of the network ($1 \leq m \leq n-2$) at time $t = 0$. Then, we compare the recovery time and the recovery rate between the case that the end-to-end path recovery scheme [17] is applied and the case that our recovery scheme is applied. In our scheme, a master controller of a diverting cycle assign wavelengths to disrupted lightpaths in increasing order of the hop length of the diverting segment.

5.3.2 Recovery Time and Recovery Rate

Figure 5.7 illustrates the recovery times and the recovery rates of our scheme and the end-to-end restoration scheme, where $n = 30$ and $w = 64$. The horizontal axes represent time

from a massive failure and the vertical axes represent the average recovery rate. The end-to-end restoration scheme, referred to as "e2e" in the figure, recovers lightpath connections gradually due to the end-to-end propagation delay of control messages and the blocking of the wavelength reservation. On the other hand, our scheme, referred to as "local", recovers many of disrupted lightpath connections simultaneously. In those cases, our scheme can recover the lightpath connections to more than 95%. However, our scheme utilizes the wavelength resources less effectively than the end-to-end restoration scheme because our scheme must assign the same wavelength for local recovery of a lightpath as the disrupted lighpath has used before and because the routes diverted by our scheme are not necessarily shortest routes among the available ones. Hence, our scheme cannot recover all the lightpath connections in some cases. To optimize the wavelength utilization and achieve as same recovery rate as the end-to-end recovery, we should optimize the routes of lightpaths by applying a reconfiguration algorithm such as [34].

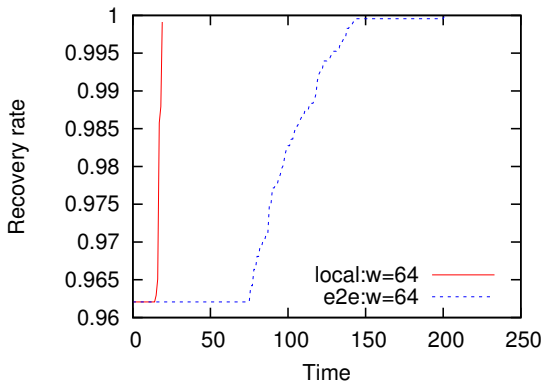
5.3.3 Number of Control Messages

Figure 5.8 illustrates the distribution of the average number of the control messages that each controller processes. The horizontal axes represent the rank of the controllers, which is the indices in the order of decreasing the number of processed control messages, and the vertical axes represent the frequency of the number of control messages processed by a controller. From these results, the end-to-end restoration scheme distributes more control messages over the network as the number of lightpaths or the scale of the massive failure gets larger. On the other hand, since most of control messages are for configuring the diverting cycle, our scheme restricts the increase of the number of control messages processed at nodes having nothing to do with the massive failure. The total numbers of control messages are shown in Figure 5.9. From these results, it is observed that our scheme can avoid the congestion in the control plane even when the scale of a massive failure is large.

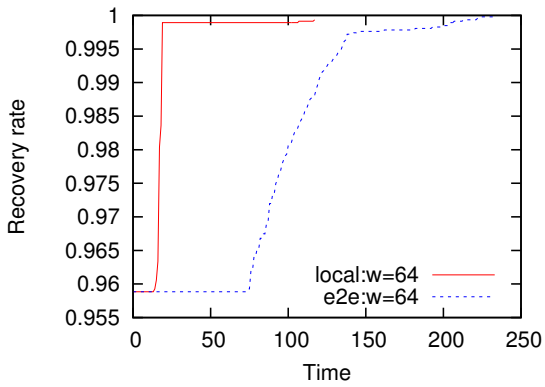
5.4 Summary

As the scale of wavelength-routed networks grows, the probability that a network failure occurs becomes larger. In addition, multiple failures could occur in certain regions of networks, called massive failures in this paper, due to natural disasters and so on. It is difficult to let all the nodes in a network get to know the location and the scale of a massive failure because the control plane is congested by a lot of control messages to inform all the nodes of failures or link state updates. In this paper, we propose a local restoration scheme to limit the number of distributed control messages after network failures. From the results of our evaluations, it is shown that the lightpath connections are quickly and recovered to almost 100% with our scheme when the scales of massive failures are not large. In addition, our scheme can avoid the congestion of the control plane even when the scales of massive failures are large. The congestion avoidance during the restoration is significant to progress the recovery process over the network smoothly.

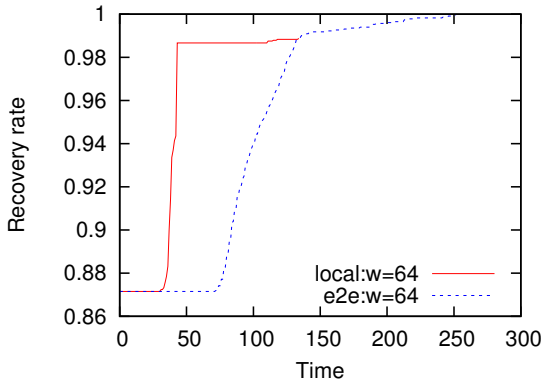
Our future work includes accelerating the speed of making up a diverting cycle. Currently, the order of time that our scheme takes to configure a diverting cycle for $m \times m$ massive failures is $O(m)$. By revising the order of the configuration speed, our scheme becomes more efficient to large-scale failures. We will also consider controller failures and design tolerant control planes against controller failures.



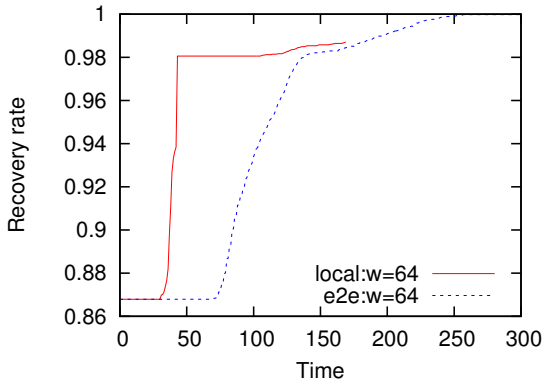
(a) $m = 2, l = 2.0$



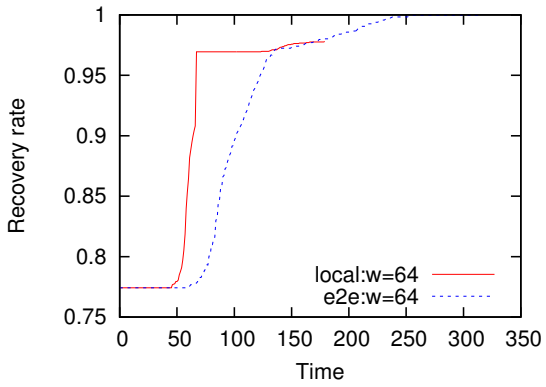
(b) $m = 2, l = 4.0$



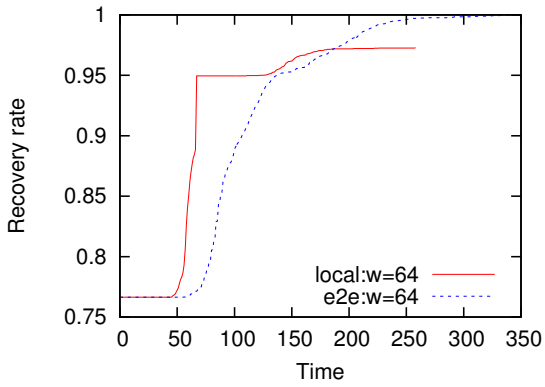
(c) $m = 6, l = 2.0$



(d) $m = 6, l = 4.0$

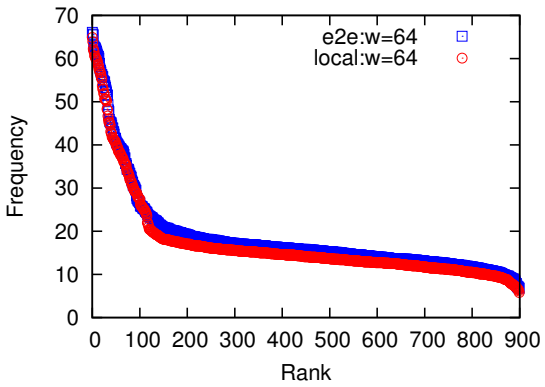


(e) $m = 10, l = 2.0$

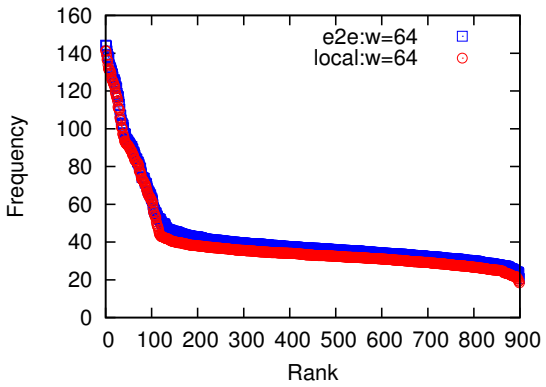


(f) $m = 10, l = 4.0$

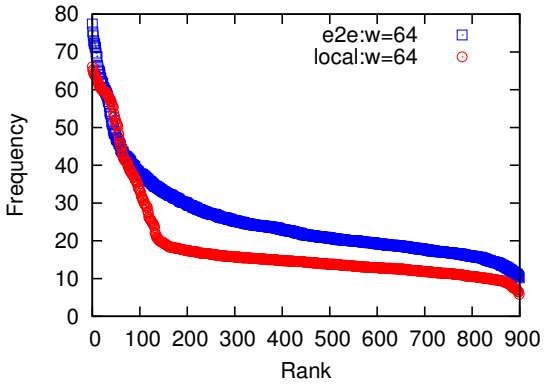
Figure 5.7: Recovery rate from a massive failure ($n = 30$, $w = 64$, averaged over 10 simulations)



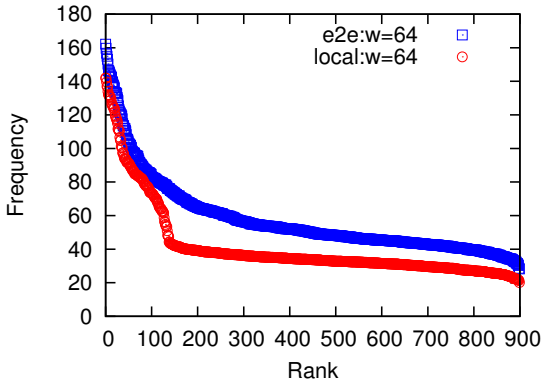
(a) $m = 2, l = 2.0$



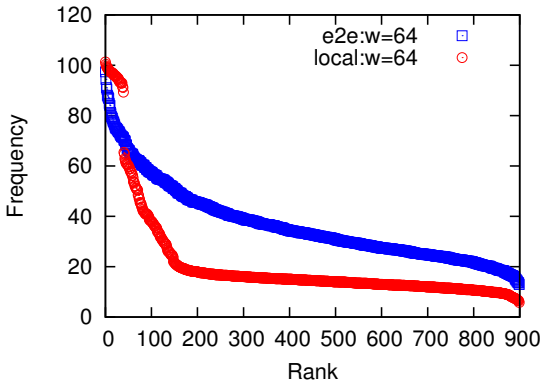
(b) $m = 2, l = 4.0$



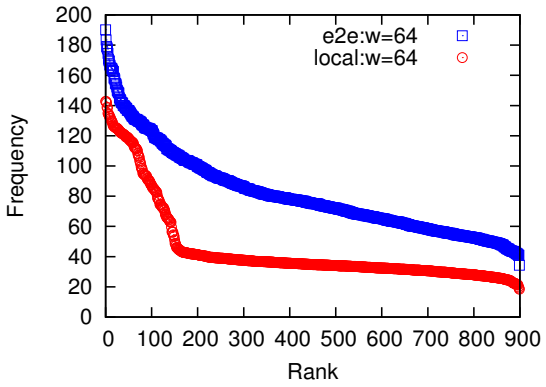
(c) $m = 6, l = 2.0$



(d) $m = 6, l = 4.0$

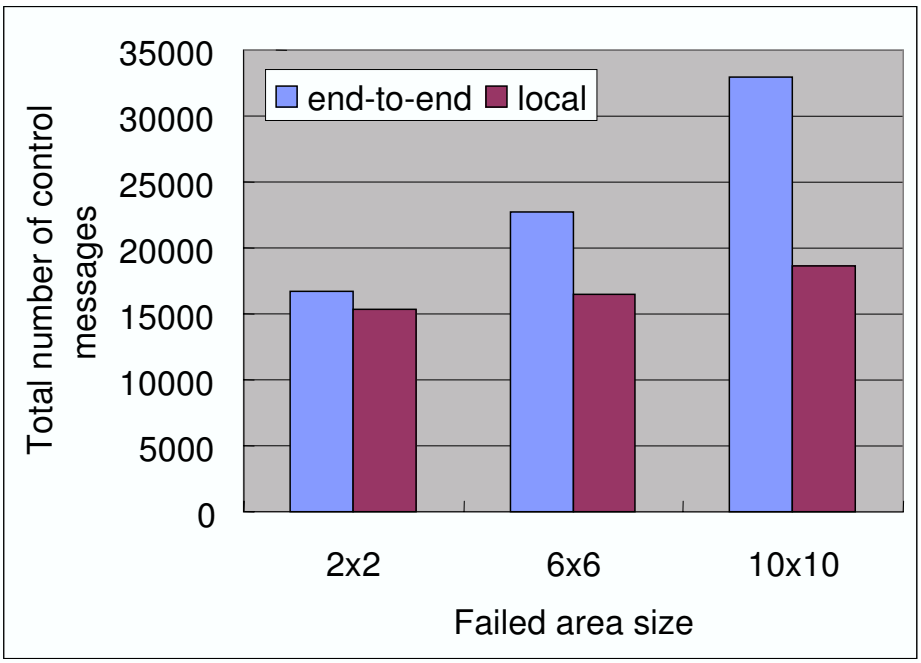


(e) $m = 10, l = 2.0$

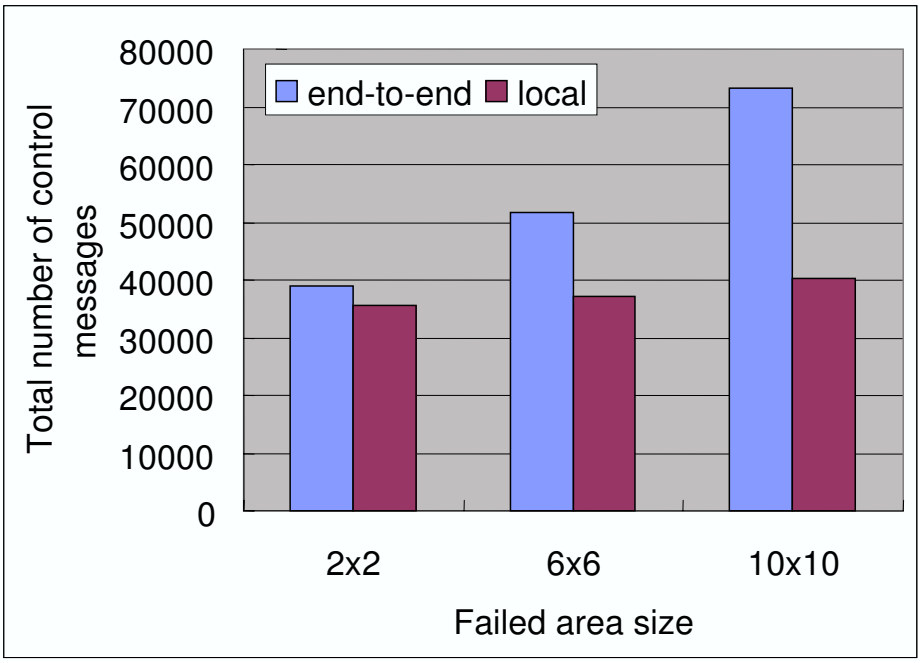


(f) $m = 10, l = 4.0$

Figure 5.8: Distribution of control messages ($n = 30$, $w = 64$, averaged over 10 simulations)



(a) $l = 2.0$



(b) $l = 4.0$

Figure 5.9: Total number of control messages (averaged over 10 simulations)

Chapter 6

Conclusion and Future Work

Wavelength-routed networks are high-capacity transport networks. Data communications over wavelength-routed network are based on the circuit-switching paradigm; nodes are connected with dedicated virtual circuits called lightpaths. The standardization of the control architectures for wavelength-routed networks has been progressed to interconnect wavelength-routed networks. As a result, large-scaled wavelength-routed networks are comprised. Due to this enlargement, the volume of the traffic carried over the networks increases. The probability that a network failure occurs gets higher because of the increase of the number of network components. In addition, the amount of the information for managing networks also increases. Hence, flexibility, reliability, and scalability are significant property for large-scaled wavelength-routed networks.

In Chapter 2, we propose a reconfiguration algorithm for wavelength-routed mesh networks to provide flexible and reliable backbones. Our basic idea is to use wavelength resources reserved for backup lightpaths which are not always utilized. Our algorithm is based on five procedures to set up and tear down lightpaths. In addition to simply setting up or tearing down lightpaths, we have considered three other procedures to incorporate wavelength resources for backup lightpaths. Since the backup lightpaths are not always used for transporting the actual traffic, we exploit their wavelength resources assuming that failure does not occur during reconfiguration.

In Chapter 3, we propose a scalable routing scheme for large-scaled wavelength-routed networks. To achieve this objective, we at first investigate the structure and the property of

the network topology of large-scaled wavelength-routed networks. According to the analogy between the process of the Internet's growth and that of wavelength-routed networks' growth, it is speculated that the physical topologies of large-scaled wavelength-routed networks have the power-law connectivity. In the networks having the power-law connectivity, most of the nodes have just a few links although some nodes have a number of links. We construct logical topologies over physical topologies by configuring virtual fibers and route lightpaths in logical topologies, not in physical topologies. By adopting our method, performances of WDM networks with the power-law connectivity are improved without any cost for network equipments and link state based routings.

In Chapter 4, we evaluate the performance of GMPLS RSVP-TE; we investigate how control parameter settings affect the performance of GMPLS RSVP-TE and when the message retransmission of GMPLS RSVP-TE works effectively. To more precisely understand the influence of each control parameter to the network performance and the relation between control parameter settings, we extend the Markov model in [51] for GMPLS RSVP-TE. Using the Markov model, we describe the behavior of GMPLS RSVP-TE in detail and analyze the steady-state probabilities of a lightpath session. We then investigate the network performance, such as resource utilization and LSP setup delay of GMPLS RSVP-TE.

In Chapter 5, we propose a local restoration scheme to protect the control plane from the influence of massive failures. This scheme locally configures a cycle enclosing the failed part in a distributed manner. By using this diverting cycle, most of the failed lightpaths are locally recovered at a time. Thus, the number of control messages distributed into the network is reduced. The results of computer simulations show that our scheme recovers the lightpath connectivity to almost 100% more quickly than the path restoration scheme when the scale of massive failures is not large. When the scale of massive failures is large, our scheme reduces the number of control messages to about the half comparing to the end-to-end path restoration scheme.

The future work includes finding a scheme for lightpath management to enhance the resource utilization and the reliability of hierarchical wavelength-routed networks based on GMPLS and ASON architectures. Another future work is to find a scheme for management of the control planes that are resilient to failures on controllers.

Finally, we believe that those above discussions contribute to the design and management of future wavelength-routed networks widespread around the world.

Bibliography

[1] L. G. Roberts, "Beyond Moore's law: Internet growth trends," *IEEE Computer*, vol. 33, no. 1, pp. 117–119, Jan. 2000.

[2] K. G. Coffman and A. M. Odlyzko, "Internet growth: Is there a "Moore's Law" for data traffic?," in *Handbook of Massive Data Sets*, ch. 1, pp. 47–93, Kluwer Academic Publishers, Mar. 2002.

[3] B. Mukherjee, *Optical Communication Networks*. McGraw-Hill, July 1997.

[4] B. Mukherjee, *Optical WDM Networks*. Springer, Jan. 2006.

[5] A. Sano, H. Masuda, Y. Kisaka, S. Aisawa, E. Yoshida, Y. Miyamoto, M. Koga, K. Hagimoto, T. Yamada, T. Furuta, and H. Fukuyama, "14-tb/s (140 x 111-gb/s PDM/WDM CSRZ-DQPSK transmission over 160 km using 7-thz bandwidth extended L-band EDFAs," in *European Conference on Optical Communication (ECOC 2006)*, (Cannes, France), Sept. 2006.

[6] E. Mannie, "Generalized Multi-Protocol Label Switching (GMPLS) architecture," *RFC 3945*, Oct. 2004.

[7] G. Li, J. Yates, D. Wang, and C. Kalmanek, "Control plane design for reliable optical networks," *IEEE Communications Magazine*, vol. 40, no. 2, pp. 90–96, Feb. 2002.

[8] A. Jajszczyk and P. Rozycki, "Recovery of the control plane after failures in ASON/GMPLS networks," *IEEE Network*, vol. 20, no. 1, pp. 4–10, Jan. 2006.

[9] ITU-T, "Architecture for the automatically switched optical networks (ASON)," *Recommendation G.8080/Y.130411*, Nov. 2001.

[10] ITU-T, "G.8080 amendment 1," *Recommendation G.8080/Y.1304*, Mar. 2003.

[11] R. Ramaswami and K. N. Sivarajan, "Design of logical topologies for wavelength-routed optical networks," *IEEE Journal on Selected Areas in Communications*, vol. 14, no. 5, pp. 840–851, June 1996.

[12] R. Dutta and G. N. Rouskas, "A survey of virtual topology design algorithms for wavelength routed optical networks," *Optical Network Magazine*, vol. 1, no. 1, pp. 73–89, Jan. 2000.

[13] M. Murata, "Challenges for the next-generation Internet and the role of IP over photonic networks," *IEICE Transactions on Communications*, vol. E83-B, no. 10, pp. 2153–2165, Oct. 2000.

[14] S. Arakawa and M. Murata, "On incremental capacity dimensioning for reliable IP over WDM networks," in *Proceedings of Optical Networking and Communications Conference (Opticomm 2001)*, vol. 4599, (Denver, CO), pp. 153–162, Aug. 2001.

[15] S. Arakawa and M. Murata, "Lightpath management of logical topology with incremental traffic changes for reliable IP over WDM networks," *Optical Network Magazine*, vol. 3, no. 3, pp. 68–76, May 2002.

[16] G. Mohan and C. S. R. Murthy, "Lightpath restoration in WDM optical networks," *IEEE Network*, vol. 14, no. 6, pp. 16–23, Nov. 2000.

[17] S. Ramamurthy, L. Sahasrabuddhe, and B. Mukherjee, "Survivable WDM mesh networks," *Journal of Lightwave Technology*, vol. 21, no. 4, pp. 870–883, Apr. 2003.

[18] G. Shen and W. D. Grover, "Extending the p-cycle concept to path segment protection for span and node failure recovery," *IEEE Journal on Selected Areas in Communications*, vol. 21, no. 9, pp. 1306–1319, Oct. 2003.

[19] P. H. Ho and H. T. Mouftah, "A framework for service-guaranteed shared protection in WDM mesh networks," *IEEE Communications Magazine*, vol. 40, no. 2, pp. 97–103, Feb. 2003.

[20] S. Koo and S. Subramaniam, "Performance evaluation of optical mesh restoration schemes," *Information Sciences*, vol. 149, no. 1–3, pp. 183–195, Jan. 2003.

[21] I. Chlamtac, A. Ganz, and G. Karmi, "Lightpath communications: An approach to high bandwidth optical WAN's," *IEEE Transactions on Communications*, vol. 40, no. 7, pp. 1171–1182, July 1992.

[22] B. Mukherjee, D. Banerjee, S. Ramamurthy, and A. Mukherjee, "Some principles for designing a wide-area WDM optical network," *IEEE/ACM Transactions on Networking*, vol. 4, no. 5, pp. 684–695, Oct. 1996.

[23] J. Bannister, J. Touch, A. Willner, and S. Suryaputra, "How many wavelengths do we really need? A study of the performance limits of packet over wavelength," *Optical Networks Magazine*, vol. 1, no. 2, pp. 11–28, Apr. 2000.

[24] R. H. Cardwell, O. J. Wasem, and H. Kobrinski, "WDM architectures and economics in metropolitan areas," *Optical Network Magazine*, vol. 1, no. 3, pp. 41–50, July 2000.

[25] Cisco Systems, Inc., "Comparing metro WDM systems: Unidirectional vs. bidirectional implementations," *White Paper*, Jan. 2001.

[26] F. Bruyére, "Metro WDM," *Alcatel Telecommunications Review*, Mar. 2002.

[27] R. Ramaswami and A. Segall, "Distributed network control for optical networks," *IEEE/ACM Transactions on Networking*, vol. 5, no. 6, pp. 936–943, Dec. 1997.

[28] X. Yuan, R. Melhem, and R. Gupta, "Distributed path reservation algorithms for multiplexed all-optical interconnection networks," *IEEE Transactions on Computers*, vol. 48, no. 12, pp. 1355–1363, Dec. 1999.

[29] K. Lu, J. P. Jue, G. Xiao, I. Chlamtac, and T. Ozugur, "Intermediate-node initiated reservation (IIR): A new signaling scheme for wavelength-routed networks," *IEEE Journal on Selected Areas in Communications*, vol. 21, no. 8, pp. 1285–1294, Oct. 2003.

[30] H. Zang, J. P. Jue, and B. Mukherjee, "A review of routing and wavelength assignment approaches for wavelength-routed optical WDM networks," *Optical Network Magazine*, vol. 1, no. 1, pp. 47–60, Jan. 2000.

[31] A. Jukan and G. Franzl, "Path selection methods with multiple constraints in service-guaranteed WDM networks," *IEEE/ACM Transactions on Networking*, vol. 12, no. 1, pp. 59–72, Feb. 2004.

[32] J. Ji, G. Mohan, E. C. Tien, and K. C. Chua, "Dyanamic routing with inaccurate link state information in integrated IP-over-WDM networks," *Computer Networks*, vol. 46, no. 6, pp. 829–851, Dec. 2004.

[33] S. Arakawa, T. Toku, and M. Murata, "Evaluation of routing algorithms for distributed lightpath establishment in wavelength-routed networks," in *Proceedings of Workshop on IEEE/CreateNet Workshop on Guaranteed Optical Service Provisioning (GOSP 2005)*, (Boston, MA), pp. 358–367, Oct. 2005.

[34] 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, no. 3, pp. 265–277, Nov. 2003.

[35] S. Ishida, S. Arakawa, and M. Murata, "Proposal of procedures to reconfigure logical topologies in reliable WDM-based mesh networks," in *Proceedings of SPIE Asia-Pacific Optical and Wireless Communications (APOC 2002) Optical Networking II*, vol. 4910, (Shanghai, China), pp. 115–125, Oct. 2002.

[36] S. Ishida, S. Arakawa, and M. Murata, "Dynamic reconfiguration of logical topologies in WDM-based mesh networks," in *Proceedings of the 7th IFIP Working Conference on Optical Network Design and Modelling (ONDM2003)*, vol. 1, (Budapest, Hungary), pp. 93–112, Feb. 2003.

[37] S. Ishida, S. Arakawa, and M. Murata, "An algorithm to reconfigure logical topologiees in reliable WDM networks," *Technical Report of IEICE (PS2002-1)*, pp. 49–54, Apr. 2002. (*in Japanese*).

[38] I. Baldine and G. N. Rouskas, "Dynamic reconfiguration policies in multihop WDM networks," *Journal of High Speed Networks*, vol. 4, no. 3, pp. 221–238, 1995.

[39] J.-F. P. Labourdette and A. S. Acampora, "Logically rearrangeable multihop lightwave networks," *IEEE Transactions on Communications*, vol. 39, no. 8, pp. 1223–1230, Aug. 1991.

[40] I. Baldine and G. N. Rouskas, "Dynamic load balancing in broadcast WDM networks with tuning latencies," in *Proceedings of 17th Annual Joint Conference of the IEEE Computer and Communications Societies (Infocom '98)*, (San Francisco, CA), pp. 78–85, Mar. 1998.

[41] J.-F. P. Labourdette, G. W. Hart, and A. S. Acampora, "Branch-exchange sequences for reconfiguration of lightwave networks," *IEEE Transactions on Communications*, vol. 42, no. 10, pp. 2822–2832, Oct. 1994.

[42] A. Narula-Tam and E. Modiano, "Dynamic load balancing in WDM packet networks with and without wavelength constraints," *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 10, pp. 1972–1979, Oct. 2000.

[43] S. Ishida, S. Arakawa, and M. Murata, "Virtual fiber configuration for dynamic lightpath establishment in large-scaled optical networks," *Photonic Network Communications*, vol. 12, no. 1, pp. 87–98, July 2006.

[44] S. Ishida, S. Arakawa, and M. Murata, "Virtual fiber configuration method for dynamic lightpath establishment in large-scaled WDM networks," in *Proceedings of 9th Conference on Optical Network Design and Modelling (ONDM 2005)*, (Milan, Italy), pp. 153–161, Feb. 2005.

[45] S. Ishida, S. Arakawa, and M. Murata, "On a power-law relationship in wavelength-routed networks," *Technical Report of IEICE (PN2003-27)*, pp. 13–16, Dec. 2003. (*in Japanese*).

[46] S. Ishida, S. Arakawa, and M. Murata, "Quasi-static lightpath configuration method in large-scaled WDM networks," *Technical Report of IEICE (CS2004-8)*, pp. 37–42, May 2004.

[47] S. Ishida, S. Arakawa, and M. Murata, "Analyses of soft-state signaling protocols in GMPLS-based WDM networks," submitted to *IEEE/OSA Journal of Lightwave Technology*, July 2006.

[48] S. Ishida, S. Arakawa, and M. Murata, "Performance analysis of soft-state lightpath management in GMPLS-based WDM networks," in *Proceedings of Third International Conference on Broadband Communications, Networks, and Systems (Broadnets 2006)*, (San José, CA), Oct. 2006.

[49] S. Ishida, S. Arakawa, and M. Murata, "Performance analysis of soft-state signaling protocols in wavelength-routed networks," *Technical Report of IEICE (PN2005-46)*, pp. 1–6, Dec. 2005. (*in Japanese*).

[50] L. Berger, "Generalized Multi-Protocol Label Switching (GMPLS) signaling Resource ReSerVation Protocol-Traffic Engineering (RSVP-TE) extensions," *RFC 3473*, Jan. 2003.

[51] P. Ji, Z. Ge, J. Kurose, and D. Towsley, "A comparison of hard-state and soft-state signaling protocols," in *Proceedings of ACM SIGCOMM '03*, (Karlsruhe, Germany), pp. 251–262, Aug. 2003.

[52] S. Ishida, S. Arakawa, and M. Murata, "Local recovery from massive failures in large-scaled WDM networks," submitted to *11th Conference on Optical Network Design and Modelling (ONDM 2007)*, Dec. 2006.

[53] M. T. Frederick, P. Datta, and K. Somani, "Sub-graph routing: A generalized fault-tolerant strategy for link failures in WDM optical networks," *Computer Networks*, vol. 50, no. 2, pp. 181–199, Feb. 2006.

[54] P. Erdös and A. Rényi, "On the evolution of random graphs," *Publications of the Mathematical Institute of the Hungarian Academy of Sciences*, vol. 5, pp. 17–61, 1960.

[55] A.-L. Barabási and R. Albert, "Emergence of scaling in random networks," *Science*, vol. 286, no. 15, pp. 509–512, Oct. 1999.

[56] M. E. J. Newman, "Random graphs as models of networks," in *Handbook of Graphs and Networks*, ch. 2, pp. 35–68, WILEY-VCH, Nov. 2002.

[57] L. Berger, D. Gan, G. Swallow, P. Pan, F. Tommasi, and S. Molendini, "RSVP refresh overhead reduction extensions," *RFC 2961*, Apr. 2001.

[58] R. Braden, L. Zhang, S. Berson, S. Herzog, and S. Jamin, "Resource ReSerVation Protocol (RSVP) - version 1 functional specification," *RFC 2205*, Sept. 1997.

[59] Z. Zhou and D. Gao, "An efficient adaptation of RSVP-TE in GMPLS," in *Proceedings of the 2004 Internatinal Symposium of Performance Evaluation of Computer and Telecommunication Systems (SPECTS 2004)*, (San José, CA), pp. 93–97, July 2004.

[60] H. Wang, R. Karri, M. Veeraraghavan, and T. Li, "A hardware-accelerated implementation of the RSVP-TE signaling protocol," in *Proceedings of IEEE International Conference of Communications (ICC 2004)*, vol. 27, (Paris, France), pp. 1609–1614, June 2004.

Appendix A

Description of the State Transition of RSVP-TE for h-Hop LSP

We explain the operations of RSVP-TE at each state of the Markov chain in Fig. 4.6 below, skipping the explanations of states S_{6h+1} to S_{12h-1} since the transitions among these states are same as the transitions among the states S_1 to S_{6h-1}.

S_0: The initial state. When an LSP setup request arrives at a source node, the Markov chain goes to S_1.

S_1: The source node makes a Path state and sends a Path trigger message downstream. If the message is lost, the Markov chain goes to S_2. If a downstream node receives the message and there is an available label, the Markov chain goes to S_3. If a downstream node receives the message but there is no available label, the Markov chain goes to S_5.

S_2: The source node sends a Path refresh message. If a downstream node receives the message and there is an available label, the Markov chain goes to S_3. If the downstream node receives the message but there is no available label, the Markov chain goes to S_5.

S_{3j}: Each intermediate node makes a Path state and sends a Path trigger message. If the downstream node receives the message and there is an available label, the Markov chain goes to S_{3j+3}. If the downstream node receives the

message and there is no available label, the Markov chain goes to S_{3j+5}. If the message is lost, the Markov chain goes to S_{3j+1}. $j = 1, 2, \cdots, h-1$.

S_{3j+1}: Each intermediate node sends a Path refresh message. If a downstream node receives the message and there is an available label, the Markov chain goes to S_{3j+3}. If a downstream node receives the message and there is no available label, the Markov chain goes to S_{3j+5}. $j = 1, 2, \cdots, h-1$.

S_{3j+2}: Each intermediate node sends a PathErr message. the Markov chain goes to S_{3j-1}. $j = 1, 2, \cdots, h-1$.

S_{3h}: A destination node creates a Path state. The destination node also creates a Resv state and sends a Resv trigger message. If an upstream node receives the message and reserves a label, the Markov chain goes to S_{3h+3}. If an upstream node fails to reserve a label, the Markov chain goes to S_{3h+5}. If the message is lost, the Markov chain goes to S_{3h+1}.

S_{3h+1}: The destination node sends a Resv refresh message. If an upstream node receives the message and reserves a label, the Markov chain goes to S_{3h+3}. If an upstream node fails to reserve a label, the Markov chain goes to S_{3h+5}.

S_{3h+2}: The destination node sends a PathErr message. The Markov chain goes to S_{3h-1}.

S_{3h+3j}: Each intermediate node sends a Resv trigger message. If an upstream node receives the message and reserves a label, the Markov chain goes to $S_{3h+3j+3}$. If an upstream node fails to reserve a label, the Markov chain goes to $S_{3h+3j+5}$. If the message is lost, the Markov chain goes to $S_{3h+3j+1}$. $j = 1, 2, \cdots, h-2$.

$S_{3h+3j+1}$: Each intermediate node sends a Resv refresh message. If an upstream node receives the message and reserves a label, the Markov chain goes to $S_{3h+3j+3}$. If an upstream node fails to reserve a label, the Markov chain goes to $S_{3h+3j+5}$. $j = 1, 2, \cdots, h-2$.

$S_{3h+3j+2}$: Each intermediate node sends a ResvErr message downstream. The Markov chain goes to $S_{3h+3j-1}$. $j = 1, 2, \cdots, h-1$.

S_{6h-3}: An intermediate node sends a Resv trigger message to the source node. If the source node receives the message, the Markov chain goes to S_{6h}. Otherwise, the Markov chain goes to S_{6h-2}.

S_{6h-2}: An intermediate node sends a Resv refresh message to the source node. If the source node receives the message, the Markov chain goes to S_{6h}.

S_{6h}: An LSP is established in this state. If the data transmission is completed, the Markov chain goes to S_{12h}. If a Path state at the first node from the source node is deleted by false removal, the Markov chain goes to S_{6h+2}. If a Path state at the i-th node from the source node is deleted by false removal, the Markov chain goes to $S_{6h+3j-2}$ ($j = 2, 3, \cdots, h$). If a Resv state at the i-th node from the destination node is deleted by false removal, the Markov chain goes to $S_{9h+3j-2}$ ($j = 1, 2, \cdots, h$).

S_{12h}: The source node sends a PathTear message. If a downstream node receives the message, the Markov chain goes to S_{12h+2}. If the message is lost, the Markov chain goes to S_{12h+1}.

S_{12h+1}: A Path state at the node next to a source node is deleted by state timeout. The Markov chain goes to S_{12h+2}.

S_{12h+2j}: Each intermediate node sends a PathTear message. If a downstream node receives the message, the Markov chain goes to $S_{12h+2j+2}$. If the message is lost, the Markov chain goes to $S_{12h+2j+1}$. $j = 1, 2, \cdots, h-2$.

$S_{12h+2j+1}$: A Path state at a i-th node is deleted by state timeout. The Markov chain goes to $S_{12h+2j+2}$. $j = 1, 2, \cdots, h-2$.

S_{14h-2}: A Path state at the penultimate node sends a PathTear message. If the destination node receives the message, the Markov chain goes to S_0. If the message is lost, the Markov chain goes to S_{14h-1}.

S_{14h-1}: A Path state at the destination node is deleted by state timeout. The Markov chain goes to S_0.