Attractor-based Virtual Network Reconfiguration Under Dynamic Traffic: Towards Cognitive Optical Networking

Toshihiko Ohba

January 2018

List of publication

Journal papers

1. Toshihiko Ohba, Shin'ichi Arakawa, Yuki Koizumi, and Masayuki Murata, "Scalable Design Method of Attractors in Noise-induced Virtual Network Topology Control," *IEEE/OSA Journal of Optical Communications and Networking*, vol. 7, pp. 851–863, September 2015.

2. Toshihiko Ohba, Shin'ichi Arakawa, and Masayuki Murata, "Virtual Network Reconfiguration in Elastic Optical Path Networks for Future Bandwidth Allocation," *IEEE/OSA Journal of Optical Communications and Networking*, vol. 8, pp. 633–644, September 2016.

Refereed Conference Papers

1. Toshihiko Ohba, Shin'ichi Arakawa, Yuki Koizumi, and Masayuki Murata, "Hierarchical Design of an Attractor Structure for VNT Control Based on Attractor Selection," in *Proceedings of IEEE Consumer Communications and Networking Conference*, pp. 330–336, January 2015.

2. Toshihiko Ohba, Shin'ichi Arakawa, and Masayuki Murata, "A Bayesian-based Approach for Virtual Network Reconfiguration in Elastic Optical Path Networks," in *Proceedings of Optical Fiber Communication Conference*, pp. Th1J–7, March 2017.

Non-Refereed Technical Papers

1. Toshihiko Ohba, Shin'ichi Arakawa, Yuki Koizumi, and Masayuki Murata, "Design and Control of an Attractor Structure for Virtual Network Topology Control Based on Attractor Selection," *Technical Report of IEICE (PN2012-79)*, vol. 112, pp. 7–12, March 2013 (in Japanese).

2. Toshihiko Ohba, Shin'ichi Arakawa, Yuki Koizumi, and Masayuki Murata, "Evaluation of Diversity of Attractors for Virtual Network Topology Control Based on Attractor Selection," *Technical Report of IEICE (PN2013-15)*, vol. 113, pp. 41–46, August 2013 (in Japanese).

3. Toshihiko Ohba, Shin'ichi Arakawa, Yuki Koizumi, and Masayuki Murata, "Hierarchical Design of an Attractor Structure with Topological Diversity for VNT Control Based on Attractor Selection," *Technical Report of IEICE (PN2014-18)*, vol. 114, pp. 43–48, September 2014 (in Japanese).

4. Toshihiko Ohba, Shin'ichi Arakawa, and Masayuki Murata, "Noise-induced Virtual Network Topology Control for Elastic Optical Networks," *Technical Report of IEICE (PN2015-33)*, vol. 115, pp. 55–60, November 2015 (in Japanese).

5. Toshihiko Ohba, Shin'ichi Arakawa, and Masayuki Murata, "A Bayesian-based Virtual Network Reconfiguration in Elastic Optical Path Networks," *Technical Report of IEICE (PN2016-33)*, vol. 116, pp. 45–50, November 2016 (in Japanese).

6. Toshihiko Ohba, Shin'ichi Arakawa, and Masayuki Murata, "Virtual Network Reconfiguration Based on Bayesian Attractor Model with Linear Regression," *Technical Report of IEICE (PN2017-37)*, vol. 117, pp. 57–63, November 2017 (in Japanese).

Preface

Thanks to the large bandwidth of optical fibers, optical networks have the potential to support growing Internet traffic. However, the emergence of new network services with wide range of required bandwidth has caused large traffic fluctuations. One approach for the network operator to accommodate fluctuating traffic demand on an optical network is to construct a virtual network (VN), and reconfigure a VN following traffic changes. When traffic fluctuations cause temporal traffic congestion, the network operator resolves the traffic congestion by reconfiguring the VN. Although typical VN reconfiguration methods design an optimal VN using the end-to-end traffic demand matrix, they have difficulty in adapting to traffic fluctuations.

Our research group has previously proposed a VN reconfiguration approach based on attractor selection, which models the behavior where living organisms adapt to unknown changes in their surrounding environment and recover their condition, that observes only the service quality on a VN (i.e., link-level load information), which can be retrieved in a much shorter time than the traffic demand matrix. Attractors correspond to VN candidates, and this approach reconfigures a VN guided by the attractors, that is, this approach gradually reconfigures a VN to have a network topology close to one of the VN candidates. However, there are several problems to be solved in this approach. First, there is no guideline on how to design attractors. It is crucial to design attractors properly since this approach reconfigures a VN guided by the attractors. When the attractors are not designed properly, it takes a long time to find a solution (i.e., a VN that can accommodate traffic demand). Second, this approach cannot applied to elastic optical path networks, which have been

shown to be a promising candidate for future resource-efficient optical networks. Although elastic optical path networks can achieve higher utilization efficiency of spectrum resources than traditional Wavelength Division Multiplexing (WDM)-based networks, it is essential to tackle the problem of allocating spectrum resources to lightpaths under the spectrum contiguity and continuity constraints. Third, simply applying this approach may over-reconfigure a VN in nature, which disrupts network services accommodated on the VN.

In this thesis, to solve the above problems, we investigate an attractor-based VN reconfiguration framework for optical networks. First, we propose a design method of attractors (i.e., VN candidates). In this method, we design attractors with a wide variety characteristics so that the attractor selection-based VN reconfiguration can quickly adapt to various traffic fluctuations. Our basic approach is to prepare VN candidates which the bottleneck links (lightpaths) are different from each other. However, our exhaustive algorithm based on this approach has a problem of requiring large amounts of computational time for large-scale networks. To solve this problem, we also propose a method that hierarchically contracts a network topology so that our algorithm can be applied to large-scale networks. Evaluation results show that the VN reconfiguration using attractors obtained by our design method finds a solution in a shorter time.

Second, we propose an attractor selection-based VN reconfiguration method for elastic optical path networks. We newly define the potential bandwidth as a metric that reflects the bandwidth that can be additionally offered under the spectrum contiguity and continuity constraints. Then, our method reconfigures a VN based on attractor selection so that both the service quality on the VN and the potential bandwidth get improved. In addition, our method adjusts the bandwidth according to the link utilization of lightpaths that form the VN to provide the required bandwidth. Evaluation results show that the proposed method can set aside about 50 % of resources for future use, while improving the service quality on a VN to the same extent as the existing heuristic method, considering the spectrum contiguity and continuity constraints.

Third, to reduce the number of VN reconfigurations, we introduce a cognitive mechanism that perceives current traffic situation and adapts to the situation. Specifically, we propose another VN reconfiguration method based on the Bayesian Attractor Model (BAM), which models the human behavior of making appropriate decisions by recognizing the surrounding situation, and we establish a VN reconfiguration framework that deals with known traffic situations by the BAM-based method and deals with unknown traffic situations by the attractor selection-based method. The BAM-based method memorizes a set of VN candidates, each of which works well for a pre-specified traffic situation, and identifies the current traffic situation using the BAM, and then retrieves the most promising VN from the set. We use certain patterns of incoming and outgoing traffic at edge routers as the traffic situation, since this information can be obtained more easily than the traffic demand matrix. However, for the case where the retrieved VN cannot accommodate traffic demand, we apply the attractor selection-based method. Evaluation results show that the VN reconfiguration framework can reach a VN suitable for the current traffic situation with fewer VN reconfigurations. Furthermore, we extend the above VN reconfiguration framework to deal with the case where the identification of the current traffic situation fails. When the identification fails, the BAM-based method cannot configure a promising VN. We therefore propose a method that configures a promising VN when the identification fails, and incorporate this method into the above VN reconfiguration framework. The proposed method utilizes a set of pre-specified traffic situations, and fits the current traffic situation by linear regression, and then configures a VN using the obtained regression coefficients. Evaluation results show that the linear regression-based method can configure a suitable VN in most cases when failing to identify the current traffic situation. We also investigate how to select and update the set of pre-specified traffic situations, and found that it is effective to select a set of pre-specified traffic situations to have linear independence.

Acknowledgments

I would like to express my sincere appreciation to everyone who supported me in various ways throughout my Ph.D. This thesis could not have been accomplished without their assistance.

First of all, I express my grate gratitude to my supervisor, Professor Masayuki Murata of Graduate School of Information Science and Technology, Osaka University, for his insightful suggestions and valuable discussions. He brought me to an attractive research field, and made my research life fruitful.

I am heartily grateful to the members of my thesis committee, Professor Takashi Watanabe, Professor Teruo Higashino, and Professor Toru Hasegawa of Graduate School of Information Science and Technology, Osaka University, and Professor Morito Matsuoka of Cyber Media Center, Osaka University, for their multilateral reviews and perceptive comments.

Furthermore, I would like to owe my special thanks to Associate Professor Shin'ichi Arakawa of Graduate School of Information Science and Technology, Osaka University, for his continuous support and encouragement. He taught me the fun of thinking about and solving problems.

Moreover, I am deeply grateful to Assistant Professor Yuichi Ohsita, Assistant Professor Yuki Koizumi, Specially Appointed Assistant Professor Naomi Kuze, Specially Appointed Assistant Professor Tatsuya Otoshi of Graduate School of Information Science and Technology, Osaka University, and Assistant Professor Daichi Kominami of Graduate School of Economics, Osaka University, for their valuable comments on my study.

I express my appreciation to all of past and present colleagues, friends, and secretaries of

the Advanced Network Architecture Research Laboratory, Graduate School of Information Science and Technology, Osaka University.

Finally, I cannot conclude my acknowledgement without expressing my thanks to my parents and family. Thank you for your giving me invaluable supports throughout my life.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

The Internet is indispensable to our society since it allows us to communicate with remote areas, as well as provides an infrastructure for various businesses today. In recent years, new network services and applications, such as video streaming and cloud computing, have emerged one after another. This change of environments surrounding the Internet has led to drastic growth in Internet traffic. Specifically, the volume of Internet traffic has grown by 20 times over the past decade [1]. Furthermore, various kinds of devices, such as personal computers, mobile phones and sensing devices, have come to connect to the Internet. With the development of new Internet technologies and services that utilize Internet of Things (IoT) and Artificial Intelligence (AI), Machine-to-Machine (M2M) communications also have more presence, and this trend will continue for the future.

Thanks to the large bandwidth of optical fibers, optical networks have the potential to support this growing traffic demand. The development of optical communication technologies has enabled optical networks to offer much higher bandwidth [2–5]. However, the emergence of new network services and applications with wide range of required bandwidth has caused large fluctuations in traffic demand. This results in creating a gap between the bandwidth provided by optical networks and the bandwidth required by network users.

Although network operators can support traffic fluctuations by providing larger bandwidth than the bandwidth actually used, it leads to inefficient utilization of network resources. Therefore, it is essential that optical networks have flexibility to accommodate such fluctuating traffic demand by dynamically providing the sufficient bandwidth with limited network resources.

Network virtualization has attracted attention as a key technology to provide a flexible network infrastructure [6–8]. Software-defined networking (SDN) is an emerging paradigm that paves the way to realize the network virtualization [9]. One approach for the network operator to accommodating traffic demand on an optical network is to construct a virtual network (VN) and reconfigure a VN following traffic changes. Fig. 1.1 shows the outline of the VN reconfiguration approach. In this approach, the network operator observes the traffic information from the network, and calculates a suitable VN based on the collected traffic information, and then configures the VN by slicing physical resources, such as wavelengths in traditional Wavelength Division Multiplexing (WDM)-based networks or frequency slots in elastic optical path networks. A VN consists of a set of lightpaths (i.e., virtual links) and client nodes (e.g., IP routers), and provides the connectivity for network equipment and the bandwidth to accommodate traffic demand. Traffic demand is transferred over the VN in a multi-hop manner. When fluctuations in traffic demand cause temporal traffic congestion, the network operator resolves the traffic congestion by reconfiguring the VN. In this way, network operators accommodate fluctuating traffic demand on optical networks and provide good service quality using limited network resources.

Many researches have been devoted to investigating methods to reconfigure a VN for optical networks. Basically, the existing methods configure a VN that achieves some objectives, i.e., design an optimal virtual topology and allocate resources, by using mixed integer linear programming (MILP) or heuristic algorithms, given the end-to-end traffic demand matrix [10–25]. Refs. [10–22] propose VN reconfiguration methods that use directly observed information of the traffic demand matrix as an input. However, it requires a large amount of computational resources to directly observe information about the traffic demand matrix since traffic inspection is necessary to measure the traffic volume for each

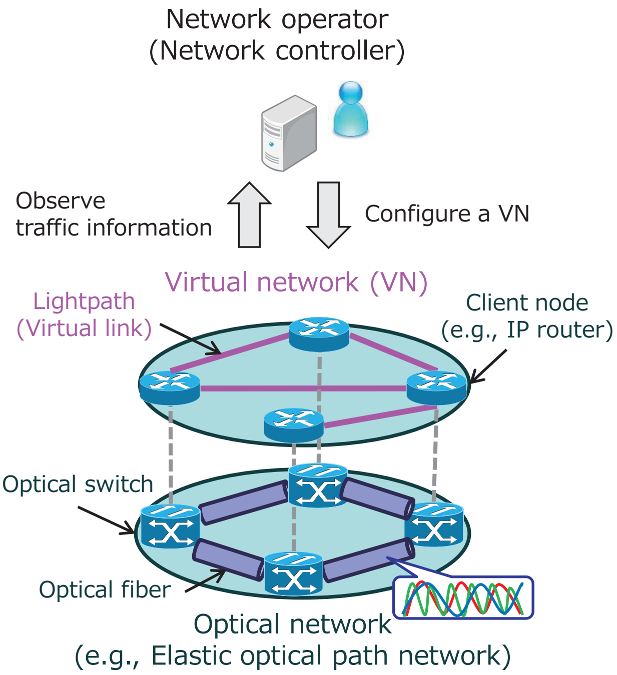


Figure 1.1: Virtual Network Reconfiguration

end-to-end node pair [26]. Therefore, in general, we obtain the information of the traffic demand matrix by long-term packet sampling. As the network scale becomes larger, the amount of information about the traffic demand matrix to be collected grows significantly, which needs more time and effort to reconfigure a VN. Thus, existing VN reconfiguration methods with the knowledge of the directly observed traffic demand matrix have difficulty in following traffic fluctuations.

Refs. [23–25] propose methods to configure a VN using an estimated traffic demand matrix. There are also many studies for estimating the traffic demand matrix, such as Refs. [26–31]. However, since the approaches for estimating the traffic demand matrix fit the collected traffic information to a specific traffic model or past traffic data, they cannot deal with irregular traffic fluctuations. Although a variety of engineering techniques can reduce estimation errors, they do not guarantee the accuracy of the estimation. That is, estimation errors in the traffic demand matrix is inevitable. When a VN configured by using an estimated traffic demand matrix including estimation errors cannot accommodate traffic demand, we do not have a way for configuring the optimal VN because we have incorrect knowledge of the traffic demand matrix. Therefore, it is difficult for the optimization approaches that use the information of the traffic demand matrix to adapt to changing traffic demand.

To adapt to fluctuations in traffic demand, our research group has previously proposed a VN reconfiguration approach without using the traffic demand matrix [32, 33]. This approach observes only the service quality on a VN (i.e., link-level load information), and reconfigures a VN based on attractor selection [34], which models the behavior where living organisms adapt to unknown changes in their surrounding environment and recover their condition. The service quality on a VN can be retrieved in a much shorter time, typically 5 minutes or less, than the traffic demand matrix. Fig. 1.2 shows the outline of the attractor selection-based VN reconfiguration approach. In this VN reconfiguration approach, attractors, which are a subset of the equilibrium points in the solution space, correspond to VN candidates. The basic mechanism of this approach can be described as follows:

$$\frac{d\mathbf{x}}{dt} = \alpha \cdot f(\mathbf{x}) + \eta, \tag{1.1}$$

where $\mathbf{x}$ is the shape of the virtual network topology, $f(\mathbf{x})$ represents the deterministic behavior, and η represents the stochastic behavior. Here, the deterministic behavior updates $\mathbf{x}$ so that the potential function defined using $f(\mathbf{x})$ decreases, i.e., reconfigures a VN to have a network topology close to one of the VN candidates (attractors), and the stochastic behavior makes random changes to the current VN. These behaviors are controlled by activity α, which is simple feedback that reflects the service quality on a VN. In Refs. [32,33], the simulation results targeting for traditional WDM-based networks have shown that this approach can respond to traffic changes. However, there are several problems to be solved in this VN reconfiguration approach.

First, there is no guideline on how to design attractors (i.e., VN candidates) in the attractor selection-based VN reconfiguration approach. Since the deterministic behavior of

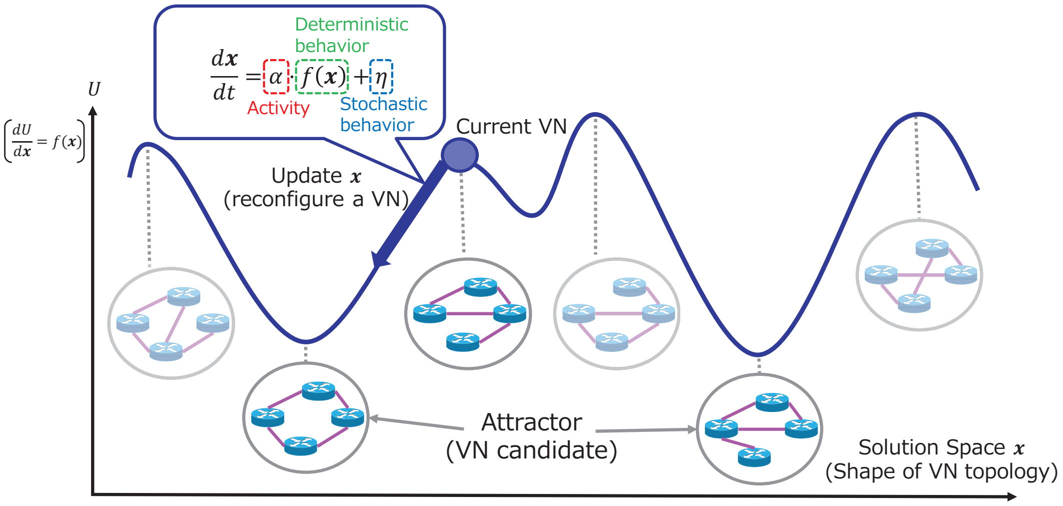

$\frac{dx}{dt} = \alpha \cdot f(x) + \eta$
Deterministic behavior
Activity
Stochastic behavior
U
$\frac{dU}{dx} = f(x)$
Current VN
Update x (reconfigure a VN)
Attractor (VN candidate)
Solution Space x (Shape of VN topology)

Figure 1.2: Virtual Network Reconfiguration Based on Attractor Selection

the attractor selection-based approach reconfigures a VN to have a network topology close to one of the VN candidates (attractors), it is crucial to design the attractors properly so that the VN reconfiguration can adapt to various traffic fluctuations. When the VN candidates are not designed properly, it takes a long time for the VN reconfiguration to find a solution (i.e., a VN that can accommodate traffic demand). For example, assuming that VN candidates are tuned for only a certain pattern of traffic demand, the attractor selection-based approach may not quickly find a solution when faced with another pattern of traffic demand since the search of a solution is guided by the attractors. An extreme approach is to prepare all VN candidates as attractors. However, the number of VN candidates that can be kept as attractors is limited to 10% to 15% of the number of possible lightpaths according to the properties of the Hopfield Network [35]. Since the Hopfield Network is one of well-known model to store and read bit-patterns using weighted matrix, we use the properties of the Hopfield Network to estimate the limitation.

Next, the previous approach in Refs. [32, 33] cannot applied to elastic optical path networks. Recently, there have been research efforts to breaking down the spectrum width of optical fibers into finer-grained frequency slots [3, 4]. Especially, elastic optical path networks have been shown to be a promising candidate for future resource-efficient optical networks. In elastic optical path networks, spectrum resources are divided into narrower frequency slots such that sufficient spectrum resources can be allocated to provide the fine-grained bandwidth. Elastic optical path networks thus achieve higher utilization efficiency of spectrum resources than traditional WDM-based networks. However, to properly reconfigure a VN and thereby provide sufficient bandwidth, it is essential to tackle the problem of allocating spectrum resources to lightpaths under the spectrum contiguity and continuity constraints. In elastic optical path networks, the frequency slots allocated to a certain lightpath must be adjacent to each other to satisfy the spectrum contiguity constraint. Furthermore, the same frequency slots must be allocated on all optical fibers in the end-to-end lightpath to satisfy the spectrum continuity constraint. The spectrum continuity constraint in elastic optical path networks corresponds to the wavelength continuity constraint in traditional WDM-based networks. For example, when t units of spectrum resources are required to establish a lightpath with sufficient bandwidth, t contiguous frequency slots must be allocated to the lightpath due to the spectrum contiguity constraint, and the same t contiguous frequency slots must be allocated on each optical fiber in the lightpath due to the spectrum continuity constraint. Thus, these resource allocation constraints in elastic optical path networks are more difficult to satisfy than that of traditional WDM-based networks.

By solving the above problems, we expect to establish an attractor selection-based VN reconfiguration method for optical networks that quickly adapts to various traffic fluctuations. However, simply applying the attractor selection-based approach may over-reconfigure a VN in nature since the attractor selection-based VN reconfiguration approach gradually reconfigures a VN in the process of searching for a solution. The attractor selection-based approach gradually reconfigures a VN using the stochastic behavior, which makes random changes to the current VN. This results in repeatedly reconfiguring a VN, which may lead to over-reconfiguration. The over-reconfiguration disrupts network services accommodated on the VN.

In this thesis, to solve the above problems, we investigate an attractor-based VN reconfiguration framework for optical networks that quickly adapts to various traffic fluctuations with fewer VN reconfigurations.

1.2 Outline of Thesis

Scalable Design of Attractors for Virtual Network Reconfiguration Based on Attractor Selection [36–40]

We first propose a method that designs attractors (i.e., VN candidates) for VN reconfiguration based on attractor selection. In this method, we design attractors with a wide variety characteristics so that the attractor selection-based VN reconfiguration can adapt to various traffic fluctuations. Our basic approach is to classify various VN candidates into groups based on their characteristics and select an attractor from each group. We use edge betweenness centrality, which is the number of shortest paths that go through the link (lightpath), as a characteristic of the VN candidates. We then classify VN candidates that have different bottleneck links from each other into different groups. The bottleneck link is the link that has the largest value of edge betweenness centrality among the links that form a VN candidate. However, our exhaustive algorithm based on this approach has a problem of requiring large amounts of computational time for large-scale networks that have more than 10 nodes. To solve this problem, we also propose a method that hierarchically contracts a network topology so that our algorithm can be applied to large-scale networks. Evaluation results show that our method can design VN candidates that achieve better service quality on a VN than the randomly generated VN candidates, even when targeting for a 1000-node network. As a result, the VN reconfiguration using attractors obtained by our design method finds a solution in a shorter time against various traffic fluctuations.

Virtual Network Reconfiguration Based on Attractor Selection for Elastic Optical Path Networks [41, 42]

Second, we propose an attractor selection-based virtual network reconfiguration method for elastic optical path networks. We newly define the potential bandwidth as a metric that reflects the bandwidth that can be additionally offered under the spectrum contiguity and continuity constraints. Then, our method reconfigures a VN based on attractor selection so that both the service quality on the VN and the potential bandwidth get improved. In addition, our method adjusts the bandwidth according to the link utilization of lightpaths that form the VN to provide the required bandwidth. Evaluation results show that the proposed method can set aside about 50 % of resources for future use, while improving the service quality on a VN to the same extent as the existing heuristic method, considering the spectrum contiguity and continuity constraints.

Virtual Network Reconfiguration Based on Bayesian Attractor Model [43–45]

Third, to reduce the number of VN reconfigurations, we introduce a cognitive mechanism that perceives current traffic situation and adapts to the situation. Specifically, we propose another VN reconfiguration method based on the Bayesian Attractor Model (BAM) [46], which models the human behavior of making appropriate decisions by recognizing the surrounding situation. The key idea of this method is to memorize a set of VN candidates, each of which works well for a pre-specified traffic situation, and then retrieve a suitable VN candidate for the current traffic situation from this set. We use certain patterns of incoming and outgoing traffic at edge routers to characterize the traffic situation, as this information can be obtained more easily than the traffic demand matrix. By identifying the current traffic situation to the closest one among the stored traffic situations using the BAM, this method retrieves and configures the most promising VN. However, for the case where the retrieved VN cannot accommodate traffic demand, we apply the attractor selection-based VN reconfiguration method. That is, we establish a VN reconfiguration framework that deals with known traffic situations by the BAM-based method and deals with unknown traffic situations by the attractor selection-based method. Evaluation results show that the BAM-based method can identify the current traffic situation by observing the amounts of incoming and outgoing traffic at edge routers. As a result, our VN reconfiguration framework can reach a VN suitable for the current traffic situation with fewer VN reconfigurations.

Finally, we extend the above VN reconfiguration framework to deal with the case where the identification of the current traffic situation fails. The BAM-based method can retrieve the most promising VN when the identification of the current traffic situation succeeds. However, when the identification fails, the BAM-based method cannot configure a promising VN. Although the BAM-based method can reduce failure of the identification by increasing the number of pre-specified traffic situations, it takes a longer time to identify more traffic situations. We therefore propose a method that configures a promising VN when the identification fails, and incorporate this method into the above VN reconfiguration framework. Our method utilizes a set of pre-specified traffic situations, and the current traffic situation is fitted by linear regression when the identification fails. Then, our method configures a VN using the obtained regression coefficients. Evaluation results show that the linear regression-based method can configure a suitable VN in most cases when failing to identify the current traffic situation, which leads to reduction of the number of VN reconfigurations. We also investigate how to select and update the set of pre-specified traffic situations, and found that it is effective to select a set of pre-specified traffic situations to have linear independence.

Chapter 2

Scalable Design of Attractors for Virtual Network Reconfiguration Based on Attractor Selection

2.1 Introduction

Wavelength-routed networks based on wavelength division multiplexing (WDM) technology are flexible infrastructures that can reconfigure the connectivity of network equipment and/or bandwidth in a dynamical manner. Many researches have investigated methods for accommodating IP traffic over WDM networks [47–49]. IP over WDM networks consist of two layers, the WDM network and IP network (Fig. 2.1). In the WDM network, optical cross-connects (OXCs) are interconnected by optical fibers. A set of optical channels, called lightpaths, are established between IP routers via OXCs. Lightpaths and IP routers form a virtual network (VN) that accommodates the IP traffic on the WDM network. IP packets in the form of electric signals are converted into optical signals and OXCs switch optical signals in the WDM network. When fluctuations in traffic demand cause traffic congestion temporarily, the traffic congestion is resolved by reconfiguring the VN according to the traffic changes such that the VN can accommodate the changing traffic demand.

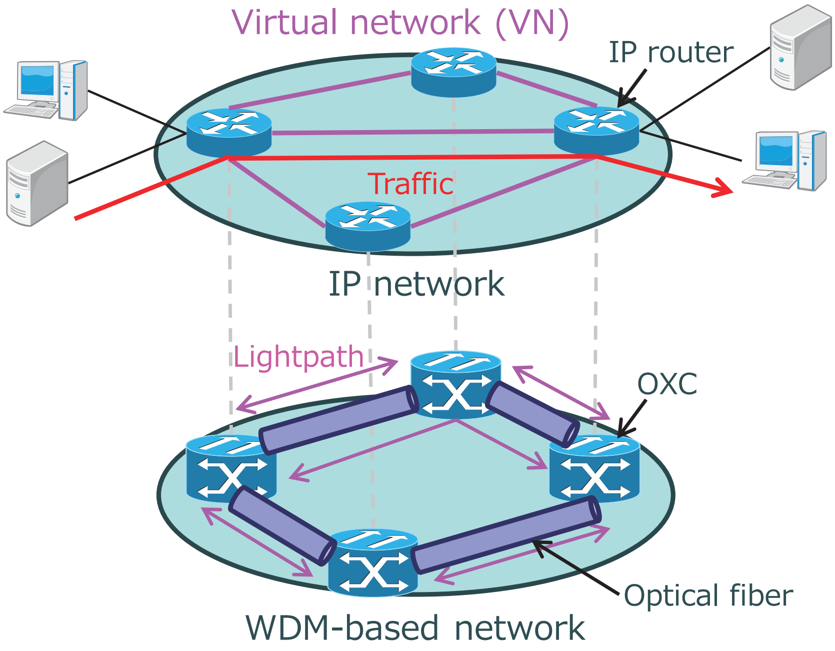


Figure 2.1: IP over WDM network

Many studies have been devoted to developing methods for accommodating IP traffic on a VN according to traffic changes [13–15]. Refs. [13–15] propose methods for configuring a VN by solving a mixed integer linear program (MILP), which aims to minimize the maximum link utilization of the VN, the packet delay, or the number of resources that form the VN (e.g., wavelengths and router ports). These methods use traffic information in the form of traffic demand matrices. The existing heuristic approaches such as I-MLTDA and MLDA [16] also use the information of traffic demand matrices. However, it generally takes a long time to retrieve the information of the traffic demand matrix. The methods proposed in Refs. [13–16] therefore reconfigure the VN based on long-term measurements of traffic demand.

However, when traffic demand fluctuates rapidly, it is difficult for methods that design the VN based on traffic demand matrices to reconfigure the VN following traffic changes. Changes in the environment surrounding the Internet in recent years, such as advances in personal Internet-enabled devices and the emergence of new Internet services, cause large fluctuations in traffic demand. For example, Refs. [50, 51] note that flash crowds of traffic have recently become more likely to occur. A flash crowd is a phenomenon where traffic to a certain web server rapidly increases within a short period of time. A traffic engineering method that does not retrieve traffic demand matrix information has been proposed in Ref. [23]. This method reconfigures the VN by estimating the traffic demand matrices. However, estimation errors in traffic demand matrices are unavoidable in general when there are large fluctuations in traffic demand. As a result, the method based on traffic estimation does not always reconfigure the VN such that the VN can accommodate changing traffic demand. It is therefore important to devise a method for adaptively reconfiguring the VN in response to traffic changes that occur over a short period of time.

Our research group has proposed a VN reconfiguration method that is adaptive to traffic changes and accommodates IP traffic effectively [32, 33]. This method is based on a dynamical system, called the attractor selection, which models the behavior by which living organisms adapt to unknown changes in their surrounding environment and recover their condition. In our VN reconfiguration method, attractors, which are a subset of the equilibrium points in the solution space, correspond to VN candidates. The basic mechanism of this VN reconfiguration comprises deterministic and stochastic behaviors. Here, the deterministic behavior represents a VN reconfiguration directed to attractors, and the stochastic behavior represents a randomized factor to change VN configurations. These behaviors are controlled by feedback indicating a condition of the IP network. Here, the condition can be a service quality of IP network such as, for example, maximum utilization of virtual links. That is, our VN reconfiguration method does not collect the traffic demand matrices, but instead collects the condition of the network and reconfigures the VN based on the feedback. Although it is necessary to collect information about the condition of the IP network, such as load information on all links (lightpaths) in the IP network, this can be retrieved in a much shorter time, typically 5 minutes or less, than the traffic demand matrices used by existing VN reconfiguration methods. When there are large fluctuations in traffic demand, our VN reconfiguration method searches for a solution: a VN that can accommodate the traffic demand. The search for a solution is not made purely randomly by the stochastic behavior; it is also guided to attractors by the deterministic behavior. We have shown in Refs. [32, 33] that this method can reconfigure a VN adaptively in response to fluctuations in the network environment such as traffic changes and node failure.

In our VN reconfiguration method based on attractor selection, it is crucial to design attractors properly since attractors define the attractive states of the VN reconfiguration. In Refs. [32, 33], we used randomly generated VN candidates. However, if the VN candidates are not designed properly, it takes a long time for the VN reconfiguration method to find a solution. For example, assuming that VN candidates are tuned for only certain patterns of traffic demands, our VN reconfiguration method may not quickly find a solution when faced with unknown traffic changes since the search for a solution is guided by the attractors. Thus, a remaining challenge is how to design attractors that can handle fluctuations in the network environment. An extreme approach is to prepare all VN candidates as attractors. However, the number of VN candidates that can be kept as attractors is limited to 10% to 15% of the number of possible lightpaths according to the properties of the Hopfield Network [35]. The Hopfield Network is one of well-known model to store and read bit-patterns using weighted matrix. So, we use the properties of the Hopfield Network to estimate the limitation. We therefore propose a new method for designing VN candidates. Our approach classifies various VN candidates into groups based on their characteristics and selects an attractor from each group. However, an exhaustive algorithm based on this approach requires a large amount of computational time for large-scale networks, which consist of more than about 10 nodes. We therefore also propose a method for hierarchically contracting the network topology so that the algorithm can be applied to large-scale networks. By preparing a limited number of VN candidates that can accommodate various patterns of traffic demand, various kinds of VNs can be searched by the attractor selection. Our VN reconfiguration method can thus find a solution within a short period of time. In other words, this makes our VN reconfiguration method more adaptive to traffic changes.

The rest of this chapter is organized as follows. In Section 2.2, we explain our VN reconfiguration method based on attractor selection. We then propose a method for designing

VN candidates and evaluating our algorithm in Section 2.3. We also propose a method for hierarchically designing VN candidates for large-scale networks in Section 2.4. In Section 2.5, we evaluate the method for hierarchically designing VN candidates and the VN reconfiguration method using the attractors obtained by this method. We conclude this chapter in Section 2.6.

2.2 Virtual Network Reconfiguration Based on Attractor Selection

We will start by explaining the VN reconfiguration method based on attractor selection proposed in [32, 33]. In this chapter, traffic is assumed to flow between IP routers via the shortest path of the VN. We simply refer to link utilization on the VN as link utilization.

2.2.1 Overview of VN Reconfiguration Based on Attractor Selection

Dynamic systems that are driven by the attractor selection adapt to unknown changes in their surrounding environments [34]. In the attractor selection, attractors are a subset of the equilibrium points in the solution space where the system conditions are preferable. The basic mechanism of the attractor selection comprises both deterministic behavior and stochastic behavior. The behavior of a dynamic system driven by attractor selection is described as follows:

$$\frac{d\mathbf{x}}{dt} = \alpha \cdot f(\mathbf{x}) + \eta. \tag{2.1}$$

The state of the system is represented by $\mathbf{x} = (x_1, ..., x_i, ..., x_n)$ (where n is the number of state variables). $f(\mathbf{x})$ represents the deterministic behavior and η represents the stochastic behavior. These behaviors are controlled by activity α, which is simple feedback of the system conditions. When the current system conditions are suitable for the environment and the value of α is large, the deterministic behavior drives the system to the attractor. When the current system conditions are poor, that is, when the value of α is small, the stochastic behavior dominates the control of the system. While the stochastic behavior dominates over the deterministic behavior, the state of the system fluctuates randomly due to noise η and the system searches for a solution where the system conditions are preferable. In this way, attractor selection adapts to environmental changes using both deterministic behavior and stochastic behavior based on the activity.

Our VN reconfiguration method considers the state of the system $\mathbf{x}$ as the state of all possible lightpaths that form the VN and uses the condition of the IP network as the activity. Our VN reconfiguration method then configures the VN so that the condition of the IP network improves when the condition of the IP network becomes poor due to fluctuations in traffic demand.

2.2.2 Dynamics of VN Reconfiguration

Our VN reconfiguration method decides whether or not to set up a lightpath l_i based on a state variable $x_i (\in \mathbf{X})$. The dynamics of the state variable x_i are defined by

$$\frac{dx_i}{dt} = \alpha \cdot \left(\varsigma\left(\sum_j W_{ij} x_j \right) - x_i \right) + \eta. \tag{2.2}$$

The activity α indicates the condition of the IP network. The term $\varsigma(\Sigma_j W_{ij} x_j) - x_i$ represents the deterministic behavior where $\varsigma(z) = \tanh(\frac{\mu}{2} z)$ is a sigmoidal regulation function, where μ indicates the parameter of the sigmoidal function. The first term is calculated using a regulatory matrix W_{ij}. The second term η represents the stochastic behavior and is white Gaussian noise with a mean value of zero. After x_i is updated on the basis of Eq. (2.2), we decide whether or not to set up the lightpath l_i. Specifically, we set the threshold to zero and if x_i is greater than or equal to the threshold, we set up the lightpath l_i and otherwise tear down the lightpath l_i.

Activity

Our VN reconfiguration method uses maximum link utilization on the IP network as a performance metric. Although it is necessary to collect load information on all links (lightpaths) in the IP network, this can be retrieved in a much shorter time than the traffic demand matrices used by existing VN reconfiguration methods. We convert the maximum link utilization on the IP network, u_{max}, into the activity α using the following expression Eq. (2.3). The activity is in the range of [0, γ]. The constant number θ is the threshold for the VN reconfiguration. When the maximum link utilization is more than the threshold θ, the activity rapidly approaches zero and our VN reconfiguration method searches for a solution that improves the condition of the IP network. The constant δ determines the gradient of the function.

$$\alpha = \frac{\gamma}{1 + \exp(\delta \cdot (u_{max} - \theta))} \tag{2.3}$$

Regulatory Matrix

We set the regulatory matrix so that it has a set of VN candidates as attractors. That is, we set the regulatory matrix $\mathbf{W}$ so that $d\mathbf{x}/dt$ in Eq. (2.1) is equal to zero when the VN configured by our VN reconfiguration method $\mathbf{x} = (x_1, \cdots, x_i, \cdots, x_n)$ is one of the attractors. To store attractors in the regulatory matrix, we use a method to decide the regulatory matrix using the pseudo inverse matrix, which is shown in Ref. [52]. Specifically, assuming that we set m VN candidates as attractors and one of the candidates is represented by $\mathbf{x}^{(k)} = (x_1^{(k)}, \cdots, x_i^{(k)}, \cdots, x_n^{(k)})(1 \leq k \leq m)$, the regulatory matrix that has m attractors is

$$\mathbf{W} = \mathbf{X}^{+}\mathbf{X}, \tag{2.4}$$

where $\mathbf{X}$ is a matrix that has $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, ..., \mathbf{x}^{(m)}$ in each row and $\mathbf{X}^{+}$ is the pseudo inverse matrix of $\mathbf{X}$.

2.3 Attractor Design Method

2.3.1 Attractor Design Problem

We suppose that we are designing attractors (i.e., VN candidates) for a network with n nodes. Although the size of the solution space is 2^{n^2}, the number of VN candidates that can be kept as attractors is limited to about 10% to 15% of the number of possible lightpaths, n^2 [35]. Moreover, VN candidates should be diverse enough to allow them to adapt to various fluctuations in traffic demands. Therefore, the problem of properly designing the VN candidates to use as attractors comes down to the problem of selecting $0.1n^2$ VN candidates that have a wide diversity from within the solution space.

For this problem, we focus on the characteristics of the VN candidates. Since the VN configured by our VN reconfiguration method based on attractor selection finally converges to one of the attractors, one of the attractors should accommodate the current traffic demand. In other words, the traffic demand that can be accommodated should differ among the VN candidates. It is important that the attractors have different characteristics in order to produce a diverse range of VN candidates. We therefore take the approach of classifying VN candidates into groups based on their characteristics and selecting one attractor from each of the VN candidate groups. By preparing a limited number of VN candidates with diverse characteristics, attractor selection is able to search various kinds of VNs. As a result, our VN reconfiguration method finds a solution quickly, making our VN reconfiguration method more adaptive to traffic changes.

2.3.2 Exhaustive Algorithm for Designing Attractors with Different Characteristics

We develop an algorithm that selects attractors for our VN reconfiguration method. The goal of our algorithm is to select $0.1n^2$ attractors with a diverse range of characteristics from the 2^{n^2} solution space. An outline of our algorithm is as follows.

1. Enumerate isomorphic VN candidates of the VN g.

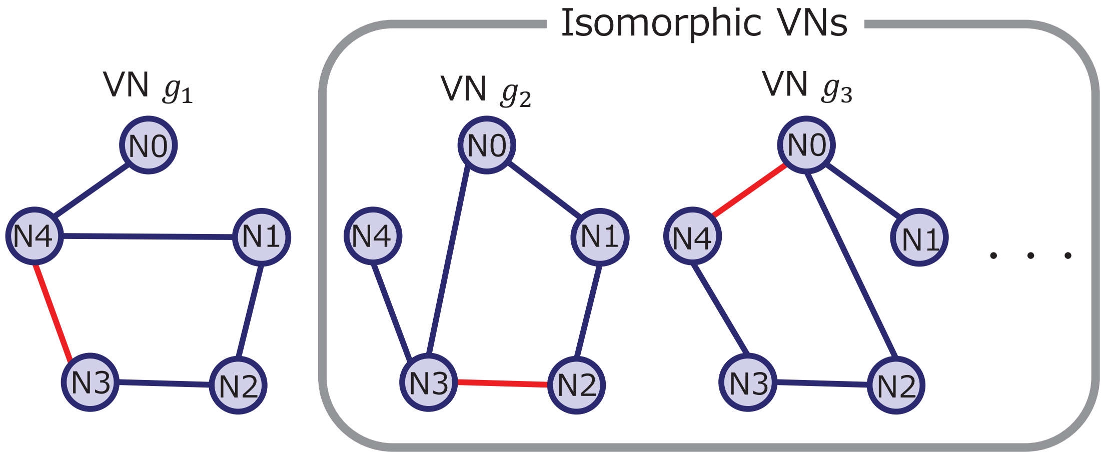


Figure 2.2: An example of isomorphic VNs

2. Classify the enumerated VN candidates based on their characteristics.

3. Select an attractor from each group of VN candidates.

In this algorithm, a VN g is given in advance. We use a heuristic method to configure the VN g based on the traffic demand matrix T. Note that although we use the traffic demand matrix T to design VN candidates, we do not use the traffic demand matrix T in our VN reconfiguration method. The detail of the algorithm is described below.

Enumeration of VN candidates

We enumerate isomorphic VNs of g. The isomorphic VNs are generated by exchanging all the nodes of the VN g. Fig. 2.2 illustrates examples of isomorphic VNs. In Fig. 2.2, the VN g_1 consists of five nodes $N0, N1, ..., N4$ and the VNs g_2 and g_3 are isomorphic VNs of g_1. The isomorphic VN g_2 is generated by shifting $N0$ of the VN g_1 to $N1$, $N1$ to $N2$, $N2$ to $N3$, $N3$ to $N4$, $N4$ to $N0$. The isomorphic VN g_3 is generated by shifting $N0$ of the VN g_1 to $N4$, $N4$ to $N3$, $N3$ to $N2$, $N2$ to $N1$, $N1$ to $N0$. However, VN candidates that do not meet restrictions on resources in a physical network, such as the number of router ports of each node, are excluded. Thus, the number of enumerated VN candidates is at most $n!$.

In Fig. 2.2, we assume that the VN g_1 is configured by a heuristic method based on the traffic demand matrix T_1 and that the traffic load is highest on the red-lined link between nodes $N3$ and $N4$. Since the VN g_1 is configured by a heuristic method based on the traffic demand matrix T_1, the VN g_1 can accommodate T_1. Let us assume that a traffic demand matrix T_2 is generated by exchanging all the rows of T_1 and exchanging all the columns of T_1. Specifically, T_2 is generated by shifting the first row of T_1 to the second row, the second row of T_1 to the third row, $\ldots$, and finally the last row of T_1 to the first low of T_2, and shifting each column of T_1 similarly. Since we shift $N0$ of the VN g_1 to $N1$, $N1$ to $N2$, $N2$ to $N3$, $N3$ to $N4$, $N4$ to $N0$ to generate the isomorphic VN g_2, the traffic load on the link between the nodes $N2$ and $N3$ becomes the highest and the VN g_2 can accommodate T_2. That is, it is expected that any of the isomorphic VNs can accommodate changing traffic demand unless all of the values in the traffic demand matrix become too large. Hereafter, we denote G as the set that includes the VN g and the enumerated VN candidates.

Classification of VN candidates

We classify the VN candidates that belong to G into groups on the basis of their characteristics. We use edge betweenness centrality, which is the number of shortest paths that go through the link, as a characteristic of the VN candidates. We then classify VN candidates that have different bottleneck links from each other into different groups, as shown in Fig. 2.3. The bottleneck link is the link that has the largest value of edge betweenness centrality among the links that form a VN candidate. When each of the VN candidates that have been selected as attractors have different bottleneck links, it is expected that any of VN candidates selected as attractors will be able to accommodate various patterns of traffic demand. Note that in our VN reconfiguration method based on attractor selection, the maximum link utilization indicates the condition of the IP network. It is likely that a link with high link utilization also has a high value of edge betweenness centrality. Therefore, we classify the VN candidates that have the same bottleneck links into the same group. The following gives a formal definition of the VN candidate groups.

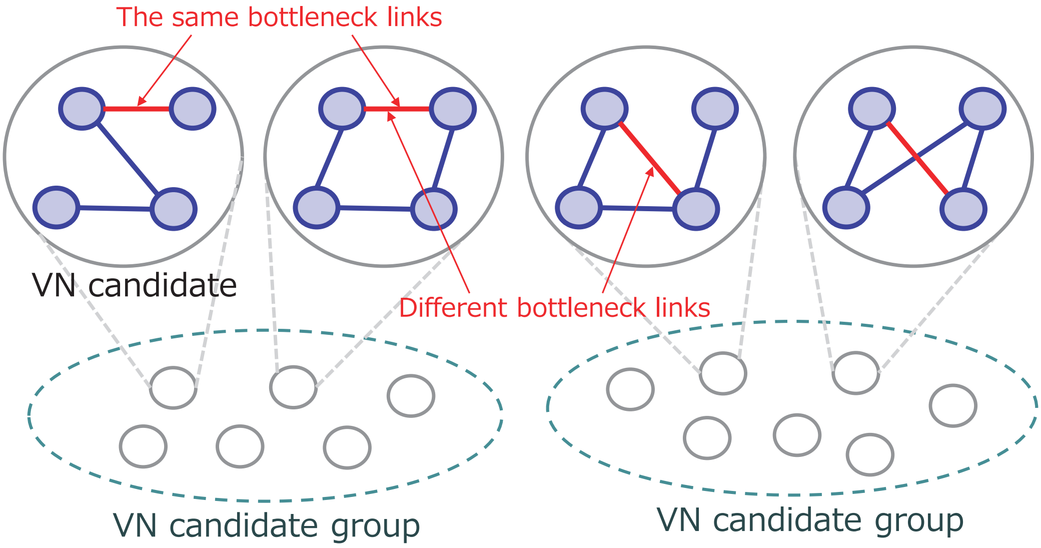


Figure 2.3: Classification of VN candidates

- $p = (s, d)$: The identifier for a node pair that has a source node s and a destination node d
- l_p: A link (lightpath) established between the node pair p
- $C(g_i, l_p)$: The value of edge betweenness centrality for the link l_p in the VN candidate g_i

Using the above notation, the VN candidate group G_p that is expected to have the bottleneck link l_p is given by

$$G_p = \{g_i | g_i \in G, C(g_i, l_p) = \max_q C(g_i, l_q)\}. \tag{2.5}$$

In this way, we divide the set of the VN candidates into groups. The number of groups is at most n^2 since the number of possible lightpaths is n^2. However, since the number of VN candidates that can be kept as attractors is $0.1n^2$, we further merge the VN candidate groups.

We merge VN candidate groups if the traffic loads of their bottleneck links are highly correlated. The condition is satisfied when the correlation of the traffic loads is high between two links connected via a node of low degree. Fig. 2.4 illustrates the condition used to merge VN candidate groups. When traffic flows from a source node s to a destination node d via a node a whose degree is low, part of the traffic that flows on link $l_{(s,a)}$ also flows on link $l_{(a,d)}$. That is, if link $l_{(s,a)}$ is a bottleneck link, it is likely that the traffic load on link $l_{(a,d)}$ is also high. We therefore treat VN candidates that belong to groups $G_{(s,a)}$ and $G_{(a,d)}$ as having similar characteristics. Based on this heuristic, we merge the VN candidates groups as follows

$$G_{(s,d)} \leftarrow G_{(s,a)} \cup G_{(a,d)} \cup G_{(s,d)}, \tag{2.6}$$

where the degree of node a is low. In Eq. (2.6), we also treat the VN candidates in the group $G_{(s,d)}$, which has the bottleneck link $l_{(s,d)}$, as having similar characteristics to the groups $G_{(s,a)}$ and $G_{(a,d)}$. The reason is that it is likely that the traffic load on the link $l_{(s,d)}$ is high when the links $l_{(s,a)}$ and $l_{(a,d)}$ are bottleneck links. We select the nodes a, s and d in ascending order of degree, since the correlation of the traffic loads on the links $l_{(s,a)}$ and $l_{(a,d)}$ is high when the degree of the node a is low. However, since each group has different VN candidates, we select nodes a, s and d based on the average value of degree among all the VN candidates in the group. We repeatedly merge the VN candidate groups until the number of VN candidate groups is about $0.1n^2$.

Selection of attractors from VN candidate groups

We finally select one attractor from each of the VN candidate groups. We select the VN candidate that has the lowest maximum value of edge betweenness centrality among the VN candidates group to use as the attractor, since the smaller the value of edge betweenness centrality, the more likely it is that the maximum link utilization is reduced.

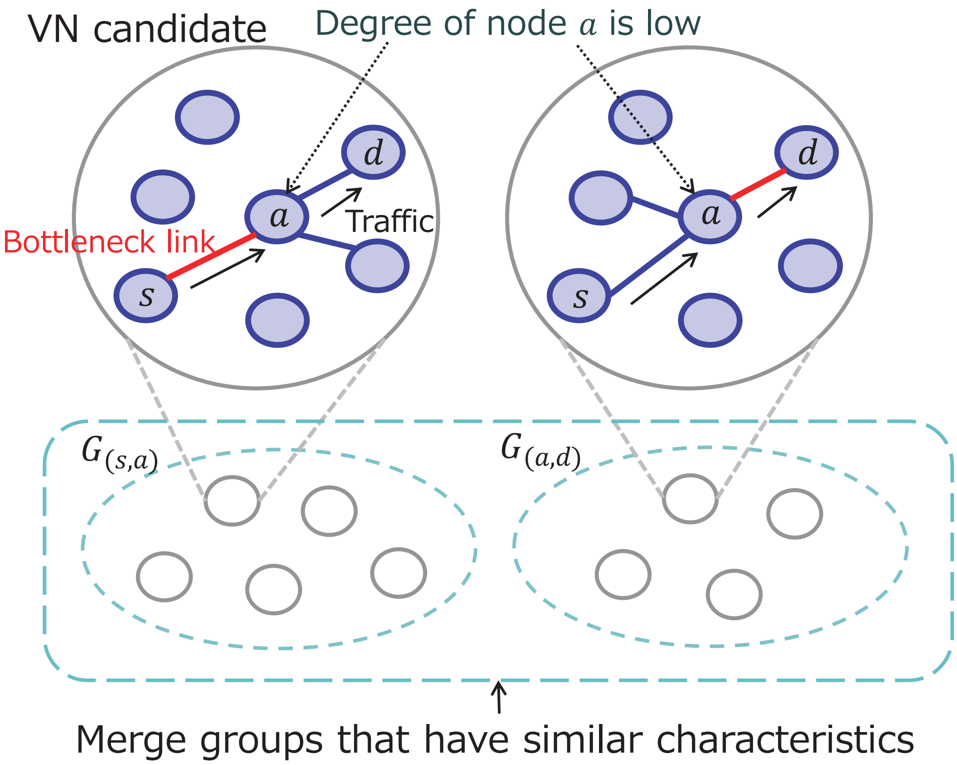


Figure 2.4: Merger of VN candidates groups

2.3.3 Effect of Design Approach: Engineered or Random

In this section, we evaluate the performance of the VN candidates obtained by the algorithm described in Section 2.3.2. We evaluate the effect of our approach to designing attractors, using a 10-node network that has a ring topology for the physical network topology. Each node has five router ports, comprising five transmitters and five receivers. We configure the VN candidates using I-MLTDA [16] as the heuristic method with a traffic demand matrix with elements that follow a log-normal distribution. We obtain 10 VN candidates by following the algorithm in Section 2.3.2. For the evaluation, we use 1,000 patterns of traffic demand between each node pair according to a log-normal distribution. We compare this to a method that constructs VN candidates by establishing lightpaths between node pairs chosen in a uniformly random manner. This is because we used randomly generated

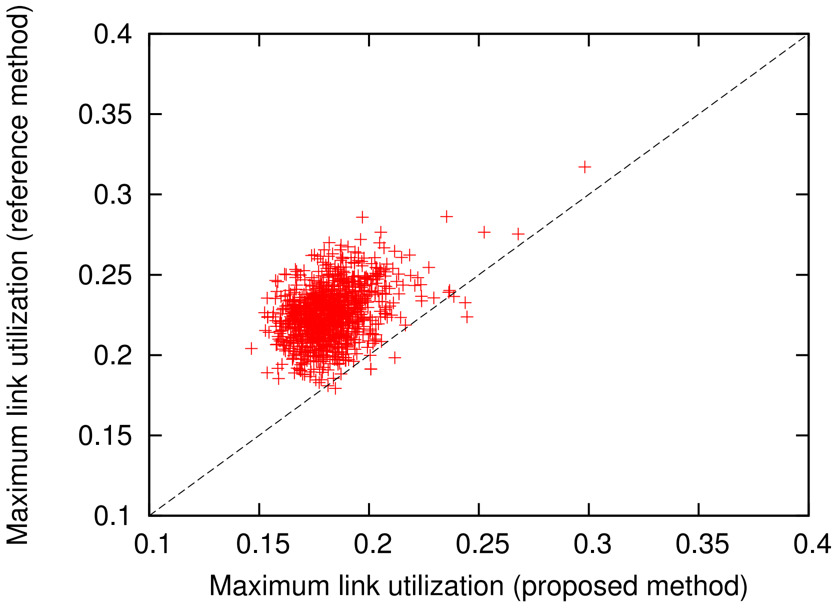


Figure 2.5: Distribution of maximum link utilization: 10 nodes, 5 ports

VN candidates in Refs. [32,33]. The number of VN candidates is the same for all methods.

Fig. 2.5 shows the distribution of maximum link utilization for each traffic pattern. Here, we see the lowest value of the maximum link utilization among the VN candidates for each traffic pattern. The horizontal axis shows the maximum link utilization of the VN candidates by our method and the vertical axis shows that of the VN candidates by the method for comparison. In Fig. 2.5, we can see that the maximum link utilization of the VN candidates by our method is lower for more traffic patterns than the other method. Specifically, the VN candidates by our method make the maximum link utilization lower than the ones by the other method for 991 traffic patterns. That is, our algorithm can design VN candidates that reduces traffic load for a wider variety of traffic demand than the other method. Thus, our approach allows us to design better attractors by classifying VN candidates into groups based on their characteristics and selecting an attractor from each group.

2.4 Scalable Design Method of Attractors

2.4.1 Scalability Problem of Exhaustive Algorithm

Although we can design better attractors based on our approach, as shown in Section 2.3.3, the algorithm described in Section 2.3.2 requires a large amount of computational time for large-scale networks. This is because the number of enumerated VN candidates increases explosively as the number of nodes n increases. Using an ordinary PC, we can design VN candidates for 10-node networks within 10 minutes calculation. However, the calculation time increases exponentially as the number of nodes increases. The calculation time is two hours for a 11-node network, and 24 hours for a 12-node network. We therefore take the approach of contracting the physical network topology and applying the algorithm in Section 2.3.2 to the contracted network topology. Specifically, we divide a physical network topology into clusters where each cluster has several nodes, as shown in Fig. 2.6. We reduce the number of nodes in the network topology by treating the clusters as nodes while properly covering the entire solution space, and design attractors with sufficiently diverse characteristics by applying the algorithm in Section 2.3.2 to the contracted network topology. Note that we can guarantee the diversity of attractors since we consider the characteristics of VN candidates diverse when the VN candidates have different bottleneck links.

2.4.2 Algorithm for Designing Attractors Hierarchically

This section gives an outline of our method for designing VN candidates hierarchically (see Fig. 2.7).

Step.1 Divide the physical network topology into clusters and set the clusters in multiple layers.

Step.2 Construct VN candidates in the clusters in the bottom layer

Step.3 Construct VN candidates in the upper layers by following the algorithm in Section 2.3.2.

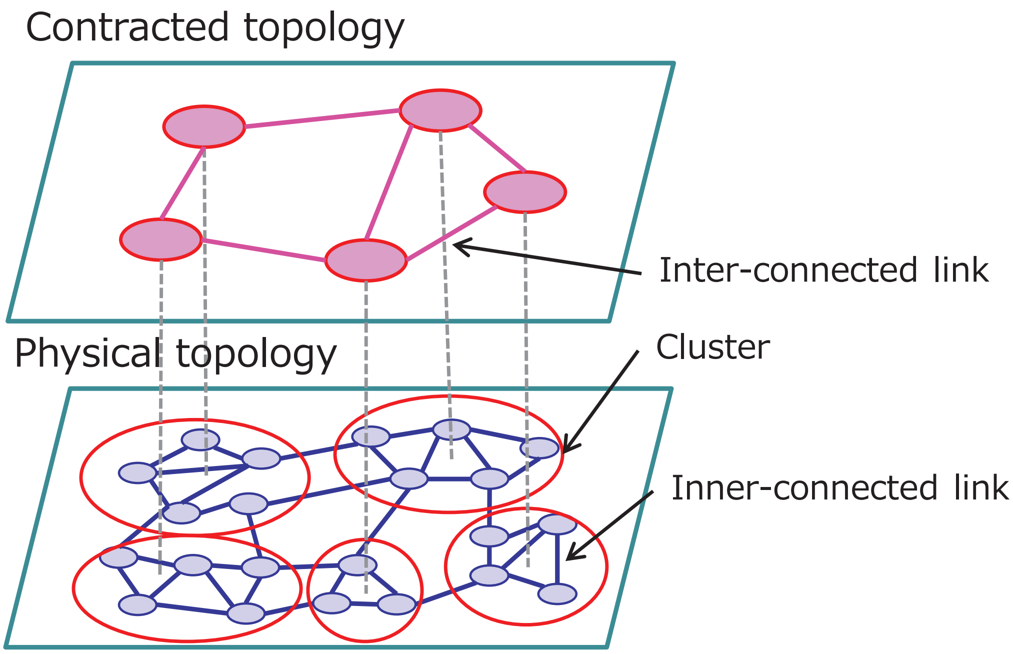


Figure 2.6: Contraction of the physical network topology

Step.4 Connect lightpaths between clusters and nodes in the clusters.

We explain the detail of the algorithm below.

Step.1 Cluster division of a physical network topology

We divide a physical network topology into c clusters. When the number of vertexes in a cluster is more than c, we divide the cluster into clusters recursively until the number of vertexes in the cluster is less than or equal to c; in other words, we decide the clusters in multiple layers. An upper layer consists of clusters that have nodes in the lower layer. For example, in a three-layer network, the top layer consists of clusters that have nodes in the middle layer. Nodes in the middle layer are then clusters that have nodes in the bottom layer (Fig. 2.8). We decide clusters based on the physical network topology. That is, we decide clusters so that nodes in a cluster are densely connected by optical fibers and nodes

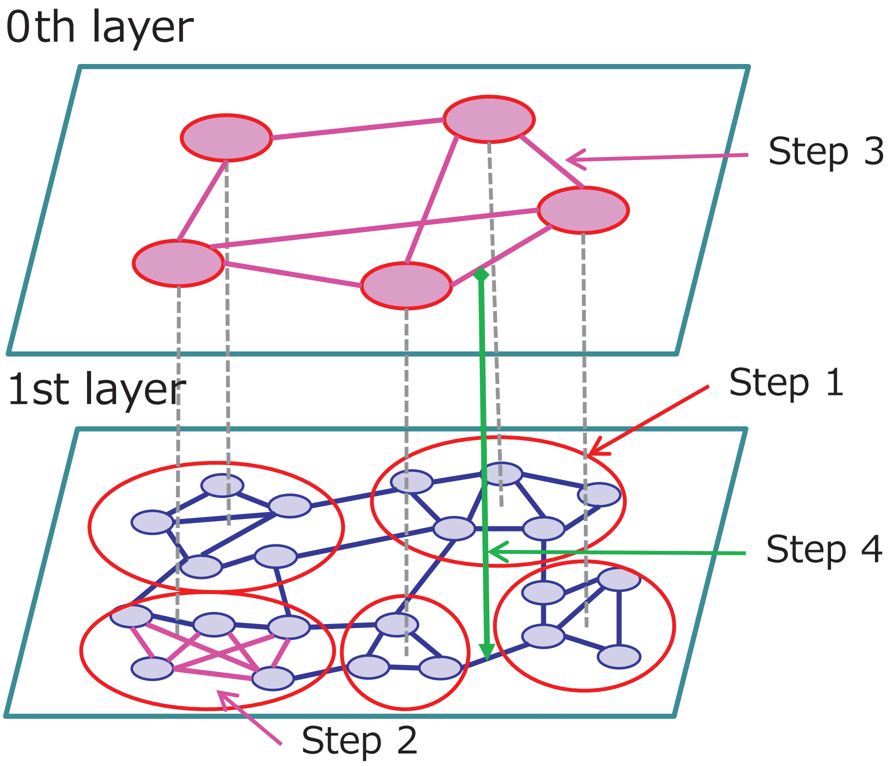


Figure 2.7: Outline of the method for designing attractors hierarchically

between clusters are sparsely connected by optical fibers.

Step.2 Construction of VN candidates inside clusters in the bottom layer

We construct VN candidates inside clusters in the bottom layer. We construct a VN candidate that has a full-mesh topology or a star topology with several hub nodes in the clusters in the bottom layer. This is because clusters can adapt to traffic changes in the cluster and can maintain connectivity when a network failure occurs.

Step.3 Construction of VN candidates in upper layers

We design VN candidates with a diverse range of characteristics in the upper layers, which has c nodes (clusters), by following the algorithm in Section 2.3.2. However, we do not apply the merge procedure of the algorithm since the number of VN candidate groups in an upper layer is at most $c(c-1)/2$, which is much smaller than $0.1n^2$, and is small enough to be kept as attractors. When we consider bidirectional lightpaths between nodes in the upper layer, the number of VN candidate groups is at most $c(c-1)/2$, which is the number of combination of nodes in the upper layer. We construct VN candidates in the upper layers as follows.

Step.3-1 Calculate VN candidates using a heuristic method and enumerate isomorphic VN candidates.

Step.3-2 Classify the enumerated VN candidates into at most $c(c-1)/2$ groups based on the edge betweenness centrality.

Step.3-3 Select an attractor from each group of VN candidates.

Step.4 Connection between clusters

We connect lightpaths between clusters to nodes in the clusters. That is, we map lightpaths between nodes in the upper layers to lightpaths between the corresponding clusters in the lower layers. We establish lightpaths between clusters from the kth layer to the $(k+1)$th layer, that is, from an upper layer to a lower layer. We decide the number of lightpaths mapped to a lower layer so that we can maximally utilize the router ports. We establish lightpaths between nodes in the clusters as follows.

- C_x^k: The xth cluster in the kth layer
- V_x^k: Nodes that belong to C_x^k
- $l_{i,j}^k$: A lightpath bidirectionally established between C_i^k and C_j^k
- k_u: The number of lightpaths connected to a node u (the degree of node u)

The probability of establishing a lightpath $l_{i,j}^k$ between $u(\in V_i^k)$ and $v(\in V_j^k)$ is given by

$$P_{u,v} = (k_u k_v)^{-1}. \tag{2.7}$$

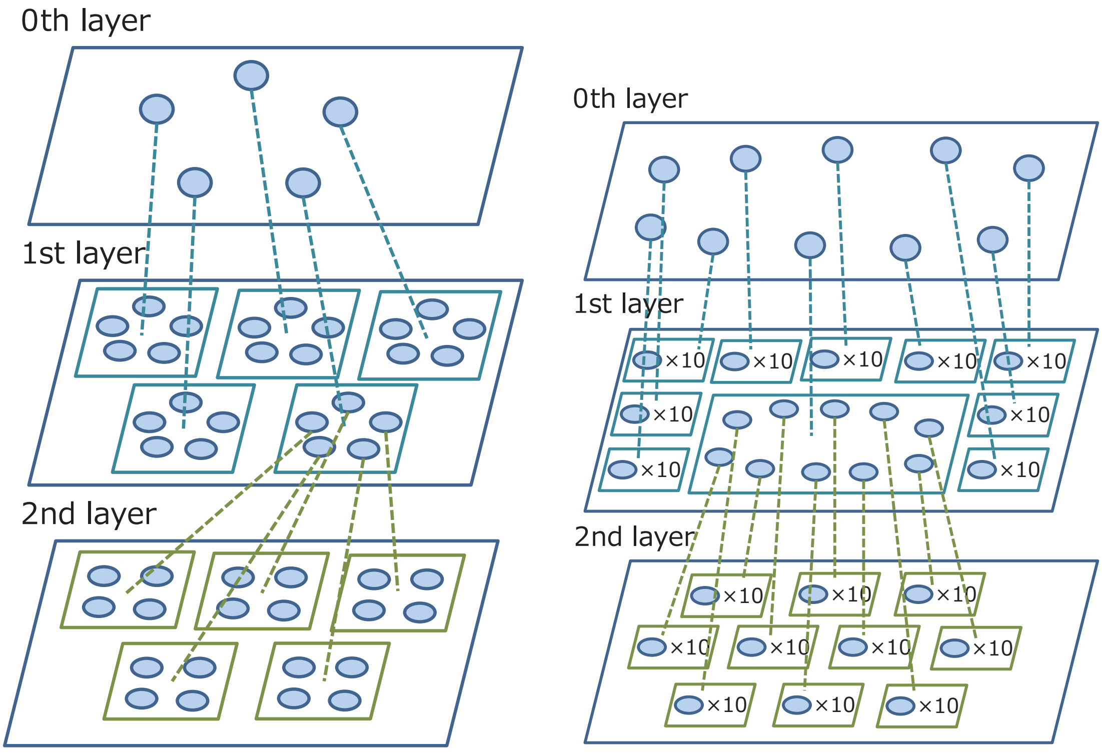
Eq. (2.7) is intended to balance the traffic loads. Since it is likely that a larger amount of traffic flows via a node as the degree of the node increases, we connect nodes that have a low degree.

2.5 Evaluation of Scalable Design Method

2.5.1 Performance of VN Candidates Obtained by Scalable Design Method

In this section, we evaluate the performance of the VN candidates by the method in Section 2.4. We first consider a 100-node network where each node has 32 router ports. We consider the three-layer network as shown in Fig. 2.8(a), where clusters in the same layer have the same number of nodes. This is because we intended to evaluate the effectiveness of our method by eliminating the influence of structural differences in physical topologies. The topology in the top layer and the topology in each cluster in the middle layer are treated as consisting of five nodes with three router ports, and we obtain seven VN candidates. The reason why the number of VN candidates is seven is that the enumerated VN candidates are classified into seven groups; only seven lightpaths become bottleneck links among the enumerated VN candidates. In the middle layer, since there are seven VN candidates in each cluster and the number of clusters is five, the maximum number of VN candidates in the middle layer becomes 7^5, counting all combinations. However, the number is too large to be kept as attractors. Therefore, we use the same VN candidate in all clusters in the middle layer. As a result, we take seven VN candidates for the middle layer. The VN candidates in each cluster in the bottom layer have a full-mesh topology. When a lightpath is established between two nodes in an upper layer, five bidirectional lightpaths are established between the corresponding clusters in the lower layer. In this way, we connect seven VN candidates in the top layer, seven VN candidates in the middle layer, and one VN candidate in the bottom layer. Finally, we obtain 49 VN candidates.

Table 2.1 shows the number of ports used in each layer in the VN candidates. In the top layer, ports are used to establish lightpaths between nodes that belong to different



(a) 100-node network

(b) 1000-node network

Figure 2.8: Clusters in the networks

clusters in the middle layer. Each of the VN candidates in the top layer has 14 lightpaths. When a lightpath is established between two nodes in the top layer, five bidirectional lightpaths are established between the corresponding clusters in the middle layer. Moreover, we map lightpaths in the middle layer to the bottom layer in the same way. Thus, we use $14 \times (5 \times 2)^2 = 1400$ ports in the top layer. In the middle layer, ports are used to establish lightpaths between nodes that belong to different clusters in the bottom layer. Each of the VN candidates in a cluster in the middle layer has 14 lightpaths. When a lightpath is established between two nodes in the middle layer, five bidirectional lightpaths are established between the corresponding clusters in the bottom layer. The number of clusters in the middle layer is five. Therefore, we use $14 \times (5 \times 2) \times 5 = 700$ ports in the

Table 2.1: Number of ports used in each layer: 100-node network

Layer	Number of ports
0th layer	1400
1st layer	700
2nd layer	300

middle layer. In the bottom layer, ports are used to establish lightpaths between nodes inside the same clusters. In bottom layer, there are 25 clusters and each cluster consists of four nodes. The VN candidates in a cluster in the bottom layer have a full-mesh topology. Therefore, we use $25 \times 12 = 300$ ports in the bottom layer. We can see that the number of ports in the middle layer is about twice that of the bottom layer, and the number of ports in the top layer is twice that of the middle layer.

For evaluation, we use 1,000 patterns of traffic demand between each node pair according to a log-normal distribution. We assume that the transmission capacity of each lightpath is equal in all the evaluation. Therefore, since the total traffic demand is excessive for the 100-node network, we use one third of the traffic demand used in Section 2.3.3. We make a comparison with the method of constructing VN candidates by establishing lightpaths between node pairs chosen in a uniformly random manner, because we used randomly generated VN candidates in Refs. [32, 33]. The number of VN candidates is the same for all methods.

Fig. 2.9 shows the distribution of maximum link utilization for each traffic pattern. This figure shows the lowest value of the maximum link utilization among the VN candidates for each traffic pattern. In Fig. 2.9, we can see that the maximum link utilization of the VN candidates by our method is less than that of the VN candidates by the other method for all traffic patterns. This shows that the method in Section 2.4 can design VN candidates that better reduce traffic loads in response to various traffic demand conditions, compared with the other method. That is, the method in Section 2.4 can design better VN candidates than the other method for larger networks where we cannot apply the algorithm in Section 2.3.2.

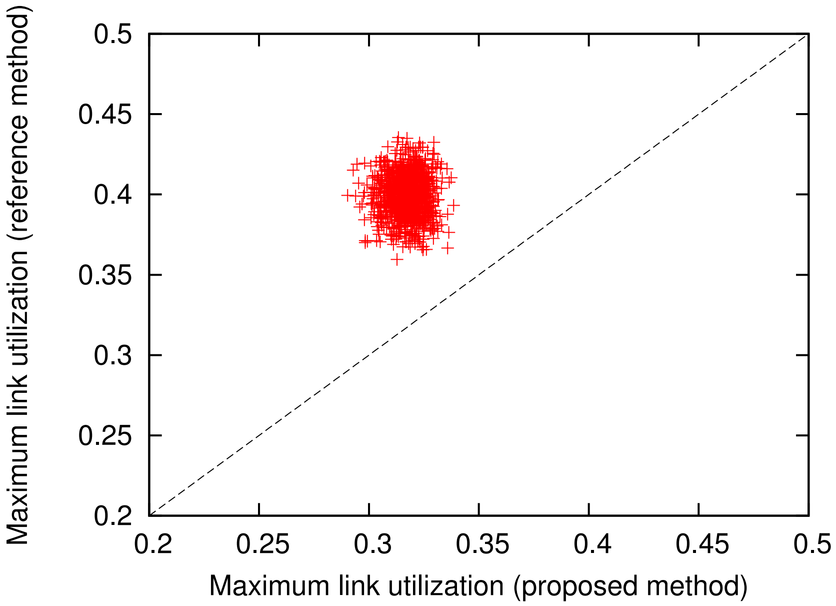


Figure 2.9: Distribution of maximum link utilization: 100 nodes, 32 ports

Note that establishing more lightpaths between clusters than inside clusters, that is, using more ports in the upper layer than in the lower layer, leads to reduction of the maximum link utilization. Assuming that the number of lightpaths between clusters is small, traffic loads on the lightpaths are high due to traffic aggregation. Traffic loads on links in upper layers are generally higher than in lower layers since a node in an upper layer consists of a larger number of nodes in the bottom layer (i.e., IP routers). For example, in the 100-node network in Fig. 2.8(a), each node at the top layer has 20 routers, while each node at the middle layer has four routers. Thus, traffic loads on links in upper layer become high in general. Actually, in obtaining Fig. 2.9, we use more lightpaths in upper layers, as shown in Table 2.1. When the number of lightpaths between clusters is high enough, the traffic demand that is transferred between clusters becomes more distributed, and the traffic load on each lightpath between clusters is reduced. By establishing more lightpaths between clusters, VN candidates obtained by our method can accommodate traffic effectively.

Further, we evaluate the performance of VN candidates when applying our method to larger-scale networks. We use a 1,000-node network where each node has 64 router ports.

We consider the three-layer network as shown in Fig. 2.8(b), where clusters in the same layer have the same number of nodes. The topology in the top layer and the topology in each cluster in the middle layer are treated as consisting of 10 nodes with five router ports, and we obtain 26 VN candidates. In the middle layer, since there are 26 VN candidates in each cluster and the number of clusters is 10, the maximum number of VN candidates in the middle layer becomes 26^{10}, counting all combinations. However, the number is too large to be kept as attractors. Therefore, we use the same VN candidate in all clusters in the middle layer. As a result, we take 26 VN candidates for the middle layer. The VN candidates in each cluster in the bottom layer have a star topology with four hub nodes. When a lightpath is established between two nodes in an upper layer, 14 bidirectional lightpaths are established between the corresponding clusters in the lower layer. In this way, we connect 26 VN candidates in the top layer, 26 VN candidates in the middle layer and one VN candidate in the bottom layer. Finally, we obtain 676 VN candidates.

Table 2.2 shows the number of ports used in each layer in the VN candidates. In the top layer, each of the VN candidates has 50 lightpaths. We map a lightpath in an upper layer to 14 bidirectional lightpaths in a lower layer recursively from an upper layer to a lower layer. Moreover, we additionally map a lightpath in the top layer to 48 lightpaths in the bottom layer in order to use the remaining router ports effectively. Thus, we use $50 \times ((14 \times 2)^2 + (48 \times 2)) = 44000$ ports in the top layer. Each of the VN candidates in a cluster in the middle layer has 50 lightpaths. When a lightpath is established between two nodes in the middle layer, 14 bidirectional lightpaths are established between the corresponding clusters in the bottom layer. The number of clusters in the middle layer is 10. Therefore, we use $50 \times (14 \times 2) \times 10 = 14000$ ports in the middle layer. In the bottom layer, there are 100 clusters and each cluster consists of 10 nodes. The VN candidates in a cluster in the second layer have a star topology with four hub nodes. Therefore, we use $100 \times 60 = 6000$ ports in the bottom layer. We can see that the number of ports in the middle layer is about twice that in the bottom layer, and the number of ports in the top layer is about three times that in the middle layer .

For evaluation, we use 100 patterns of traffic demand between each node pair according

Table 2.2: Number of ports used in each layer: 1000-node network

Layer	Number of ports
0th layer	44000
1st layer	14000
2nd layer	6000

to a log-normal distribution. We assume that the transmission capacity of each lightpath is equal in all the evaluation. Therefore, since the total traffic demand is excessive for the 1,000-node network, we use half the traffic demand used in the 100-node network. The comparison method is the same as used in the evaluation of the 100-node network.

Fig. 2.10 shows the distribution of maximum link utilization for each traffic pattern. In Fig. 2.10, we can see that the VN candidates by our method make the maximum link utilization lower than by the other method for all traffic patterns. We find that our method can design better VN candidates for 1,000-node networks, and we believe that our method can design better VN candidates for even larger-scale networks. Comparing Fig. 2.9 and Fig. 2.10, we can see that the variance of the lowest value of the maximum link utilization among the VN candidates for each traffic pattern for the 1,000-node network is much less than that of the 100-node network. This is because that the plotted value in Fig. 2.10 is most likely to be the suboptimal value for each traffic pattern with the same distribution. The number of VN candidates for the 1,000-node network is much larger than that of 100-node network. Specifically, the number of VN candidates is 676 for the 1,000-node network, and is 49 for the 100-node network. When the number of VN candidates is large, there is a high possibility that any of them shows the suboptimal value of the maximum link utilization. Moreover, we believe that each of the suboptimal value for each traffic pattern with the same distribution is close to each other. Therefore, the variance of the plotted values for the 1,000-node network is much less.

We now focus on the number of ports used in each layer, that is, the number of lightpaths established in each layer, in the networks used for the evaluation. Comparing the 100-node network with the 1,000-node network, although the number of layers is the same, the cluster

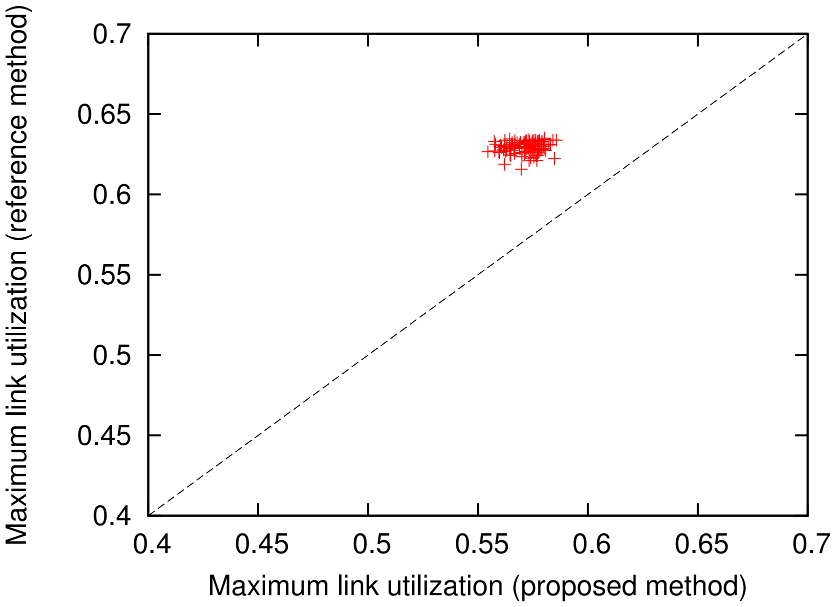


Figure 2.10: Distribution of maximum link utilization: 1000 nodes, 64 ports

size (i.e., the number of nodes in a cluster), and the number of clusters in each layer in the latter network is larger. As the number of nodes in the network increases, the cluster size and/or the number of clusters increases. The larger the cluster size, the larger the amount of traffic demand that is transferred between clusters. As a result, the traffic load on lightpaths between clusters becomes higher. However, by using more ports to establish lightpaths between clusters, the traffic demand transferred between clusters becomes more distributed, and the traffic load on each lightpath between clusters is reduced. Therefore, as the number of nodes in the network increases, we should use more ports (i.e., establish more lightpaths) in the upper layer. We actually use more ports in the upper layer than in the lower layer as the number of nodes in the network increases. In the 100-node network, the number of ports used in the upper layer is twice that used in the lower layer, as shown in Table 2.1. In the 1,000-node network, the number of ports used in the upper layer is two or three times that of the lower layer, as shown in Table 2.2.

2.5.2 Adaptability of VN Reconfiguration Based on Attractor Selection

In this section, we evaluate the adaptability of our VN reconfiguration method based on the attractor selection described in Section 2.2 using VN candidates obtained by our method as attractors. We use the term "adaptability" to represent smaller number of VN reconfigurations until convergence, i.e., a shorter time until VN reconfiguration finds a solution. When VN reconfiguration requires a shorter time to find a solution, the VN reconfiguration is more "adaptive" in response to traffic changes. We set the target maximum link utilization θ in Eq. (2.3) to 0.5, and consider our VN reconfiguration to have succeeded when the maximum link utilization is reduced to less than 0.5 within 10 successive steps of the VN reconfiguration. We assume that traffic fluctuations occur at time zero, and evaluate the number of steps in VN reconfiguration until convergence, which is required for success of the VN reconfiguration. At each step, our VN reconfiguration method collects load information on all lightpaths, calculates the activity α, and reconfigures the VN. We set μ of the sigmoidal function $\varsigma(z)$ in Eq. (2.2) to 20, and set γ to 1 and δ to 50 in Eq. (2.3).

Fig. 2.11 shows the distribution of the number of steps until convergence in the 100-node network. The traffic demand and VN candidates used as attractors are similar to those described in Section 2.5.1. The horizontal axis shows the number of steps until convergence and the vertical axis shows the complementary cumulative distribution function (CCDF) of the number of steps. This shows that VN reconfiguration using the VN candidates given by our method as attractors requires fewer steps until convergence. Since the VN candidates from our method can reduce the maximum link utilization, as shown in Section 2.5.1, our VN reconfiguration method is guided by better attractors and finds a solution, that is, a VN that can accommodate traffic, within a shorter time.

When we evaluate the VN reconfiguration method based on attractor selection in the 1,000-node network by using the VN candidates from our method as attractors, we expect to obtain similar results to the above on the basis of the results we have already obtained. The VN candidates from our method in the 100-node network can better reduce the maximum

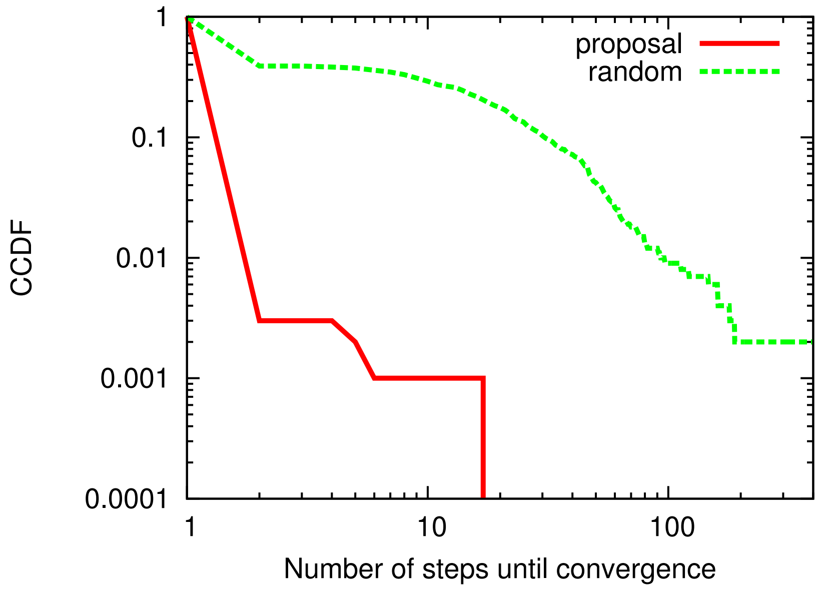


Figure 2.11: Distribution of the number of steps until convergence: 100-nodes, 32-ports

link utilization than the candidates from the other method, as shown in Fig. 2.9, and the VN reconfiguration using the VN candidates given by our method as attractors requires fewer steps until convergence, as shown in Fig. 2.11. Since the VN candidates from our method in the 1,000-node network can also better reduce the maximum link utilization, as shown in Fig. 2.10, it is expected that the VN reconfiguration using the VN candidates in the 1,000-node network given by our method as attractors requires fewer steps until convergence.

2.5.3 Effect of Physical Network Topology on Adaptability of VN Reconfiguration

In this section, we design VN candidates by dividing a large-scale network into clusters based on the physical network topology. We then evaluate the adaptability of our VN reconfiguration method using the VN candidates as attractors. We use the JPN25 model and USNET as physical network topologies. We use the method in Section 2.4 to design

Table 2.3: Number of ports used in each layer: JPN25 model

Layer	Number of ports
0th layer	140
1st layer	70

the VN candidates for the two networks. The Louvain method [53] is used to divide the physical networks into clusters such that nodes inside clusters are densely connected by optical fibers and nodes that belong to different clusters are sparsely connected.

Evaluation using the JPN25 model

The JPN25 model has 25 nodes and each node has 10 ports. We divide the physical network into five clusters, as shown in Fig. 2.12. The nodes surrounded by circles belong to the same cluster: one cluster has six nodes, three clusters have five nodes, and one cluster has four nodes. We treat the JPN25 model as a two-layer network consisting of the top layer in which the nodes are the clusters, and the bottom layer which is equivalent to the original network. The topology in the top layer is treated as comprising five nodes with three router ports and we obtain seven VN candidates by following Step 3 of our method. The VN candidates in each cluster in the bottom layer have a star topology with two hub nodes. When a lightpath is established between two nodes in the top layer, five bidirectional lightpaths are established between the corresponding clusters in the bottom layer. In this way, we connect seven VN candidates in the top layer and one VN candidate in the bottom layer. Finally, we obtain seven VN candidates.

Table 2.3 shows the number of ports used in each layer in the VN candidates. We can see that the number of ports in the top layer is twice that in the bottom layer.

Fig. 2.13 shows the distribution of the number of steps until convergence. The traffic demand and method of comparison are the same as in Section 2.5.2. The horizontal axis shows the number of steps until convergence and the vertical axis shows the CCDF of the number of steps. This shows that our VN reconfiguration using VN candidates from our method as attractors requires fewer steps until convergence. Since our VN reconfiguration

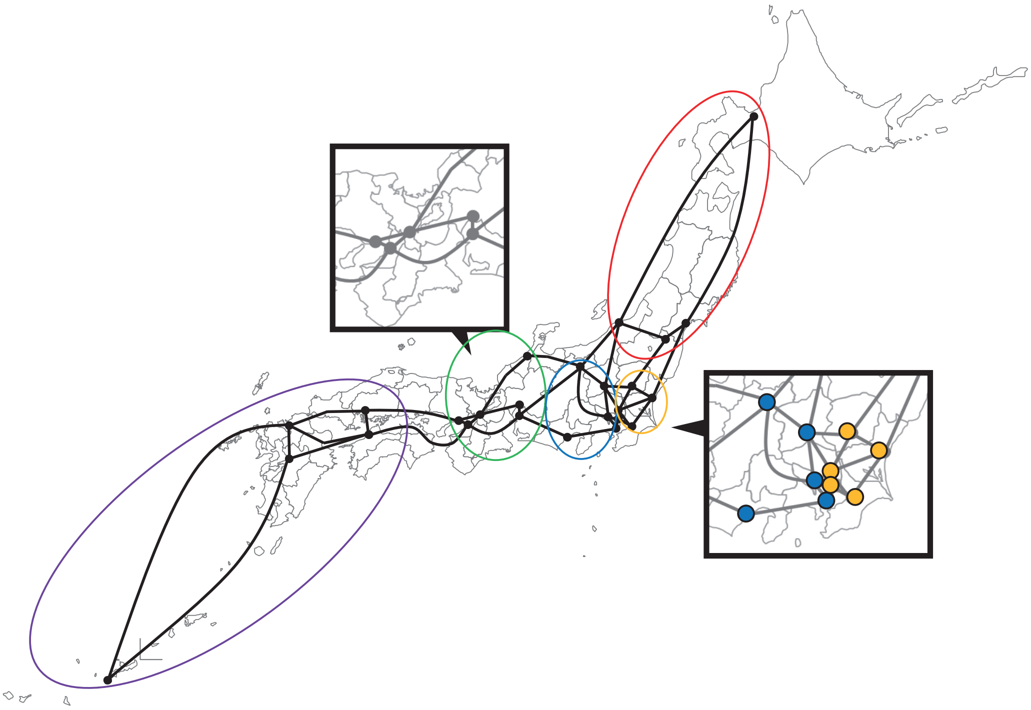


Figure 2.12: Clusters in JPN25 model

method using the VN candidates from our method as attractors finds a solution within a shorter time, this shows that our method can design better VN candidates than the other method, even when we decide clusters on the basis of the physical network topology.

Evaluation using USNET

USNET has 24 nodes and each node has 10 ports. We divide the physical network into five clusters, as shown in Fig. 2.14. The nodes surrounded by circles belong to the same cluster: one cluster has seven nodes, two clusters have five nodes, one cluster has four nodes, and one cluster has three nodes. We treat USNET as a two-layer network comprising a top layer in which the nodes are clusters and a bottom layer that is equivalent to the original network. The topology in the top layer is treated as consisting of five nodes with three

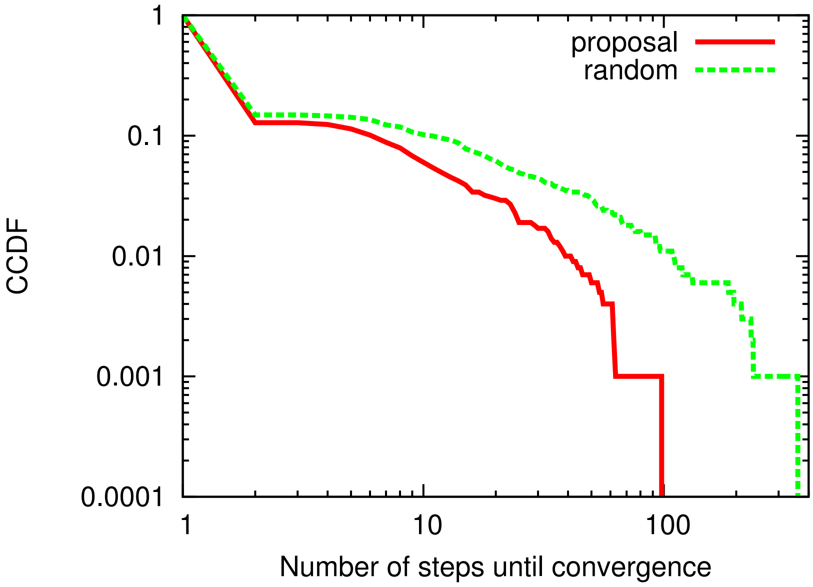


Figure 2.13: Distribution of the number of steps until convergence: JPN25 model, 10 ports

Table 2.4: Number of ports used in each layer: USNET

Layer	Number of ports
0th layer	102
1st layer	100

router ports and we obtain seven VN candidates by following Step 3 of our method. The VN candidates in each cluster in the bottom layer have a full-mesh topology. When a lightpath is established between two nodes in the top layer, four bidirectional lightpaths are established between the corresponding clusters in the bottom layer. In this way, we connect seven VN candidates in the top layer and one VN candidate in the bottom layer. Finally, we obtain seven VN candidates.

Table 2.4 shows the number of ports used in each layer in the VN candidates. We can see that the number of ports in the top layer is about the same as in the bottom layer.

Fig. 2.15 shows the distribution of the number of steps until convergence. The traffic demand and method of comparison are the same as in Section 2.5.2. The horizontal axis

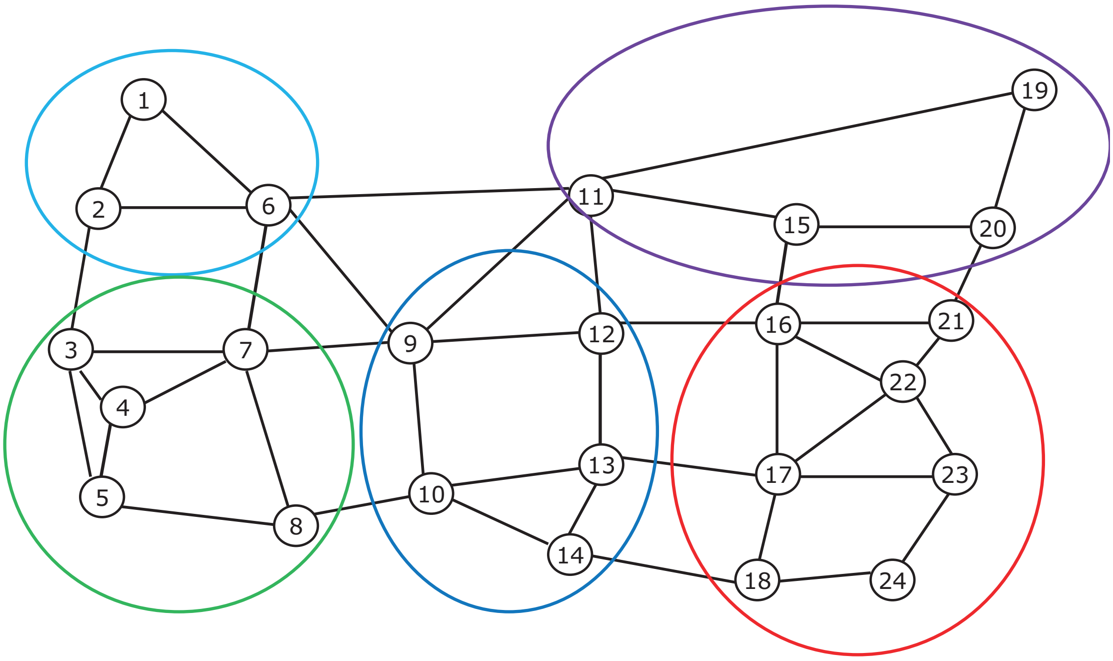


Figure 2.14: Clusters in USNET

shows the number of steps until convergence and the vertical axis shows the CCDF of the number of steps. This shows that our VN reconfiguration using VN candidates from our method as attractors requires fewer steps until convergence. That is, our VN reconfiguration method using VN candidates from our method as attractors finds a solution within a shorter time. Our method can design better VN candidates than the other method for another physical network.

As mentioned in Section 2.5.1, using more ports in an upper layer than a lower layer leads to reduction of the maximum link utilization. However, the number of ports used in the top layer is about the same as in the bottom layer in USNET, while the number of ports used in the top layer is twice that in the bottom layer in the JPN25 model. The number of clusters is the same in both networks. Therefore, difference in cluster size causes the difference in the number of ports used in each layer. The cluster size in USNET varies widely since its cluster size ranges from three to seven, while the cluster size in JPN25

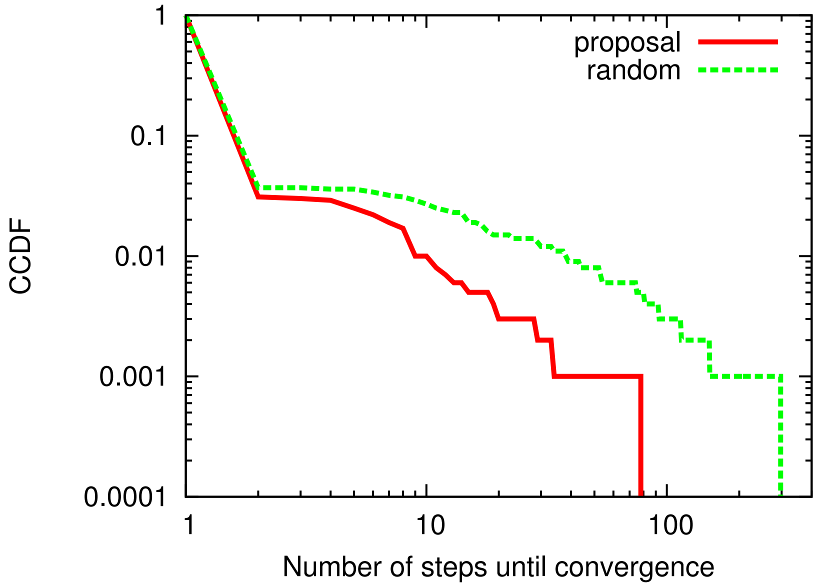


Figure 2.15: Distribution of the number of steps until convergence: USNET, 10 ports

model ranges from four to six. When the cluster size is large, a lightpath inside the cluster can be a long-distance link. That is, establishing more lightpaths inside clusters contributes to reduction of the average hop length. In general, the smaller the average hop length, the smaller the maximum link utilization of the VN. Increasing the number of lightpaths inside clusters, that is, the number of ports used in the bottom layer, results in assigning about the same number of ports in each layer. Therefore, it is necessary to adjust the number of ports used in each layer depending on the physical network topology, specifically, the cluster size and number of clusters. We can adjust the number of ports used in each layer by changing the topology in each cluster in the bottom layer, such as a full-mesh topology or a star topology with several hub nodes. When we change the topology in each cluster in the bottom layer from a full-mesh to a star, we can use the remaining ports to establish lightpaths between clusters. Actually, we use a star topology with two hub nodes in each cluster in the bottom layer for the JPN25 model to establish more lightpaths between clusters. We then use a full-mesh topology for USNET to establish more lightpaths inside clusters.

2.6 Conclusion

In this chapter, we proposed a method for designing attractors for our VN reconfiguration method based on attractor selection. Our basic approach is to prepare a limited number of attractors with a diversity of characteristics by classifying VN candidates into groups based on their characteristics and selecting an attractor from each group. In order to design attractors for large-scale networks, we also proposed a method that hierarchically contracts the network topology so that we can apply our approach to large-scale networks. We showed that the VN candidates obtained by our method can accommodate various traffic demand patterns, so that our VN reconfiguration method can find a solution, that is, a VN that can accommodate IP traffic, within a shorter time when guided by the attractors.

One future direction for this work is to investigate how to update the attractors. Since our approach selects a limited number of attractors from the solution space, it is likely that attractors that are not selected can adapt to certain changes in traffic demand. Therefore, we could update the attractors to discard certain attractors and keep new attractors, depending on the traffic demand situation. By establishing a method to update attractors, it is expected that our VN reconfiguration method based on attractor selection will become even more adaptive to traffic changes.

Chapter 3

Virtual Network Reconfiguration Based on Attractor Selection for Elastic Optical Path Networks

3.1 Introduction

Changes in the environment surrounding the Internet in recent years, such as advances in personal Internet-enabled devices and the emergence of new Internet services, has led to rapid growth in traffic demand. Thanks to the large bandwidth of optical fibers, optical networks have the potential to support this growing traffic demand. Recently, elastic optical networks based on orthogonal frequency division multiplexing (OFDM) technology have been shown to be a promising candidate for future cost-efficient optical networks [3–5].

In elastic optical networks, spectrum resources are divided into narrow frequency slots such that sufficient number of frequency slots can be allocated to provide the fine-grained bandwidth. Elastic optical networks therefore offer higher spectrum utilization efficiency than traditional wavelength-routed networks based on wavelength division multiplexing (WDM) technology. This is because WDM networks require all of wavelengths in a lightpath to be assigned even when the traffic demand between end nodes is not sufficient to fill the entire bandwidth, whereas elastic optical networks have the potential to assign spectrum resources based on traffic volumes or clients' bandwidth requirements of through fine-grained frequency slots.

Many studies have been devoted to developing methods for accommodating traffic demand over an elastic optical network [17–22, 54]. One approach to accommodating traffic demand over an elastic optical network is for the network operator to offer leased lightpaths in response to requests from service providers. In Ref. [54], routing and spectrum assignment (RSA) algorithms that offer lightpaths for each individual request are proposed. In this approach, the network operator can respond to a wide variety of requests from service providers with sufficient spectrum resources to provide the required bandwidth. Allocating sufficient frequency slots to lightpaths can lead to a reduction in power consumption. However, it is difficult to offer connections for all requests for end-to-end lightpaths since the number of transponders at each optical switch is limited. Therefore, a realistic approach for the network operator is not only to offer a certain number of leased lightpaths in response to requests, but also to configure a virtual network (VN) that accommodates other consumer traffic demand [17–22]. A VN consists of a set of lightpaths and client nodes (e.g., IP routers), with traffic demand transferred over the VN in a multi-hop manner. When fluctuations in traffic demand cause temporary traffic congestion, it is necessary to reconfigure the VN so that the traffic congestion is resolved and the VN can accommodate the changing traffic demand. Furthermore, it is essential for the network operator to be able to configure the VN using a limited set of resources (i.e., frequency slots and transponders at optical switches) by using resources effectively in order to set aside resources for leased lightpaths and accommodating increased traffic demand on the VN.

Refs. [17–22] propose methods for configuring a VN over an elastic optical network. These methods first collect traffic demand information and then configure the VN. However, it generally takes a long time to retrieve information of the traffic demand matrix. Thus, when there are large fluctuations in traffic demand, these methods have difficulty in reconfiguring the VN following traffic fluctuations. It is therefore essential to develop a method for adaptively reconfiguring a VN in response to traffic changes that occur over a short period of time. Refs. [32, 36] propose a VN reconfiguration method over traditional WDM networks that is adaptive to traffic changes and accommodates IP traffic effectively. This method is based on attractor selection [34], which is a model of the behavior by which living organisms adapt to unknown changes in their surrounding environment. This method can reconfigure a VN using a small amount of information about the service quality on the IP network, such as load information on all lightpaths, which can be retrieved in a much shorter time, typically 5 min or less, than traffic demand matrices. However, optimal utilization of elastic optical networks cannot be achieved by simply adopting this method because the method focuses on reducing traffic loads. That is, this approach assigns all of the wavelengths in order to accommodate current traffic demand, which leads to a lack of resources for accommodating future traffic demand. Therefore, applying this method to elastic optical networks results in missing the opportunity to accommodate future traffic demand.

From the above discussion, the requirements for a VN reconfiguration method for elastic optical networks are as follows:

- Reconfigure the VN to accommodate changing traffic demand
- Set aside resources to use for leased lightpaths and to accommodate increased traffic demand on the VN
- Reconfigure the VN in a shorter time period

In this chapter, we newly propose a VN reconfiguration method for elastic optical networks that can achieve these requirements. To achieve the second requirement, we newly define the potential bandwidth as a metric that reflects the bandwidth that can be offered for leased lightpaths and for increased traffic demand on the VN. Specifically, our method reconfigures the VN based on attractor selection using both information about the service quality on the VN and the potential bandwidth, and adjusts the bandwidths of the lightpaths that form the VN. Our method is based on the observation of the service quality on the VN and the potential bandwidth. Therefore, measurement of traffic demand matrices is no longer necessary in our method. Note that, since the service quality on the VN depends on the traffic demand, the information on traffic demand is indirectly utilized in our method. This is the essential difference between our method and the previous methods.

The rest of this chapter is organized as follows. In Section 3.2, we first describe our target network model and related work. We then explain the concept of the attractor selection and propose the VN reconfiguration method for elastic optical networks in Section 3.3. In Section 3.4, we evaluate the performance of our method, and we conclude this chapter in Section 3.5.

3.2 Network Model and Related Work

In this section, we describe the network model that we use in this chapter and also briefly describe existing VN reconfiguration methods for elastic optical networks.

3.2.1 Network Model

Fig. 3.1 shows the network model considered in this study. Elastic optical networks can flexibly assign spectrum resources according to changing traffic demand by introducing hardware components such as bandwidth-variable transponders (BVTs) and bandwidth-variable wavelength cross-connects (BV WXCs). Elastic optical networks employing these hardware components have been modeled as spectrum-sliced elastic optical path (SLICE) networks [5]. In SLICE networks, BV WXCs are interconnected by optical fibers. A BVT converts electric signals from a client node (e.g., an IP router) into optical signals by using sufficient spectrum resources (i.e., frequency slots). Every BV WXC on the route switches the optical signals to establish an end-to-end lightpath with the sufficient bandwidth. When the bandwidth utilization of the lightpath increases, the BVT can assign more spectrum resources to the lightpath in order to expand its bandwidth. In contrast, when the bandwidth utilization of the lightpath decreases, the BVT can release some spectrum resources to reduce the bandwidth. Adjustment of the bandwidth of lightpaths according to the situation leads to reduction in the number of active frequency slots. As a result, SLICE networks

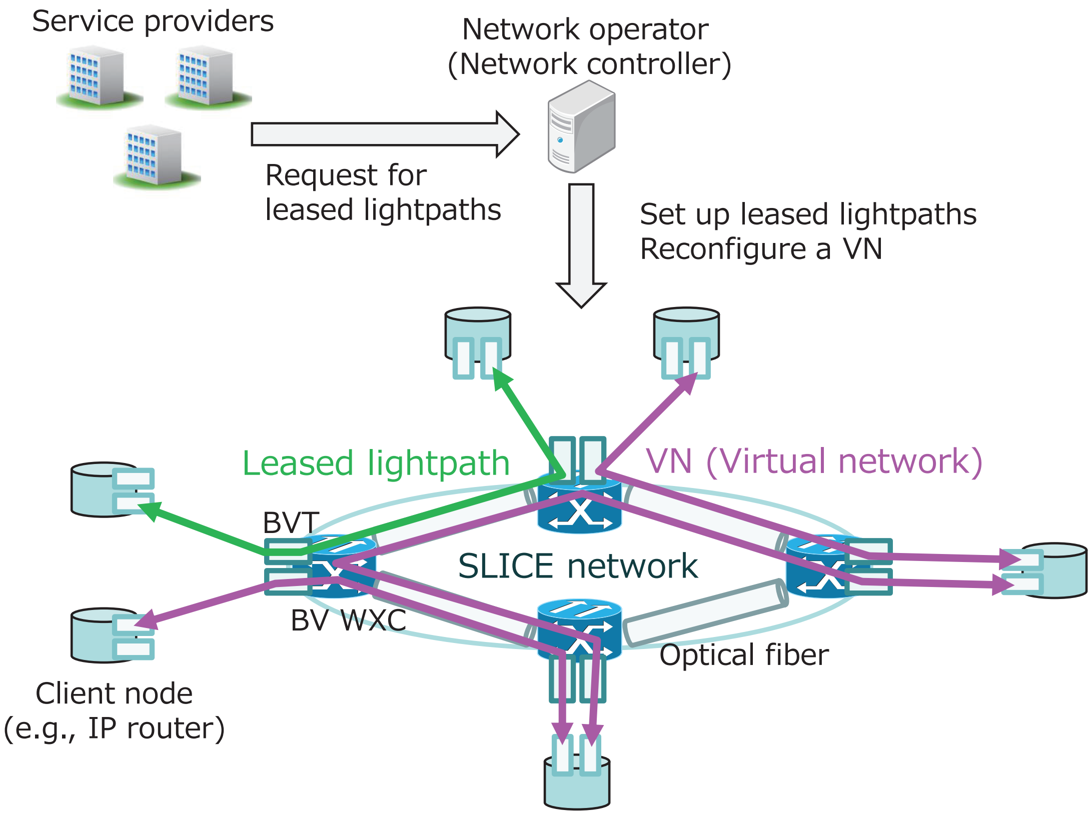


Figure 3.1: SLICE network model and operation approach

have the potential to reduce power consumption since power consumption depends on the number of active frequency slots [21,55].

The network operator accepts requests for lightpaths from service providers. When there are sufficient resources to provide a leased lightpath in response to a request, the network operator establishes a leased lightpath with the required bandwidth under the spectrum contiguity and continuity constraints. For example, when an event is held that spans more than one venue, a service provider may request dedicated lightpaths of the desired bandwidth between the venues. Furthermore, the network operator also constructs a VN to accommodate the traffic demand of other consumers. The network operator reconfigures the VN according to changes in traffic in order to maintain the service quality of network services accommodated on the VN while setting aside resources for the leased lightpaths. In this way, the network operator accepts various sources of traffic demand that are accommodated on the elastic optical network.

3.2.2 Related Work

There are many works for configuring a VN over an elastic optical network. Basically, they aim to solve routing and spectrum assignment problem for elastic optical networks using mixed integer linear programming (MILP) or using a heuristic algorithm.

Ref. [17] proposes an approach to configuring a spectrum assignment of VN over an elastic optical network. The topology of VN is assumed to be a full-mesh topology, and the algorithm adjusts the bandwidth of lightpaths based on traffic demand measurement. The method aims to minimize the packet delay and consumed capacity by MILP.

Refs. [18–20] investigate VN configuration schemes over elastic optical networks. Ref. [18] points out the advantage of BVT on total equipment costs which include the costs of slot cards, transponders, and optical switches. The routing and spectrum assignment problem is solved under static traffic demands. Ref. [19] proposes a heuristic algorithm for configuring a couple of VNs over a multi-domain elastic optical network. Using the information of traffic demands for each VN, the algorithm minimizes the total network cost, including the costs of transponders, regenerators, and spectrum resources. Ref. [20] considers modulation level assignment problem in addition to the routing and spectrum assignment problem. The heuristic algorithm in Ref. [20] jointly solves the routing, modulation level, and spectrum assignment problem. Using the traffic demand information, the algorithm tries to minimize the total network costs such as the costs of transponders and routers.

Ref. [21] presents minimal-power-consumption designs to minimize the total power consumption in elastic optical networks. The authors develop an MILP formulation and its heuristic algorithm. The traffic demand of each node pair is necessary to solve the problem. Ref. [22] proposes an MILP formulation for several schemes of protection in cases where multiple VNs run over an elastic optical network.

These previous methods [17–22] aim to configure a VN over an elastic optical network that achieves some objective by MILP or using a heuristic algorithm, based on long-term measurement of traffic demand. However, when there are large fluctuations in traffic demand, it is difficult for these methods to reconfigure the VN following traffic changes. It is therefore important to develop a method for adaptively reconfiguring a VN in response to traffic fluctuations that occur over a short period of time.

3.3 Virtual Network Reconfiguration Based on Attractor Selection for Elastic Optical Path Networks

In this section, we first briefly explain the attractor selection and then explain our VN reconfiguration method in elastic optical networks.

3.3.1 Attractor Selection

Dynamic systems driven by the attractor selection can adapt to unknown changes in their surrounding environments [34]. In the attractor selection, attractors are a subset of the equilibrium points in the solution space where the system conditions are preferable. The basic mechanism of the attractor selection consists of both deterministic behavior and stochastic behavior. The behavior of a dynamic system driven by attractor selection can be described as follows:

$$\frac{d\mathbf{x}}{dt} = \alpha \cdot f(\mathbf{x}) + \eta. \tag{3.1}$$

The state of the system is represented by $\mathbf{x} = (x_1, ..., x_i, ..., x_n)$, where n is the number of state variables. $f(\mathbf{x})$ represents the deterministic behavior and η represents the stochastic behavior. The behavior is controlled by activity α, which is simple feedback of the system conditions. When the current system conditions are suitable for the environment and the value of α is large, the deterministic behavior drives the system to the attractor. When the current system conditions are poor, that is, when the value of α is small, the stochastic behavior dominates the control of the system. While the stochastic behavior is dominant, the state of the system fluctuates randomly due to noise η and the system searches for a solution where the system conditions are preferable. In this way, attractor selection adapts to environmental changes using both deterministic behavior and stochastic behavior based on the activity.

When we investigate a method for VN reconfiguration over an elastic optical network, the method is expected to reconfigure the VN to improve the service quality on it and/or to set aside resources by properly defining the state of the system $\mathbf{x}$ and the activity α.

3.3.2 Outline of the VN Reconfiguration Method

We have developed a method for reconfiguring a VN for elastic optical networks. Our method reconfigures the virtual topology and adjusts the bandwidth of the lightpaths that form the VN in order to improve service quality on the VN while keeping some resources unused. Given a current stage of VN configuration, i.e., the virtual topology and the configuration of frequency slots, our method observes the link utilization on the VN and determines the next stage of VN configuration. The method repeatedly executes these controls based on the measured link utilization on the VN, as shown in Fig. 3.2. Here, traffic is assumed to flow between client nodes via the shortest path in the VN. We refer to link utilization on the VN as simply link utilization. The following gives an outline of our method.

(Phase 1) Reconfigure the virtual topology.

- **(Phase 1-1)** Calculate the activity based on both the service quality on the VN and information about resource utilization.
- **(Phase 1-2)** Configure the virtual topology based on attractor selection.
- **(Phase 1-3)** Allocate frequency slots to newly established lightpaths.

(Phase 2) Adjust the bandwidth of lightpaths that form the VN.

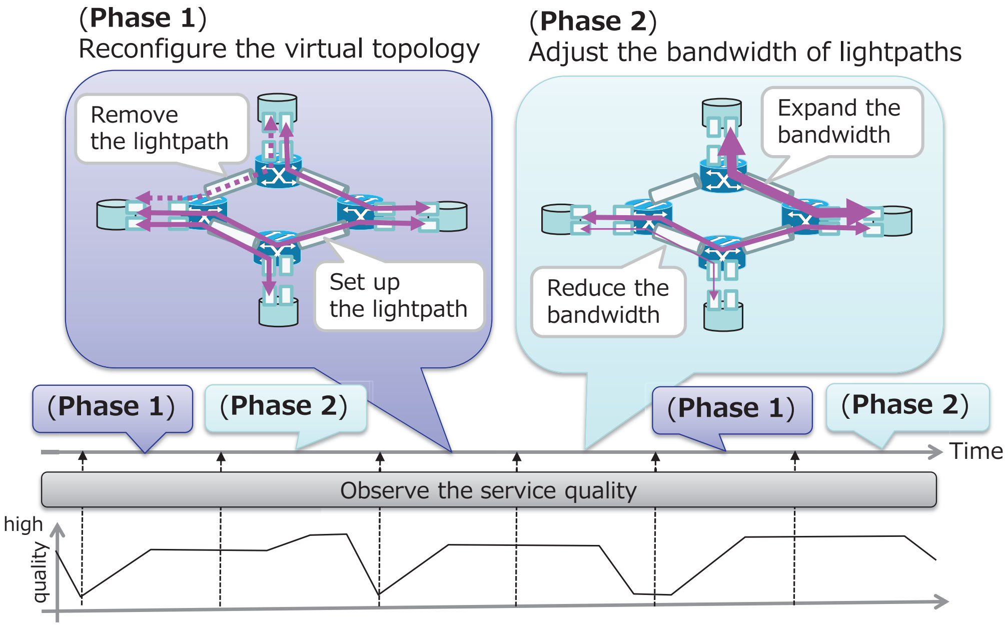


Figure 3.2: Outline of the VN reconfiguration method

The method repeatedly reconfigures a VN in response to traffic changes. As inputs, the method uses the information of link utilization on all lightpaths, the current virtual topology, and the current configuration of frequency slots. Then, the method reconfigures a VN, i.e., reconfigures the virtual topology and reallocates frequency slots, as outputs. The outputs of an execution are used as inputs for the next execution.

Detail on the VN reconfiguration in each of these phases is given in the following sections.

3.3.3 (Phase 1) Reconfiguration of the Virtual Topology

(Phase 1-1) Calculation of the activity

We calculate the activity based on both the service quality on the VN and information about resource utilization. Specifically, we use the following two performance metrics to calculate the activity.

- u_{max}: Maximum link utilization

- $B_{potential}$: Potential bandwidth

The maximum link utilization u_{max} represents the service quality on the VN. We then define the potential bandwidth $B_{potential}$ to reflect the bandwidth that can be offered for leased lightpaths and for accommodating increased traffic demand on the VN. Although our proposed method aims to set aside resources (i.e., frequency slots and BVTs) for requests for leased lightpaths and increased traffic demand, this approach to keeping as many unused resources as possible has the same goal as the approach of maximizing the traffic volume that can be accommodated in the future. Strategies for allocating resources for future traffic demand have been discussed in Ref. [56]. In Ref. [56], the authors compare two approaches: one approach is to maximize the smallest value of the bandwidth that can be added to lightpaths between every node pair, the other is to maximize the total volume of the bandwidth that can be added to end-to-end lightpaths. The latter approach was found to give lower blocking ratios for lightpath requests. That is, by introducing the latter approach, the network operator can accept more traffic demand to be accommodated in the network. We therefore also determine the potential bandwidth which reflects the total volume of bandwidth that can be additionally offered to every node pair. We define the potential bandwidth as follows:

$$B_{potential} = \sum_{s,d \in V} B^{sd}_{potential} \quad , \tag{3.2}$$

where V represents the set of nodes in the network, and $B^{sd}_{potential}$ represents the potential bandwidth for the node pair (s, d). We introduce different definitions of $B^{sd}_{potential}$ depending on whether a lightpath is established between s and d or not:

- When a lightpath is established between s and d, we define $B^{sd}_{potential}$ as the bandwidth size that can be added to the lightpath (s, d). Fig. 3.3 shows an example of $B^{sd}_{potential}$ in this case. The BVTs at nodes s and d can offer up to 40 Gbps of bandwidth, while the lightpath between s and d currently has a bandwidth of 20 Gbps. That is, the BVTs can offer an additional $40 - 20 = 20$ Gbps bandwidth. The number

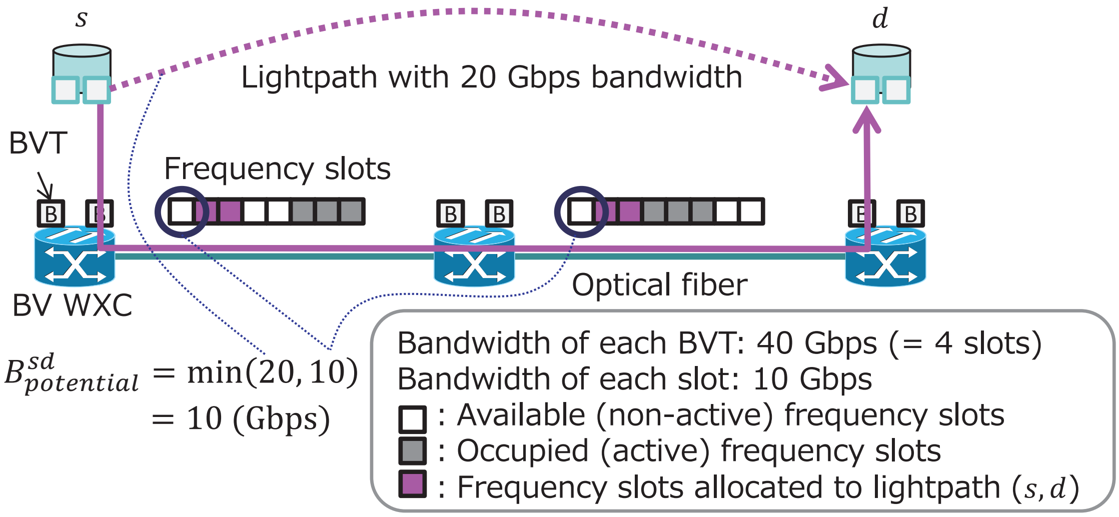

s
d
Lightpath with 20 Gbps bandwidth
BVT
Frequency slots
B
Optical fiber
BV WXC
$B^{sd}_{potential} = \min(20, 10)$
$= 10$ (Gbps)
Bandwidth of each BVT: 40 Gbps (= 4 slots)
Bandwidth of each slot: 10 Gbps
□ : Available (non-active) frequency slots
▨ : Occupied (active) frequency slots
■ : Frequency slots allocated to lightpath (s, d)

Figure 3.3: Example of the potential bandwidth for a node pair (s,d): Case where a lightpath is established between s and d

of frequency slots that can be allocated to the lightpath between s and d under the spectrum contiguity and continuity constraints is one, which gives a bandwidth of 10 Gbps. Therefore, we calculate the potential bandwidth for a node pair (s, d), $B^{sd}_{potential}$, as follows: $B^{sd}_{potential} = \min(20, 10) = 10$ Gbps.

- When a lightpath is *not* established between s and d, we define $B^{sd}_{potential}$ as the bandwidth size that can be offered if a lightpath is established between s and d. Fig. 3.4 shows an example of $B^{sd}_{potential}$ in this case. The BVTs at the nodes s and d can offer up to 40 Gbps of bandwidth. If we establish a lightpath between s and d we can allocate three frequency slots to this lightpath under the spectrum contiguity and continuity constraints, giving $10 \times 3 = 30$ Gbps. We thus calculate the potential bandwidth for the node pair (s, d), $B^{sd}_{potential}$, as follows: $B^{sd}_{potential} = \min(40, 30) = 30$ Gbps.

We get the activity α by calculating

$$\alpha = \alpha_{mlu} \cdot \alpha_{pb}, \tag{3.3}$$

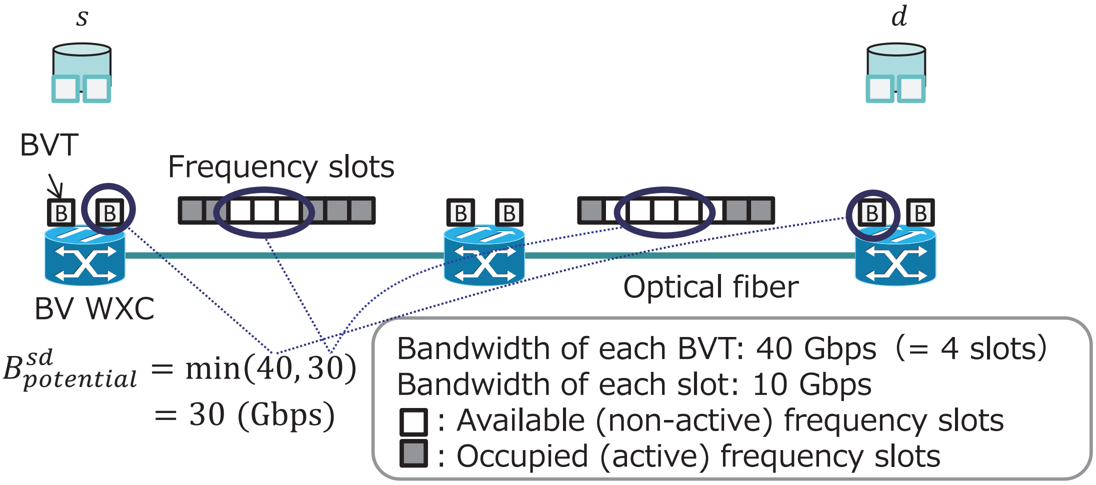

s
d
BVT
Frequency slots
B
BV WXC
Optical fiber
$B^{sd}_{potential} = \min(40, 30)$
$= 30$ (Gbps)
Bandwidth of each BVT: 40 Gbps （= 4 slots）
Bandwidth of each slot: 10 Gbps
□ : Available (non-active) frequency slots
■ : Occupied (active) frequency slots

Figure 3.4: Example of the potential bandwidth for a node pair (s,d): Case where a lightpath is *not* established between s and d

where α_{mlu} indicates the condition of the VN in terms of the service quality on the VN, and α_{pb} indicates the condition in terms of preservation of resources. We convert the maximum link utilization on the VN u_{max} into α_{mlu} by using Eq. (3.4) below. The value of α_{mlu} is in the range $[0, 1]$ and the constant value u_{maxth} is the target value of the maximum link utilization. When the maximum link utilization is less than the threshold u_{maxth}, α_{mlu} rapidly approaches 1. The constant number δ_{mlu} determines the gradient of the function. The conversion equation is

$$\alpha_{mlu} = \frac{1}{1 + \exp(\delta_{mlu} \cdot (u_{max} - u_{maxth}))}. \tag{3.4}$$

We also convert the potential bandwidth $B_{potential}$ into α_{pb} by using Eq. (3.5) below. The value of α_{pb} is in the range $[0, 1]$ and the constant number θ_{pb} is the target value of the potential bandwidth. When the potential bandwidth is more than the target value θ_{pb}, α_{pb} rapidly approaches 1. The constant value δ_{pb} determines the gradient of the function. The conversion equation is

$$\alpha_{pb} = \frac{1}{1 + \exp(\delta_{pb} \cdot (\theta_{pb} - B_{potential}))}. \tag{3.5}$$

The reason why we multiply α_{mlu} and α_{pb} together is that we aim to configure the VN to achieve the objectives for both the service quality on the VN and preservation of resources; in other words, we aim to configure a VN that can accommodate traffic demand on the VN while keeping some resources unused. The activity α takes a large value if and only if the both objectives are achieved; that is, α_{mlu} and α_{pb} take large values.

(Phase 1-2) Configuration of the virtual topology

We consider the state of the system $\mathbf{x}$ in the attractor selection as the state of all possible lightpaths that form the VN. That is, we decide whether or not to set up a lightpath l_i based on a state variable $x_i (\in \mathbf{x})$. The dynamics of the state variable x_i are defined by

$$\frac{dx_i}{dt} = \alpha \cdot \left(\varsigma \left(\sum_j W_{ij} x_j \right) - x_i \right) + \eta. \tag{3.6}$$

The activity α indicates the condition of the VN, which is calculated in Phase 1-1. The term $\varsigma(\Sigma_j W_{ij} x_j) - x_i$ represents the deterministic behavior where $\varsigma(z) = \tanh(\frac{\mu}{2} z)$ is a sigmoidal regulation function and μ is the parameter of the sigmoidal function. The first term is calculated using a regulatory matrix W_{ij}. The second term η represents the stochastic behavior and is white Gaussian noise with a mean value of zero. After x_i is updated based on Eq. (3.6), we decide whether or not to set up the lightpath l_i. Specifically, we set the threshold to zero and if x_i is greater than or equal to the threshold, we set up the lightpath l_i and otherwise remove the lightpath l_i.

We set the regulatory matrix so that it has a set of virtual topology candidates as attractors. That is, we set the regulatory matrix $\mathbf{W}$ so that $d\mathbf{x}/dt$ in Eq. (3.1) is equal to zero when the virtual topology reconfigured by our VN reconfiguration method $\mathbf{x} = (x_1, \cdots, x_n)$ is one of the attractors. For the attractors that are stored in the regulatory matrix, we use a method to decide the regulatory matrix using the pseudoinverse matrix, which is shown in Ref. [52]. Specifically, assuming that we set m VN candidates as attractors and one of the candidates is represented by $\mathbf{x}^{(k)} = (x_1^{(k)}, \cdots, x_n^{(k)})(1 \leq k \leq m)$, the regulatory matrix that has m attractors is

$$\mathbf{W} = \mathbf{X}^{+}\mathbf{X}, \tag{3.7}$$

where $\mathbf{X}$ is a matrix that has $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, ..., \mathbf{x}^{(m)}$ in each row and $\mathbf{X}^{+}$ is the pseudoinverse matrix of $\mathbf{X}$.

In these dynamics, the VN is reconfigured so that the activity α takes a large value. In other words, it is expected that our method can reconfigure a VN to improve the service quality on the VN while keeping some resources unused. Note that we can extend the definition of the activity α by using multiple performance metrics.

(Phase 1-3) Allocation of frequency slots

We now allocate frequency slots to the lightpaths we established in Phase 1-2. We set the bandwidth of these lightpaths to the maximum bandwidth a BVT can provide, and allocate the number of frequency slots corresponding to that bandwidth. For example, assuming that the bandwidth a BVT can offer is 100 Gbps and the bandwidth per frequency slot is 10 Gbps, the number of frequency slots we allocate to the lightpath in this phase is 10. At this point, we introduce an existing heuristic algorithm for allocating frequency slots for simplicity. We use the longest path first ordering algorithm [57] to determine the order of lightpaths to which frequency slots are allocated. We sort the lightpaths by the number of links on the shortest path in the physical topology (i.e., the number of physical hops), and allocate frequency slots first to the lightpath with the largest number of physical hops. We also use the first-last fit algorithm [58] for allocating frequency slots to the lightpath. An outline of this algorithm is shown in Fig. 3.5. In this algorithm, all frequency slots of each optical fiber are divided into a number of partitions, and we allocate a lightpath frequency slots from the partition that has the largest number of available frequency slots.

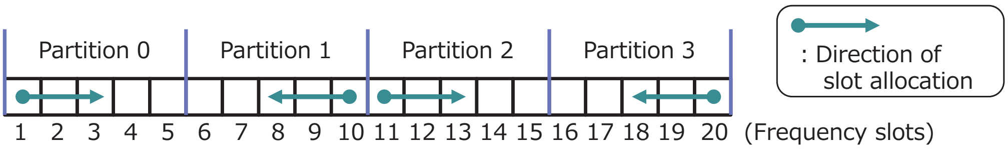

Partition 0
Partition 1
Partition 2
Partition 3
: Direction of slot allocation
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 (Frequency slots)

Figure 3.5: First-last fit algorithm

In even-numbered partitions, we select the lowest indexed slots from the list of available frequency slots, and in odd-numbered partition, we select the highest indexed slots from the list of available frequency slots.

3.3.4 (Phase 2) Adjustment of Lightpath Bandwidth

We adjust the bandwidth of each lightpath based on its link utilization. The link utilization of each lightpath is expected to be in the range $[u_{minth}, u_{maxth}]$, where u_{minth} is the lower limit of the target value of the link utilization and u_{maxth} is the upper limit of the target value of the link utilization. When the link utilization of a lightpath is lower than the lower limit value u_{minth}, it is considered that excessive allocation of frequency slots has caused degradation of spectrum utilization efficiency. Poor spectrum utilization efficiency causes increased power consumption, and decreases the amount of bandwidth that can be offered for leased lightpaths and for accommodating increased traffic demand on the VN. When the link utilization of a lightpath is higher than the upper limit u_{maxth}, it is considered that a high-loaded lightpath has caused degradation of the service quality on the VN. Therefore, we adjust the bandwidth of each lightpath l_i as follows:

- When the link utilization of the lightpath l_i, denoted by u_i, is less than u_{minth}, we reduce the number of frequency slots allocated to l_i so that u_i is higher than u_{minth}.

- When u_i is greater than u_{maxth}, we increase the number of frequency slots allocated to l_i so that u_i is lower than u_{maxth}. However, we consider spectrum contiguity and continuity constraints when adding frequency slots, and the number of frequency slots allocated to one lightpath does not exceed the number of slots corresponding to the

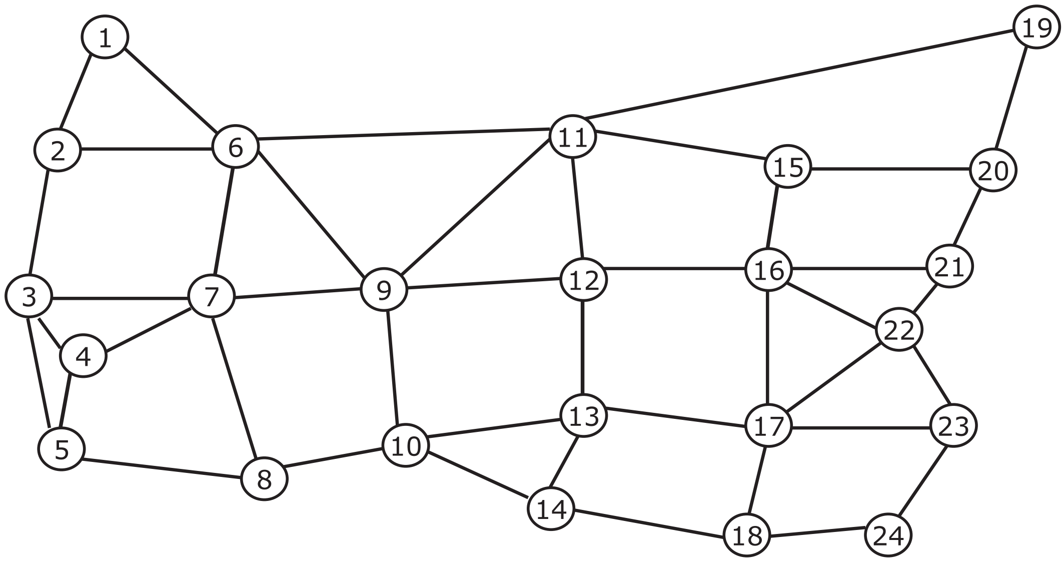


Figure 3.6: USNET

maximum bandwidth the BVT can provide.

When we adjust the number of frequency slots, we fix the central frequency (CF) and symmetrically allocate or release the slots following the Semi-Elastic scheme proposed in Ref. [59].

3.4 Performance Evaluation

In this section, we evaluate the performance of the VN reconfiguration method proposed in Section 3.3.

3.4.1 Evaluation Using USNET

We first evaluate the performance of our method using the USNET topology shown in Fig. 3.6. Table 3.1 shows the parameters related to the physical network topology. The number of nodes (i.e., BV WXCs) is 24, and the number of links (i.e., bidirectional optical fibers)

Table 3.1: Physical network topology-related parameters (USNET)

Parameter	Value
Number of nodes	24
Number of links	43
Number of BVTs	10
Bandwidth of each BVT	100 Gbps

Table 3.2: Target values for VN reconfiguration (USNET)

Parameter	Target value
u_{minth}	0.2
u_{maxth}	0.8
θ_{pb}	11040

is 43. Each BV WXC has 10 BVTs that can offer up to 100 Gbps of bandwidth. When we calculate the activity in Phase 1-1, we set $\delta_{mlu}, \delta_{pb}$ to 50 in Eqs. (3.4) and (3.5). Table 3.2 shows the target values for VN reconfiguration. We set the target value of the link utilization to the range [0.2, 0.8] and the target value of the potential bandwidth θ_{pb} to $24 \times (24 - 1) \times 100 \times 0.2 = 11040$, since we intend to set aside 20% of the bandwidth a BVT can offer between every node pair on average. When we reconfigure the virtual topology based on Eq. (3.6) in Phase 1-2, we set μ of the sigmoidal function $\varsigma(z)$ to 20. We also set the regulatory matrix W such that it contains virtual topology candidates designed by the method [36] as attractors. At the beginning of computer simulation, we configure one of the candidates as the initial virtual topology, and allocate frequency slots by following the procedure at Phase 1-3. Table 3.3 shows the parameters related to frequency slot allocation. We assume that the available spectrum width of each optical fiber is 4.75 THz and set the spectrum width of each frequency slot to 12.5 GHz. That is, the number of available frequency slots per optical fiber is 380. Each frequency slot has a bandwidth of 10 Gbps. When we use the first-last fit algorithm in Phase 1-3, we divide the spectrum width of each optical fiber into 4 partitions, so each partition has 95 frequency slots. We set the guard band between occupied frequency slots of adjacent lightpaths to 12.5 GHz,

Table 3.3: Frequency slot allocation-related parameters (USNET)

Parameter	Value
Spectrum width of each optical fiber	4.75 THz
Spectrum width of each frequency slot	12.5 GHz
Bandwidth of each frequency slot	10 Gbps
Number of partitions	4
Guard band	12.5 GHz

which corresponds to one frequency slot.

For the evaluation, the initial traffic demand between each node pair is chosen in the range [0.0, 1.5] Gbps from a uniform random distribution. We then increase the traffic demand of each node pair by a capacity chosen in the range [0.0, 0.01] Gbps from a uniform random distribution at each step of VN reconfiguration. We compare our method to a method that configures the VN to accommodate only the current traffic demand using all information of traffic demand between every node pair. Specifically, we introduce the most subcarriers first (MSF) algorithm [57] in order to determine the virtual topology, and the first-last fit algorithm in order to allocate frequency slots to the lightpaths that form the VN. The MSF algorithm establishes lightpaths between node pairs selected in ascending order of number of requested frequency slots. In this evaluation, the reference method collects information about the traffic demand between each node pair by long-term measurements. The method then sets up lightpaths and allocates frequency slots to lightpaths between node pairs in ascending order of traffic volume. The bandwidths of the lightpaths established by this reference method are the maximum bandwidth a BVT can provide. That is, this reference method does not adjust the bandwidths of the lightpaths.

Fig. 3.7 shows the potential bandwidth at each step of VN reconfiguration. The horizontal axis shows the number of steps of VN reconfiguration and the vertical axis shows the potential bandwidth at each step. The dotted line indicates the target value of the potential bandwidth θ_{pb}. At each even-numbered step, our proposed method enters Phase 1 (i.e., reconfigures the virtual topology), and at each odd-numbered step, it enters Phase 2 (i.e., adjusts the bandwidths of the lightpaths). The reference method, which is denoted by

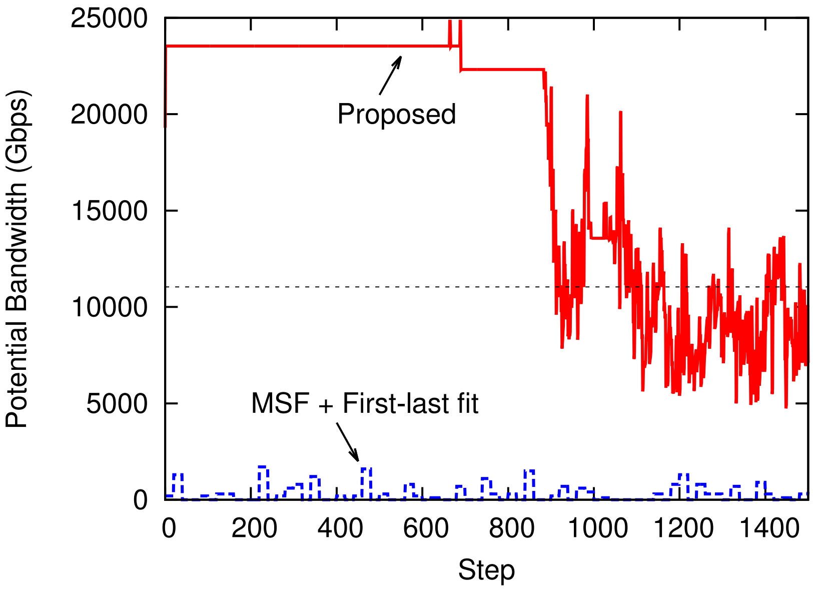
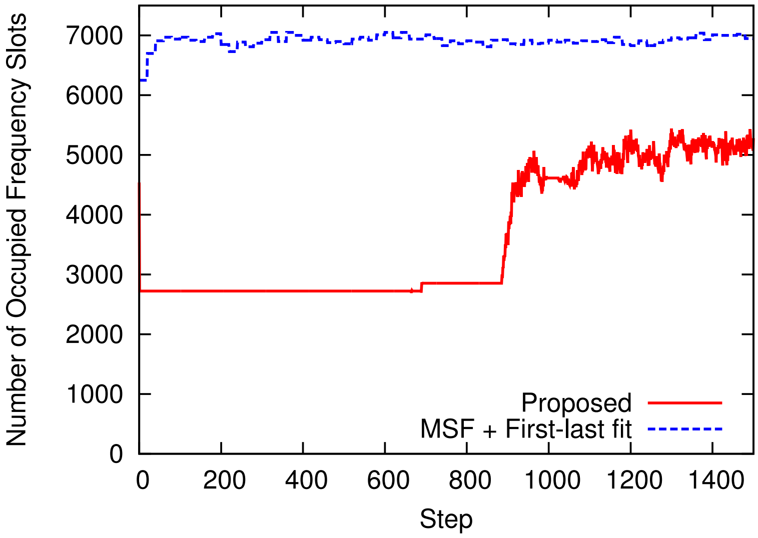


Figure 3.7: Potential bandwidth (USNET)

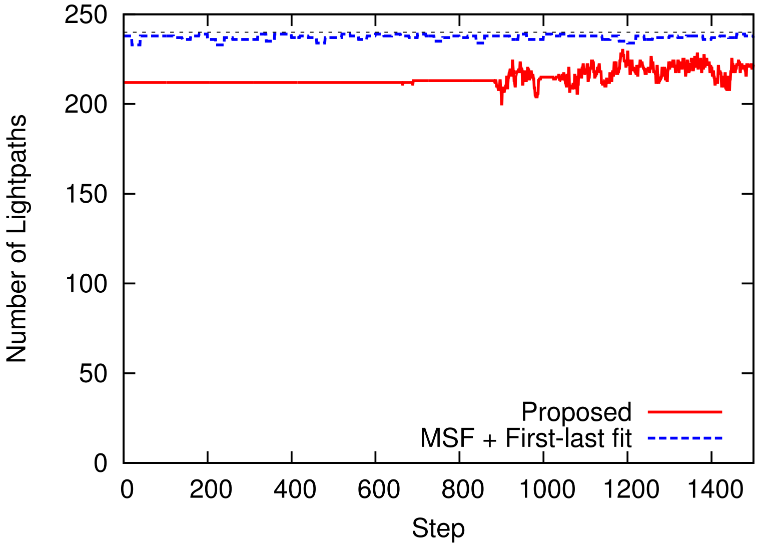
"MSF+First-last fit" in the figure, reconfigures the VN every 20 steps based on information about traffic demand between every node pair. Fig. 3.7 shows that the potential bandwidth in our method is kept higher than the target value until around step 900. The reason for the gradual decrease in the potential bandwidth is that our method expands the bandwidths of the lightpaths (i.e., adds frequency slots to lightpaths) in accordance with increases in traffic demand. In contrast, the potential bandwidth in the reference method takes a very small value. This is because the reference method establishes as many lightpaths as possible and does not adjust the bandwidths of the lightpaths in order to accommodate the current traffic demand. The above suggests that our method can set aside resources for leased lightpaths and to accommodate increased traffic demand on the VN until around step 900.

In terms of the direct effects of the high potential bandwidth maintained by the proposed method, Fig. 3.8 shows the number of resources used (i.e., frequency slots and BVTs). Fig. 3.8(a) shows the number of occupied (i.e., active) frequency slots at each step of VN reconfiguration. The horizontal axis shows the number of steps of VN reconfiguration and the vertical axis shows the total number of occupied frequency slots at each step. We can see that the number of occupied frequency slots with the proposed method is smaller than half of that with the reference method until step 900. This is because the proposed method adjusts the bandwidth of each lightpath based on the link utilization in Phase 2 in order to accommodate the current traffic demand, whereas the reference method does not. The reason for the gradual increase in the number of occupied frequency slots with the proposed method is that additional frequency slots are allocated to the lightpaths adaptively in response to increases in traffic demand. Fig. 3.8(b) shows the number of lightpaths that form the VN (i.e., the number of used BVTs) at each step of VN reconfiguration. The horizontal axis shows the number of steps of VN reconfiguration and the vertical axis shows the total number of lightpaths at each step. The dotted line indicates the maximum number of lightpaths that can be established to configure the VN. Fig. 3.8(b) shows that the number of lightpaths established by the proposed method is consistently lower than that by the reference method. In other words, the proposed method configures the VN by using fewer BVTs. This is because the proposed method reconfigures the virtual topology by using the potential bandwidth as the activity in Phase 1 in order to keep more resources unused. By keeping a high potential bandwidth, the proposed method can set aside resources for leased lightpaths and to accommodate increased traffic demand on the VN (i.e., reduce the number of resources used to configure the VN).

Fig. 3.9 shows the maximum link utilization at each step of VN reconfiguration. The horizontal axis shows the number of steps of VN reconfiguration and the vertical axis shows the maximum link utilization of the VNs configured at each step. The upper dotted line indicates the upper limit value u_{maxth} and the lower dotted line indicates the lower limit value u_{minth}. In Fig. 3.9, we can see that the maximum link utilization by our method rises at first, but is kept below the upper limit value by VN reconfiguration. That is, our method can reconfigure the VN so that it can accommodate changing traffic demand. Note that the potential bandwidth takes a higher value than the target value θ_{pb}, as shown in Fig. 3.7, when our method reconfigures the VN to accommodate the traffic demand. The maximum link utilization by the reference method rises gradually as the traffic demand increases. At



(a) Number of occupied frequency slots



(b) Number of lightpaths (number of used BVTs)

Figure 3.8: Number of used resources (USNET)

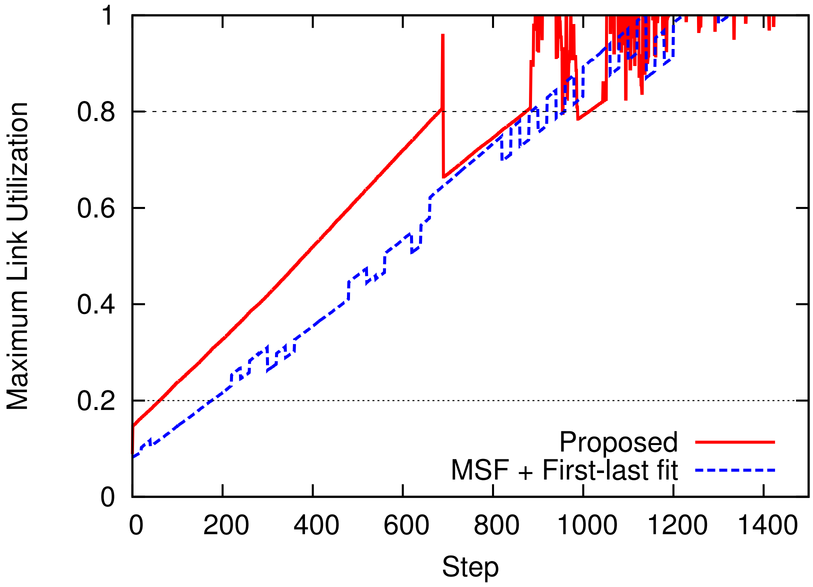


Figure 3.9: Maximum link utilization (USNET)

about step 900, the maximum link utilization by the reference method reaches almost the same value as the proposed method. That is, the VN configured by the proposed method can improve the service quality on the VN to the same degree as the VN configured by the method that uses all information of the traffic demand matrix.

Fig. 3.10 shows the distribution of the number of VN reconfiguration steps until convergence by the proposed method for 1,000 patterns of traffic demand. We generated these traffic demand patterns by increasing the demand between each node pair until the maximum link utilization by the reference method rose above the upper limit value u_{maxth}. We consider the proposed VN reconfiguration method to have succeeded if the VN reconfiguration meets the target values for both maximum link utilization and potential bandwidth (i.e., the VN reconfiguration finds a solution) within 10 successive steps of VN reconfiguration. We assume that the traffic demand is given at time zero, and evaluate the number of steps of VN reconfiguration required until the VN reconfiguration successfully converges. The horizontal axis shows the number of steps until convergence and the vertical axis shows the frequency of the number of steps until convergence. From Fig. 3.10, we can see that our

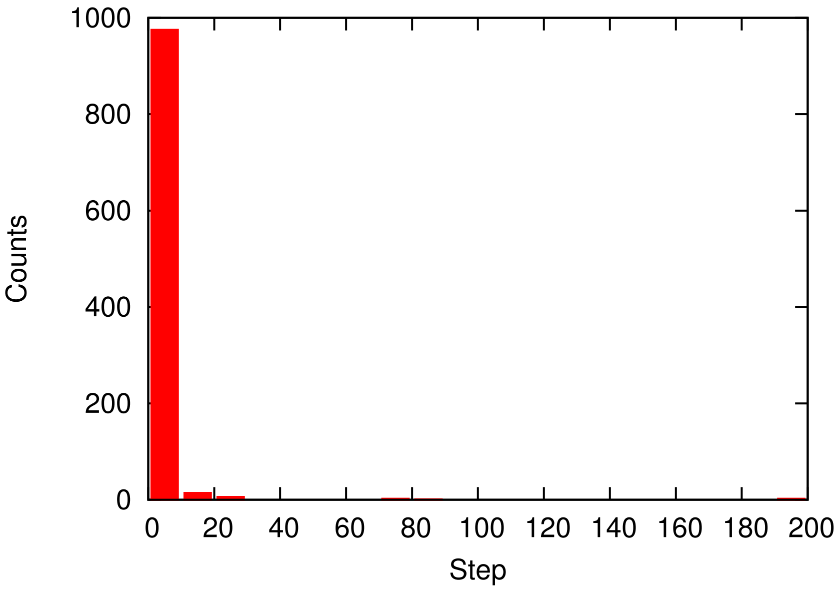


Figure 3.10: Distribution of the number of steps until convergence (USNET)

method can find a solution within 20 steps for 992 traffic patterns. Our method can thus achieve target values of both maximum link utilization and potential bandwidth for a wide variety of traffic demand without all of the information from the traffic demand matrices.

The above evaluation shows that the proposed method can configure a VN that improves the service quality on the VN while setting aside resources for leased lightpaths and accommodating increased traffic demand on the VN.

3.4.2 Evaluation Using the Simple CAIS Internet

We next evaluate the performance of our method using the Simple CAIS Internet topology, shown in Fig. 3.11. This topology has bottleneck links in terms of the allocation of frequency slots, such as the link between nodes 2 and 5. Table 3.4 shows the parameters of the physical network topology. The number of nodes is 16 and the number of links is 23. Each BV WXC has 8 BVTs that can offer up to 100 Gbps of bandwidth. Table 3.5 shows the target values for VN reconfiguration. We set the target value of the potential bandwidth

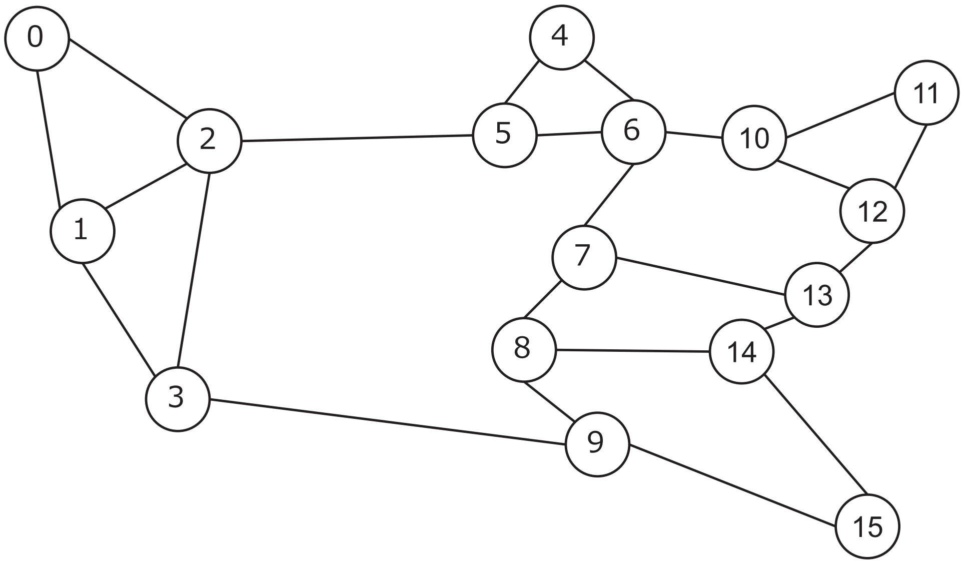


Figure 3.11: Simple CAIS Internet

Table 3.4: Physical network topology-related parameters (Simple CAIS Internet)

Parameter	Value
Number of nodes	16
Number of links	23
Number of BVTs	8
Bandwidth of each BVT	100 Gbps

θ_{pb} to $16 \times (16-1) \times 100 \times 0.2 = 4800$, with the intention of setting aside 20% of the bandwidth a BVT can offer between every node pair on average. The other target values and parameters for reconfiguring the VN in Phase 1 are the same as in Section 3.4.1. The parameters related to frequency slot allocation are also the same as in Section 3.4.1 (shown in Table 3.3). For the evaluation, we generated 1,000 patterns of traffic demand in the same way as in Section 3.4.1, and compared our method to the same method as in Section 3.4.1.

Fig. 3.12 shows the distribution of the number of VN reconfiguration steps until convergence by the proposed method for 1,000 patterns of traffic demand. The horizontal axis shows the number of steps required until the VN reconfiguration successfully converges and

Table 3.5: Target values for VN reconfiguration (Simple CAIS Internet)

Parameter	Target value
u_{minth}	0.2
u_{maxth}	0.8
θ_{pb}	4800

the vertical axis shows the frequency of the number of steps. This figure shows that our method can find a solution within 20 steps for 910 traffic patterns. That is, our method can reconfigure the VN to improve the service quality on the VN while keeping unused resources for fewer traffic patterns than the evaluation in Section 3.4.1. This is because it is difficult for the proposed method to find a solution due to the bottleneck links in the allocation of frequency slots. When the proposed method removes some lightpaths and tries to newly establish lightpaths between other node pairs in Phase 1, it is likely that the method cannot set up new lightpaths because of restrictions on resources. Therefore, although our method reconfigures a VN to set aside resources, there are cases where it takes a long time to find a VN configuration that uses fewer resources. Note that our method can find a solution within 180 steps for a further 22 traffic patterns.

The above evaluation shows that our method can reconfigure the VN to improve the service quality on the VN while keeping unused resources for a wide variety of traffic demand. However, it is likely that it will take a long time for our method to find a solution when the physical network topology has bottleneck links in the allocation of frequency slots.

3.4.3 Effect of the Granularity of Frequency Slots

In this section, we evaluate the effect of the granularity of frequency slots. Specifically, we evaluate the performance of our method using USNET when the granularity of the frequency slots is coarser.

Table 3.6 shows the parameters related to frequency slots allocation. We assume that the available spectrum width of each optical fiber is 4.75 THz and set the spectrum width of each frequency slot to 25.0 GHz. That is, the number of available frequency slots per

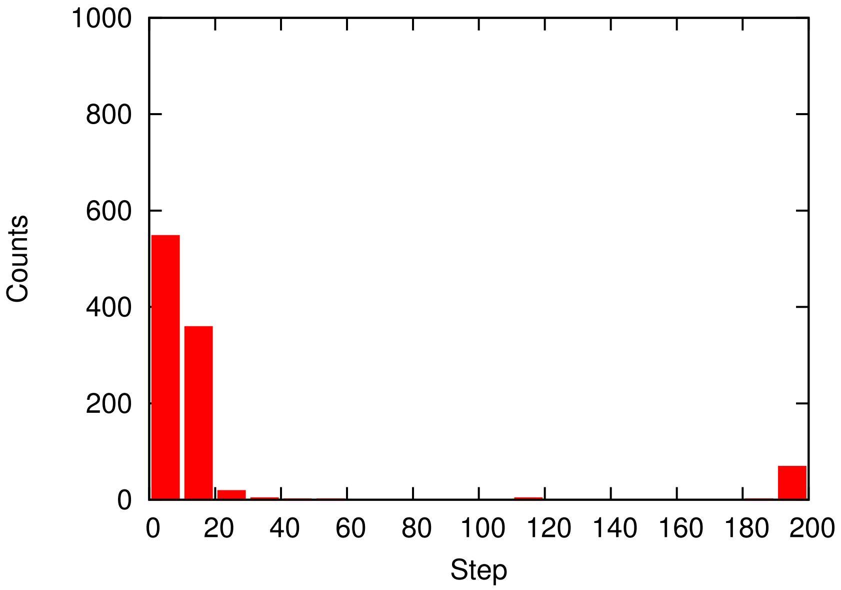


Figure 3.12: Distribution of the number of steps until convergence (Simple CAIS Internet)

Table 3.6: Frequency slot allocation-related parameters (USNET: coarse frequency slots)

Parameter	Value
Spectrum width of each optical fiber	4.75 THz
Spectrum width of each frequency slot	25.0 GHz
Bandwidth of each frequency slot	20 Gbps
Number of partitions	4
Guard band	25.0 GHz

optical fiber is 190, which is half of the number in Section 3.4.1. Each frequency slot has a bandwidth of 20 Gbps. We also set the guard band between occupied frequency slots to 25.0 GHz, which corresponds to one frequency slot. The other parameters related to the physical network topology and reconfiguration of the VN, and the target values, are similar to those in Section 3.4.1. For the evaluation, we use the same pattern of traffic demand and the reference method as in Section 3.4.1.

Fig. 3.13 shows the potential bandwidth at each step of VN reconfiguration. The horizontal axis shows the number of steps of VN reconfiguration and the vertical axis shows the potential bandwidth at each step. The dotted line indicates the target value of the potential bandwidth θ_{pb}. At each even-numbered step, the proposed method enters Phase 1 (i.e., reconfigures the virtual topology) and at each odd-numbered step, it enters Phase 2 (i.e., adjusts the bandwidths of the lightpaths). The method for comparison, which is denoted by "MSF+First-last fit" in the figure, reconfigures the VN every 20 steps based on the traffic demand information between every node pair. Fig. 3.13 shows that the potential bandwidth by the proposed method is kept higher than the target value until around step 700. However, after around step 700, the potential bandwidth is lower than the values in Fig. 3.7, which shows the potential bandwidth in the case where the granularity of frequency slots is finer. In other words, when the granularity of the frequency slots is coarser, it is more difficult to keep a high potential bandwidth in cases where the traffic loads are large. Our method expands the bandwidths of the lightpaths (i.e., adds frequency slots to lightpaths) according to the increased traffic demand. However, since the granularity of frequency slots is coarse, the potential bandwidth that can be additionally offered is sharply reduced. That is, coarser granularity of frequency slots makes it more difficult to set aside resources for leased lightpaths and accommodating increased traffic demand on the VN in cases where the traffic loads are large.

Fig. 3.14 shows the distribution of the number of VN reconfiguration steps until convergence by the proposed method for 1,000 patterns of traffic demand. We generated these traffic demand patterns by increasing the demand between each node pair until the maximum link utilization by the reference method rose above the upper limit value u_{maxth}. The horizontal axis shows the number of steps required until the VN reconfiguration successfully converges and the vertical axis shows the distribution of the number of steps. This shows that our method can find a solution within 20 steps for 991 traffic patterns. That is, our method can reconfigure the VN to improve the service quality on the VN while keeping unused resources for slightly fewer traffic patterns than the evaluation in Section 3.4.1. This is because it is difficult for the proposed method to find a solution, particularly one that keep the potential bandwidth high, because of the coarse granularity of the frequency slots.

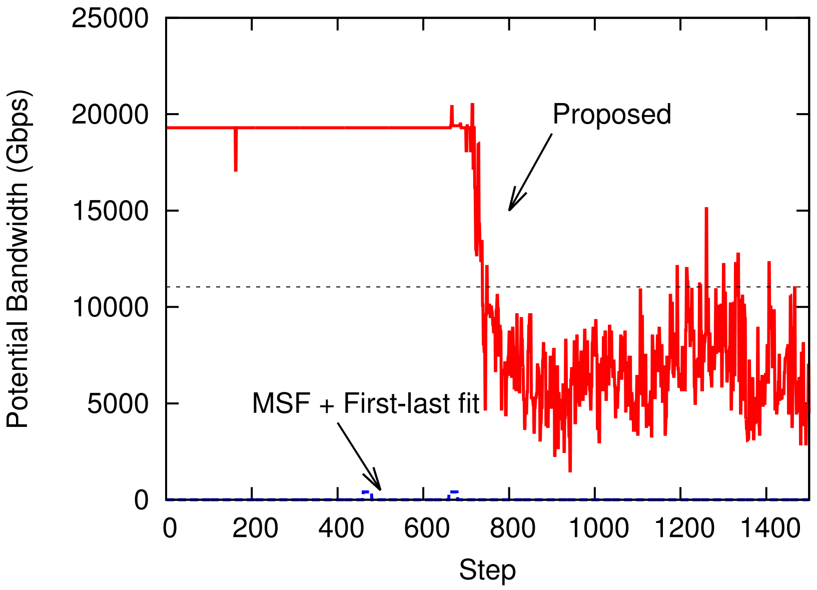


Figure 3.13: Potential bandwidth (USNET: coarse frequency slots)

From the above evaluation, we find that a coarser granularity of frequency slots makes it more difficult for the proposed method to keep a high potential bandwidth while maintaining service quality on the VN. In other words, it is easier for our method to achieve the objectives when the granularity of frequency slots is finer. Since there are efforts to breaking down the spectrum width of optical fiber into even more frequency slots [3,4], this tendency suits our method.

3.5 Conclusion

In this chapter, we proposed a resource-efficient VN reconfiguration method. Our method reconfigures a virtual topology based on attractor selection using only information about the traffic loads on every lightpath and the potential bandwidth, which we define in this chapter as a metric that reflects the bandwidth that can be additionally offered, and then the method adjusts the bandwidth of the lightpaths that form the VN. We showed that our method can configure a VN with less resources (i.e., frequency slots and BVTs) while

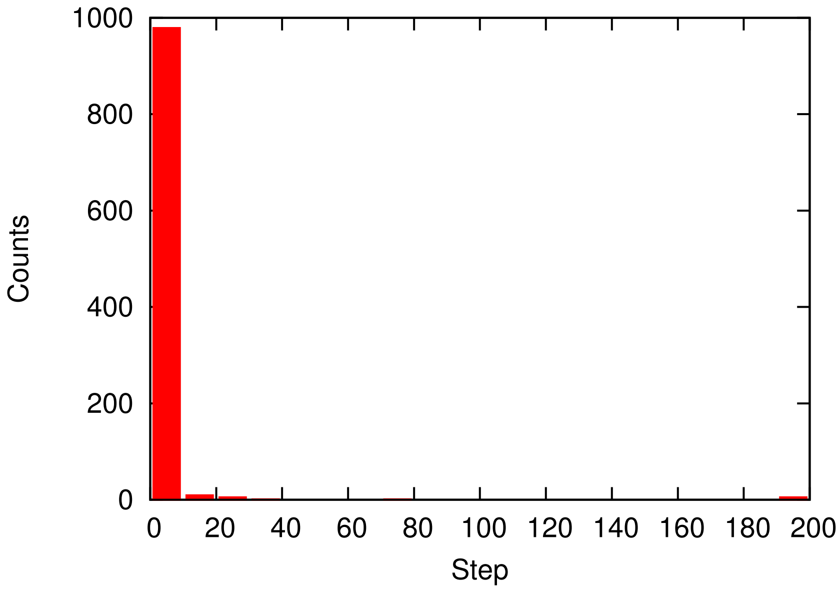


Figure 3.14: Distribution of the number of steps until convergence (USNET: coarse frequency slots)

improving the service quality on the VN to almost the same degree as a VN configured by using all information of the traffic demand matrix. By reconfiguring a VN with less resources, the network operator can provide more bandwidth to meet future requests for leased lightpaths and accommodate increased traffic demand on the VN.

One future direction for this work is how to design the physical networks (i.e., elastic optical networks) and how to enhance the network equipment. As shown in Section 3.4.2, it is likely that bottleneck links in terms of allocation of frequency slots will make it difficult to rapidly configure a VN that can accommodate traffic demand while setting aside resources. It is therefore expected that our method will be able to find a solution in a shorter time period if the physical network is better designed.

Chapter 4

Virtual Network Reconfiguration Based on Bayesian Attractor Model

4.1 Introduction

Changes in the environment surrounding the Internet in recent years, such as advances in personal Internet-enabled devices and the emergence of new Internet services, has led to rapid growth and large fluctuations in Internet traffic. Specifically, Ref. [1] points out that the volume of Internet traffic has grown by 20 times over the past decade, and will increase further in the future. Thanks to the large bandwidth of optical fibers, optical networks have the potential to support this growing traffic demand.

Network virtualization [6] is one of key technologies that allows network operators to make full use of network infrastructures, such as optical networks. In response to requests from service providers, network operators construct a virtual network (VN) in a dynamical manner by slicing physical resources such as wavelength in Wavelength Division Multiplexing (WDM)-based networks or frequency slots in elastic optical networks. A VN consists of a set of optical connections (i.e., lightpaths) and client nodes (e.g., IP routers), and provides

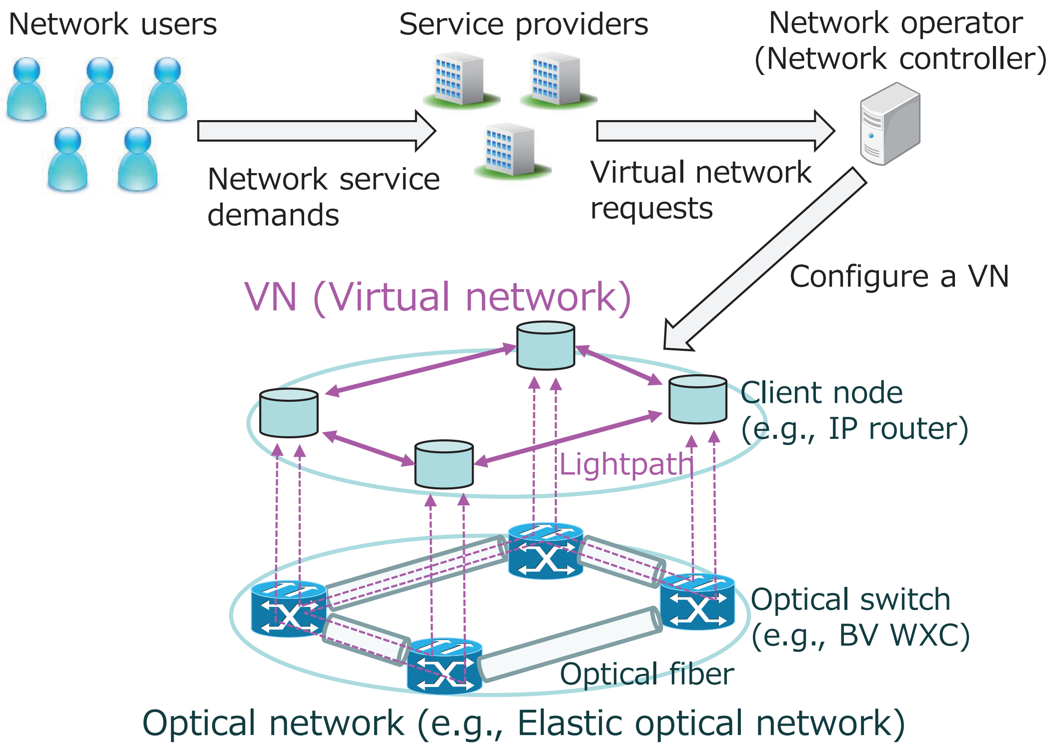


Figure 4.1: Network Virtualization in Optical Networks

the connectivity for network equipment and the bandwidth to accommodate traffic demand. That is, the VN accommodates network services, such as video-on-demand services or cloud computing services, offered by the service providers, and the network users enjoy the network services (see Fig. 4.1). When fluctuations in traffic demand cause temporary traffic congestion, it is necessary to reconfigure the VN so that the congestion is resolved and the VN can accommodate the new traffic demand.

A typical approach for constructing or reconfiguring a VN is to design an optimal virtual topology and allocate resources using knowledge of the end-to-end traffic demand matrix [19–22, 24, 25]. The methods for configuring a VN proposed in Refs. [19–22] use directly observed information about the traffic demand matrix as an input. However, it generally takes a long time and requires a large amount of CPU resources to directly retrieve information about the traffic demand matrix. As a result, the optimization approach has difficulty in reconfiguring a VN to follow short-term changes in traffic demand. Refs. [24,25] therefore proposed methods to configure a VN using an estimated traffic demand matrix. Many researches also have been devoted to investigating methods to estimate the traffic demand matrix, such as Refs. [26–31]. However, estimation errors in the traffic demand matrix are inevitable. When we configure a VN using the estimated traffic demand matrix including estimation errors and the VN cannot accommodate traffic demand, we do not have a way to configure the optimal VN because we have incorrect knowledge of the traffic demand matrix. Therefore, it is difficult for the VN reconfiguration approach using the information of the traffic demand marix to adapt to traffic fluctuations.

We previously have proposed an attractor selection-based VN reconfiguration method [41] that does not use the traffic demand matrix. The attractor selection-based method observes only the service quality, such as load information on all lightpaths that form a VN, and searches for "good" VNs (i.e., VNs that can accommodate traffic demand) by repeatedly making random changes to the current one. The link-level load information can be retrieved in a much shorter time, typically 5 minutes or less, than the end-to-end traffic demand matrix. We have shown in Ref. [41] that, in principle, the attractor selection-based method can effectively accommodate changing traffic demand. However, in practice, the attractor selection-based method repeatedly reconfigures a VN, which leads to over-reconfiguration. The over-reconfiguration disrupts network services accommodated on the VN. Therefore, it is desirable to minimize the number of VN reconfigurations needed to adapt to traffic changes.

In this chapter, we propose a VN reconfiguration method without using the traffic demand matrix that adapts to traffic fluctuations yet avoids over-reconfiguration with its consequent disruption of network services accommodated on the VN. Our basic idea is to follow the human behavior of making appropriate decisions by recognizing the surrounding situation. Current consensus in cognitive science states that the brain accumulates sensory information over a period of time, and makes a perceptual decision (i.e., categorizes observed information) once enough information has been collected [46,60,61]. That is, humans recognize their surroundings as belonging to one of several environmental types and make appropriate decisions. We employ the Bayesian Attractor Model (BAM) [46], which models this behavior using the concept of Bayesian inference [62], in our VN reconfiguration method. In our VN reconfiguration method, we define a set of pre-specified traffic situations using certain patterns of incoming and outgoing traffic at edge routers, which can be obtained more easily than all information of the traffic demand matrix [26]. We also prepare a set of VN candidates, each of which works well for a certain traffic situation. The VN reconfiguration method updates the probabilities for taking each specified situation as we observe the amounts of incoming and outgoing traffic at edge routers, and identifies the best representation of the current traffic situation using the BAM. When identification of the current traffic situation succeeds, the corresponding VN candidate suitable for the identified traffic situation is retrieved and configured. However, the retrieved VN may not be able to accommodate the actual traffic demand. In preparation for that eventuality, we incorporate the attractor selection-based method [41] into our VN reconfiguration framework in order to search for good VNs. Our VN reconfiguration framework is more stable than using the attractor selection-based method alone, since the retrieved VN will not be changed unless it cannot accommodate traffic demand.

Furthermore, we extend the above VN reconfiguration framework in order to deal with the case where identification of traffic situations fails. Although the BAM-based method can retrieve a VN suitable for the current traffic situation when the identification of traffic situations succeeds, there is no choice but to apply the attractor selection-based VN reconfiguration method when the identification fails, which leads to over-reconfiguration. We therefore propose a VN reconfiguration method to deal with the case where the identification fails, and incorporate this method into our VN reconfiguration framework. In this method, we use a set of pre-specified traffic situations, and the current traffic situation is fitted by linear regression, and then our method calculates and configures a VN using the obtained regression coefficients. Similar to the above VN reconfiguration framework, we apply the attractor selection-based method when the configured VN cannot accommodate traffic demand. However, by introducing this method, it is expected that the number of VN reconfiguration to accommodate traffic demand can be reduced even when the identification of traffic situations fails, since we calculate a VN that is suitable to some extent using the information obtained by fitting the current traffic situation. We also investigate how to select and update the set of pre-specified traffic situations.

The rest of this chapter is organized as follows. In Section 4.2, we first describe related work. We describe the BAM in Section 4.3, and explain the VN reconfiguration framework based on the BAM in Section 4.4, and then we discuss the advantages and behavior of our VN reconfiguraiotn framework in Section 4.5. Furthermore, we explain the extend VN reconfiguration framework based on the BAM with linear regression in Section 4.6, and evaluate its effectiveness in Section 4.7. We discuss how to select and update a set of pre-specified traffic situations in Section 4.8, and we give our conclusions in Section 4.9.

4.2 Related Work

There are many studies for constructing or reconfiguring a VN for optical networks. Basically, given the traffic demand matrix, they aim to configure a VN that achieves some objectives by using mixed integer linear programming (MILP) or a heuristic algorithm. For example, Refs. [19–22] investigate VN reconfiguration methods over elastic optical networks. Ref. [19] solves the routing and spectrum assignment (RSA) problem using a heuristic algorithm for configuring a couple of VNs over a multi-domain elastic optical path network. Using information about the traffic demand for each VN, the algorithm tries to minimize the total network cost, including the costs of transponders, regenerators, and spectrum resources. Ref. [20] considers the modulation level assignment problem in addition to the RSA problem. The heuristic algorithm proposed in Ref. [20] jointly solves the routing, modulation level, and spectrum assignment problems. Using traffic demand information, the algorithm minimizes total network costs, such as the costs of transponders and routers. Ref. [22] proposes a MILP formulation for several schemes of protection in cases where multiple VNs run over an elastic optical network.

However, these methods have difficulty in reconfiguring a VN following changes in traffic demand. This is because that they use directly observed information of the traffic demand matrix as an input, and in general, it takes a long time and requires a large amount of CPU resources to directly retrieve this information since traffic inspection is necessary to measure the volume for each source-destination pair.

Refs. [24, 25] therefore propose methods to configure a VN using a predicted traffic demand matrix. Ref. [24] uses a set of past traffic demand matrices as inputs, and predicts the traffic demand matrix in the near future by using the autoregressive integrated moving averages (ARIMA) technique. A multi-objective algorithm then reconfigures a VN using the predicted traffic demand matrix to minimize congestion, OPEX, and reconfiguration disruption. Ref. [25] proposes a VN reconfiguration method that aims to adapt to current and predicted traffic demand. This algorithm observes and stores the end-to-end traffic volume, and then predicts the traffic demand matrix using an artificial neural network (ANN)-based model.

Many researches have also been devoted to investigating methods to estimate the traffic demand matrix, such as Refs. [26–31]. Ref. [26] estimates the traffic demand matrix by fitting link-level load information, which is easy to collect, to a specific traffic model. In order to reduce the estimation errors in the traffic demand matrix, Ref. [29] proposes several methods to obtain additional information for estimation, such as both long-term and short-term traffic variability, and the variance of the end-to-end traffic volume. Ref. [31] proposes a method for estimating the traffic demand matrix in IP-over-WDM backbone networks using the traffic demand data in the optical layer as well as the link-level load information in the network layer. Ref. [30] estimates the traffic demand matrix by using a neural network that has learned the past traffic demands.

However, it is difficult for the optimization approach that uses an estimated traffic demand matrix to reconfigure a VN so that it can accommodate changing traffic demand. Since the approaches for estimating traffic demand matrices fit the collected information to a specific traffic model or past traffic data, they cannot deal with irregular traffic fluctuation. Although a variety of engineering techniques can reduce estimation errors, they do not guarantee the accuracy of the estimation: that is, estimation errors in the traffic demand matrix are inevitable. When we configure a VN with the traffic demand matrix including estimation errors and the VN cannot accommodate traffic demand, we do not have a way to configure the optimal VN because we have incorrect knowledge of the traffic demand matrix.

4.3 Bayesian Attractor Model

In this section, we explain the Bayesian Attractor Model (BAM) [46] that represents the human behavior of recognizing their surroundings and make appropriate decisions.

4.3.1 Outline of the Bayesian Attractor Model

The Bayesian Attractor Model (BAM) models the behavior of a human brain that accumulates sensory information over a period of time, and makes a perceptual decision (i.e., categorizes observed information) once enough information has been collected. For example, when a traffic light turns green at an intersection, we recognize that we see a green light and make the decision to proceed. More precisely, the color categories of traffic lights are retained in our brains, and we judge which category the observed sensory information belongs to. Even when the sensory information contains a large amount of noise, e.g., it is hard to see a traffic light due to bad weather or backlight, the brain takes time to accumulate evidence extracted from this noisy sensory information and make appropriate perceptual decisions.

Fig. 4.2 illustrates the outline of the BAM. The BAM has a state variable, $\mathbf{z}$, that eventually settles into a fixed point, ϕ, that is defined by the attractor dynamics [63], (i.e., the winner-take-all dynamics) as evidence is accumulated. Internally, the BAM has several fixed points ϕ_i each of which corresponds to a choice for the long-term average, $\boldsymbol{\mu}_i$, of an observed value. At a time t, the model infers the posterior distribution of the state variable $\mathbf{z}_t$, denoted by $p(\mathbf{z}_t|\mathbf{X}_{1:t})$, given observations up to time t, denoted by $\mathbf{X}_{1:t} = \{\mathbf{x}_1, \cdots, \mathbf{x}_t\}$. Eventually, the model chooses $\boldsymbol{\mu}_i$ as soon as a confidence criterion such as

$$p(\mathbf{z}_t = \phi_i|\mathbf{X}_{1:t}) \geq \lambda, \tag{4.1}$$

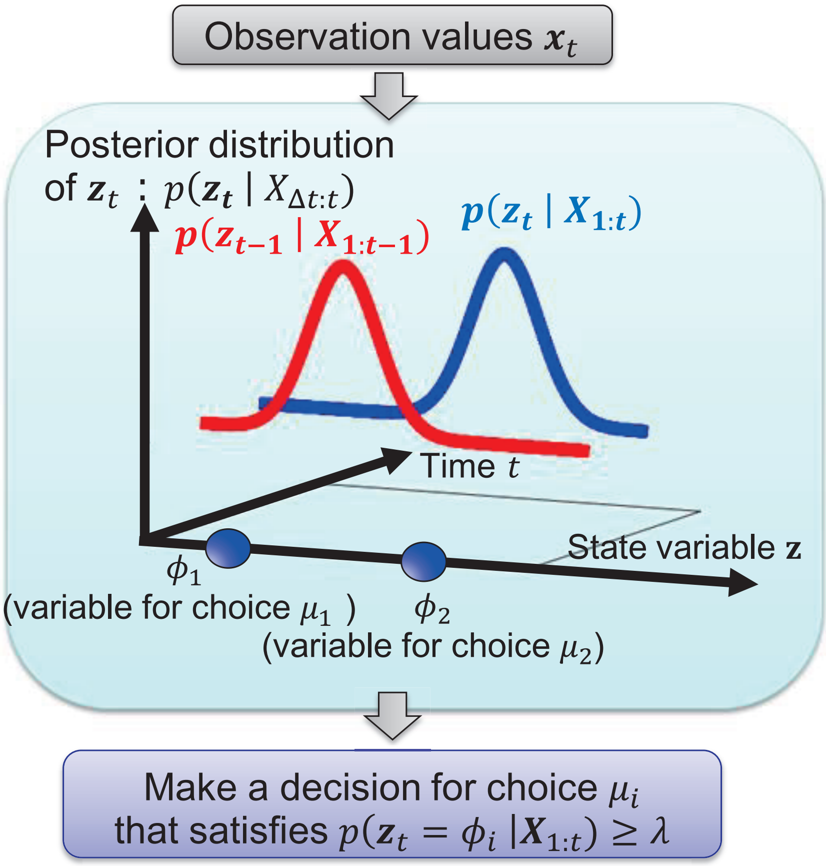

Observation values $\boldsymbol{x}_t$
Posterior distribution of $\boldsymbol{z}_t$: $p(\boldsymbol{z}_t \mid X_{\Delta t:t})$
$\boldsymbol{p(z_{t-1} \mid X_{1:t-1})}$
$\boldsymbol{p(z_t \mid X_{1:t})}$
Time t
State variable $\mathbf{z}$
ϕ_1
(variable for choice μ_1)
ϕ_2
(variable for choice μ_2)
Make a decision for choice μ_i that satisfies $p(\boldsymbol{z}_t = \phi_i \mid \boldsymbol{X}_{1:t}) \geq \lambda$

Figure 4.2: Outline of the Bayesian Attractor Model

is satisfied. Here, Ref. [46] introduces the posterior belief $p(\mathbf{z}_t = \phi_i | \mathbf{X}_{1:t})$ as the confidence measure for making a decision for the choice $\boldsymbol{\mu}_i$; in the VN reconfiguration framework, we use a different confidence measure. Thus, the BAM accumulates observation values and makes a decision when the confidence for the decision is large enough.

4.3.2 Inference Mechanism for Decision Making by the BAM

The BAM has a generative model for Bayesian inference by the decision maker (i.e., the brain). The generative model calculates the likelihood of observations under all possible choices that the decision maker considers. More precisely, the generative model predicts a probability distribution over observation values based on the current state variable and its attractor dynamics.

The generative model defines a change in the state variable from one time step to the next as

$$\mathbf{z}_t - \mathbf{z}_{t-\Delta t} = \Delta t \cdot f(\mathbf{z_{t-\Delta t}}) + \sqrt{\Delta t} \cdot \mathbf{w}_t, \tag{4.2}$$

where $\mathbf{z}_t$ is the D-dimensional state variable at a time t and $f(\mathbf{z})$ is the attractor dynamics [63]. The noise term $\mathbf{w}_t$ follows the normal distribution $N(\mathbf{0}, \mathbf{Q})$, where $\mathbf{Q} = (q^2/\Delta t)\cdot\mathbf{I}$ is the variance-covariance matrix of the noise and q is "dynamical uncertainty". This represents the amount of noise with which the decision maker expects the state variable to be changed, which is interpreted as the tendency for state variables to switch between fixed points.

The generative model predicts a probability distribution over observation values, given the state variable $\mathbf{z}$. The equation for the prediction is

$$\begin{aligned} \mathbf{x} &= \mathbf{M} \cdot \sigma(\mathbf{z}) + \mathbf{v} \qquad (4.3) \\ &= [\boldsymbol{\mu}_1, \cdots, \boldsymbol{\mu}_D] \cdot \sigma(\mathbf{z}) + \mathbf{v} \\ &= \sigma(z_1) \cdot \boldsymbol{\mu}_1 + \sigma(z_2) \cdot \boldsymbol{\mu}_2 + \cdots + \sigma(z_D) \cdot \boldsymbol{\mu}_D + \mathbf{v}, \end{aligned}$$

where $\mathbf{M} = [\boldsymbol{\mu}_1, \cdots, \boldsymbol{\mu}_D]$ contains the averages of observation values that correspond to choices and $\sigma(\mathbf{z})$ is the sigmoid function that maps all valuables $z_j \in \mathbf{z}$ to values between 0 and 1. Due to the winner-take-all dynamics of $\mathbf{z}$, the fixed point ϕ_i is mapped to a vector $\sigma(\phi_i)$ where one element is approximately 1 and the other elements are approximately 0. Thus, the linear combination $\mathbf{M} \cdot \sigma(\mathbf{z})$ associates each fixed point ϕ_i with the choice (the average of observations) $\boldsymbol{\mu}_i$. The noise term $\mathbf{v}$ follows the normal distribution $N(\mathbf{0}, \mathbf{R})$, where $\mathbf{R} = r^2\cdot\mathbf{I}$ is the variance-covariance matrix of the noise and r is "sensory uncertainty". This represents the amount of noise on observations that the decision maker expects. In contrast, the actual amount of noise by which observations deviate from the average values is denoted by s. We summarize the key parameters of the BAM in Table 4.1.

At a time t, the BAM infers the posterior distribution of the state variable $\mathbf{z}_t$, denoted by $p(\mathbf{z}_t|\mathbf{X}_{1:t})$, using the generative model and the unscented Kalman filter (UKF) [64].

Table 4.1: Key Parameters of the BAM

Parameter	Explanation
s (noise level)	the actual amount of noise on observation values
q (dynamics uncertainty)	the tendency for state variables to switch between fixed points
r (sensory uncertainty)	the amount of noise on observation values the decision maker expects

The UKF is a statistical sampling method that approximates the posterior distribution $p(\mathbf{z}_t|\mathbf{X}_{1:t})$ with a normal distribution. In the following, we briefly describe the flow of the Bayesian inference in the BAM. First, the generative model predicts the posterior distribution of the state variable at a time t using Eq. (4.2) and approximates it with a normal distribution $N(\hat{\mathbf{z}}_t, \hat{\mathbf{P}}_t)$, where $\hat{\mathbf{P}}_t$ represents the variance-covariance matrix of the predicted state variable, $\hat{\mathbf{z}}_t$. Second, the generative model predicts the possibility distribution of the corresponding observation values using Eq. (4.3) and approximates it with a normal distribution $N(\hat{\mathbf{x}}_t, \hat{\boldsymbol{\Sigma}}_t)$, where $\hat{\boldsymbol{\Sigma}}_t$ represents the variance-covariance matrix of the predicted observation values, $\hat{\mathbf{x}}_t$. Finally, the BAM calculates the observation residual between the predicted observation values $\hat{\mathbf{x}}_t$ and the actual observation values $\mathbf{x}_t$,

$$\epsilon_t = \mathbf{x}_t - \hat{\mathbf{x}}_t, \tag{4.4}$$

and updates the estimation of the state variable $\bar{\mathbf{z}}_t$ and its posterior variance-covariance matrix $\bar{\mathbf{P}}_t$ via a Kalman gain $\mathbf{K}_t$ as follows.

$$\bar{\mathbf{z}}_t = \hat{\mathbf{z}}_t + \mathbf{K}_t \cdot \epsilon_t, \tag{4.5}$$

$$\bar{\mathbf{P}}_t = \hat{\mathbf{P}}_t - \mathbf{K}_t \hat{\mathbf{C}}_t^T. \tag{4.6}$$

The Kalman gain represents the relative importance of the observation residual and is given by

$$\mathbf{K}_t = \hat{\mathbf{C}}_t \hat{\mathbf{\Sigma}}_t^{-1}, \tag{4.7}$$

where $\hat{\mathbf{C}}_t$ is the covariance matrix between the predicted state variable $\hat{\mathbf{z}}_t$ and the predicted observation values $\hat{\mathbf{x}}_t$. In this way, the BAM approximates the posterior distribution of the state variable $p(\mathbf{z}_t|\mathbf{X}_{1:t})$ with a normal distribution $N(\bar{\mathbf{z}}_t, \bar{\mathbf{P}}_t)$.

4.3.3 Challenges for BAM-based VN Reconfiguration

Ref. [46] examines the behavior of the BAM focusing on two-alternative forced choice tasks, which are most commonly employed when investigating perceptual decision-making. Specifically, Ref. [46] considers random dot motion (RDM) tasks in which subjects have to identify the direction that a randomly moving cloud of dots moves on average. Based on that analysis, it is clear that several problems need to be solved in order to apply the BAM-based approach to a VN reconfiguration method.

How to set the parameters r and q

The parameter r is the sensory uncertainty which represents the amount of noise on observations the decision maker expects. Thus, it is obvious that r should be the empirical standard deviation, s, of observations. In fact, Ref. [46] shows that the optimal Bayesian decision maker should have a generative model in which r is ideally equal to s.

The parameter q is the dynamical uncertainty which controls the propensity of the decision maker to change its decision and affects the balance between flexibility and stability in decision-making [46]. When q is small, the state valuable $\mathbf{z}$ is too stable to switch between fixed points. That is, it is difficult for the decision maker to change its decision even when the actual choice (i.e., the average of observation values) changes, since the decision maker explains away evidence for another choice as noise. When q is large, although the decision maker can change its decision rapidly, the decision maker sometimes changes its decision due to sensory noise. Therefore, it is necessary to set q to an appropriate value with which the decision maker can make fast and accurate decisions. By examining effects of the parameters r and q on the BAM-based approach in advance by off-line simulations, we can obtain appropriate parameter sets $\{r, q\}$.

In summary, we set r to the empirical standard deviation of observation values, and choose an appropriate value q corresponding to r obtained by off-line simulation of the BAM.

How to determine a criterion for decision making

We should use a criterion for decision-making that is suitable for a VN reconfiguration method. Ref. [65] introduces several definitions of confidence for decision-making: 1) the posterior belief itself, 2) the logarithm of the posterior belief, 3) the log change in the posterior belief. In our case, we use the third definition. This is because we use traffic information as observation values, and large fluctuations in traffic information make the posterior belief unstable, which makes decision-making difficult when we use the posterior belief itself as the confidence. However, a confidence measure based on a difference between posterior beliefs is more stable, which enables more stable decision-making. We give the detailed definition of the confidence in Section 4.4.2.

Calculation time of the BAM-based approach

The BAM-based approach should infer the posterior distribution of the state variable $p(\mathbf{z}_t|\mathbf{X}_{1:t})$ in a realistic calculation time. Ref. [46] only examines the behavior of the BAM with a two-dimensional state variable $\mathbf{z}_t$. However, since the BAM infers the posterior distribution using the UKF whose computational complexity is $O(D^3)$ [64], where D is the dimension of the state variable $\mathbf{z}_t$, the BAM-based approach operates with realistic calculation time even when D is large. Using an ordinary PC and our MATLAB implementation, we can actually estimate the posterior distribution within 4.0×10^{-3} sec even when D is 30, which means the calculation time is short enough. We discuss the effect of the dimension of the state variable (i.e., the number of choices) on our VN reconfiguration method in Section 4.5.4 in detail.

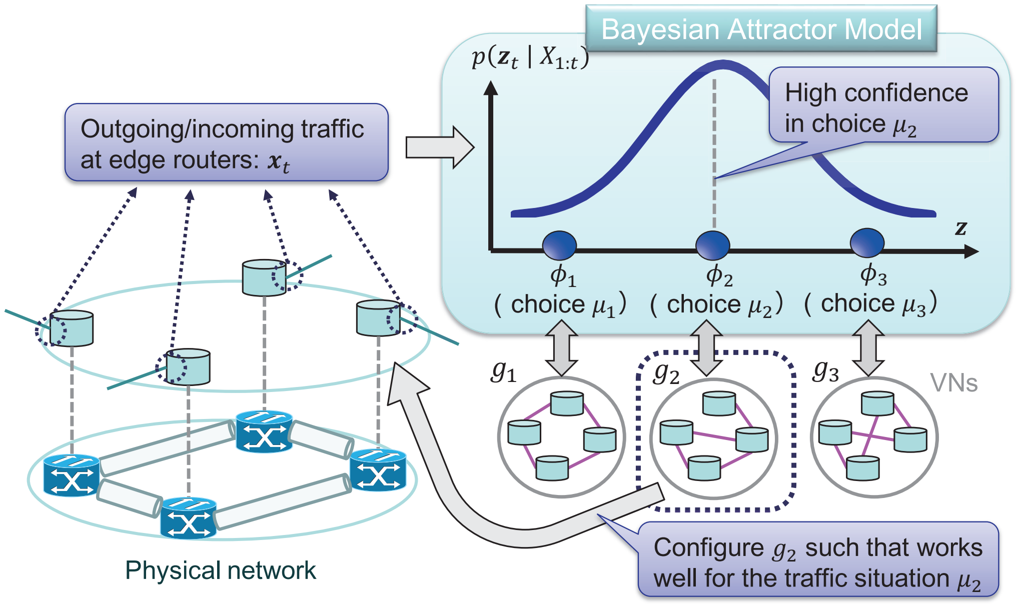

Bayesian Attractor Model
$p(\boldsymbol{z}_t \mid X_{1:t})$
High confidence in choice μ_2
$\boldsymbol{z}$
ϕ_1 (choice μ_1)
ϕ_2 (choice μ_2)
ϕ_3 (choice μ_3)
Outgoing/incoming traffic at edge routers: $\boldsymbol{x}_t$
g_1
g_2
g_3
VNs
Physical network
Configure g_2 such that works well for the traffic situation μ_2

Figure 4.3: Application of the Bayesian Attractor Model to VN reconfiguration

4.4 Virtual Network Reconfiguration Framework Based on the Bayesian Attractor Model

4.4.1 Overview of the VN Reconfiguration Framework

We have developed a BAM-based VN reconfiguration method. Fig. 4.3 briefly illustrates how to apply the BAM-based approach to the VN reconfiguration problem. In our method, when the traffic situation is identified as $\boldsymbol{\mu}_i$ given the observation values $\mathbf{X}_{1:t}$, the corresponding VN g_i suitable for the identified traffic situation $\boldsymbol{\mu}_i$ is retrieved and configured. More precisely, we select the VN g_i when the confidence in decision making for the choice $\boldsymbol{\mu}_i$ is sufficiently large. We use the amounts of incoming and outgoing traffic at edge routers as the observation values $\mathbf{X}_{1:t}$. Note that the VN g_i is prepared in advance such that the VN works well for the traffic situation $\boldsymbol{\mu}_i$.

Applying only the BAM-based approach is insufficient for a VN reconfiguration framework because the retrieved VN g_i may not be able to accommodate traffic demand even though the identification succeeds. To deal with this case, we incorporate the attractor selection-based method [41] into our VN reconfiguration framework in order to find good VNs. That is, we prepare a set of control phases and change the control phase based on both the confidence from the BAM-based approach and the service quality on the VN. More precisely, our VN reconfiguration framework is an on-line algorithm that reconfigures a VN by following the steps below, based on observation of the amounts of incoming and outgoing traffic at edge routers and the service quality on the VN (here, maximum link utilization on the VN).

(Step 1) Calculate the confidence from the BAM-based approach using the measured amounts of incoming and outgoing traffic at edge routers.

(Step 2) Change the control phase based on the confidence from the BAM-based approach and the service quality on the VN, and execute the control.

Note that the amounts of incoming and outgoing traffic at edge routers and the link utilization on the VN can be retrieved more easily than information of the end-to-end traffic demand matrix. The control phases are the following.

(Phase 1) Stay until a new traffic situation is identified.

- We do not reconfigure a VN until the current traffic situation is identified.

(Phase 2) Reconfigure the VN based on the identified traffic situation.

- We select the VN candidate g_i that works well for the identified traffic situation $\boldsymbol{\mu}_i$ (Phase 2-1).
- If VN g_i cannot accommodate the traffic demand, we search for a good VN using the attractor selection-based method [41] (Phase 2-2).

In summary, our VN reconfiguration framework first identifies traffic situations using the BAM-based approach (Phase 1) and immediately changes the VN after the identification succeeds (Phase 2-1). Then, we observe the service quality on the VN, and reconfigure the VN if necessary (Phase 2-2). Note that, this VN reconfiguration framework does not cover the case where the identification of traffic situations fails, that is, where the confidence is stable at a small value. We investigate ways to deal with the case where the identification fails in Section 4.6.

4.4.2 VN Reconfiguration Algorithm

We explain the details of the VN reconfiguration framework in the following.

Preparation

We prepare VNs g_i that work well for traffic situations $\boldsymbol{\mu}_i$ in advance. Examples of VN preparation are:

- We extract from a control history of VN configurations which show adequate performance in the specific traffic situations.
- We calculate VNs using traffic demand matrices that can be predicted from past traffic fluctuations.

We also prepare sets of the parameters $\{r, q\}$ of the BAM with which the traffic situation can be successfully identified by off-line simulations.

(Step 1) Calculate the confidence using the BAM-based approach

At a time t, we observe the amounts of incoming and outgoing traffic at edge routers, and calculate the confidence in decision making for the various choices.

First, we determine the parameters r and q used for the inference. Specifically, we calculate the empirical standard deviation s_t using the observations up to time t, $\mathbf{X}_{1:t}$, and set r to s_t. Then, we set q to the value corresponding to r obtained at the above preparatory phase. Here, we sequentially update the empirical standard deviation s_t using Welford's method [66] since it is not necessary to hold the past observation values from time 1 to time $t - 1$.

Second, we infer the posterior distribution of the state variable, $p(\mathbf{z}_t|\mathbf{X}_{1:t})$, and calculate the posterior belief for each choice, $p(\mathbf{z}_t = \phi_i|\mathbf{X}_{1:t})$.

Finally, we calculate the confidence for decision that the current traffic situation is identified as $\boldsymbol{\mu}_i$. Here, following Ref. [65], we use as the confidence the left side of Eq. (4.8), which represents the difference between logarithms of posterior beliefs. That is, we identify the current traffic situation as $\boldsymbol{\mu}_i$ when

$$\log_{10} \frac{p(\mathbf{z}_t = \phi_i|\mathbf{X}_{1:t})}{p(\mathbf{z}_t = \phi_j|\mathbf{X}_{1:t})} \geq \lambda, \tag{4.8}$$

where the posterior belief in the choice $\boldsymbol{\mu}_i$, $p(\mathbf{z}_t = \phi_i|\mathbf{X}_{1:t})$, is the largest among all the choices, and the posterior belief in the choice $\boldsymbol{\mu}_j$, $p(\mathbf{z}_t = \phi_j|\mathbf{X}_{1:t})$, is the second largest.

(Step 2) Change the control phase and execute the control

We change the control phase based on the confidence obtained at Step 1 and the service quality on the VN, and execute the control. The state transition diagram of the control phases is shown in Fig. 4.4. The label at each edge represents the transition condition, which consists of the confidence from the BAM-based approach and the service quality on the VN. The confidence becomes "stable at a large value" when Eq. (4.8) is satisfied for c consecutive times. The details of each control phase are the following.

(Phase 1) Stay until a new traffic situation is identified.

- This phase is the initial state of our VN reconfiguration framework. The control phase also changes to this phase when the confidence falls below the threshold λ.
- We do not reconfigure a VN in this phase.

(Phase 2-1) Configure the VN candidate that works well for the identified traffic situation.

- The control phase changes from Phase 1 to this phase when the confidence becomes stable at a large value.

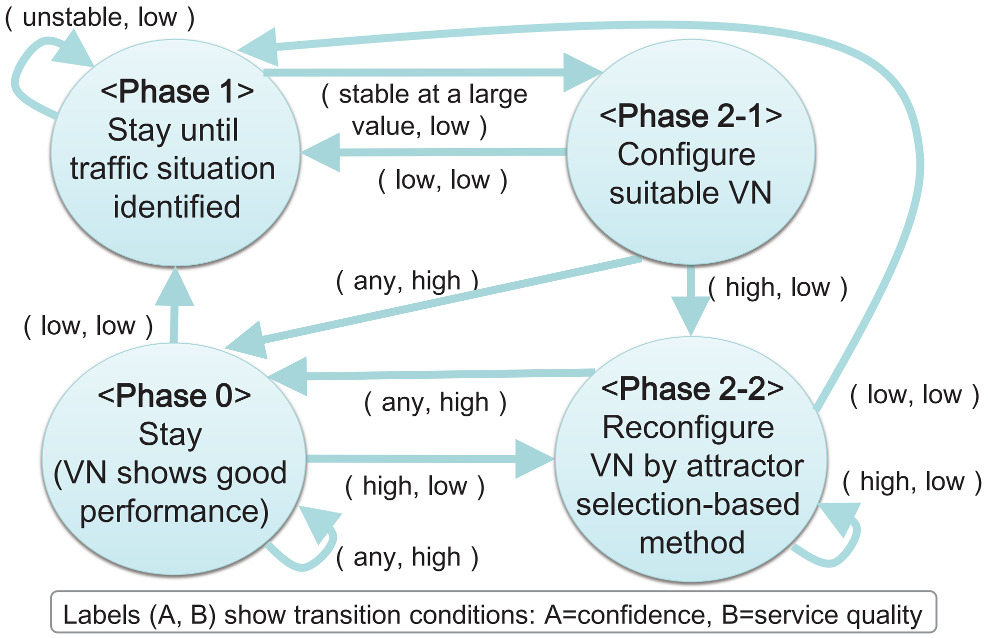

(unstable, low)
<Phase 1> Stay until traffic situation identified
(stable at a large value, low)
(low, low)
<Phase 2-1> Configure suitable VN
(any, high)
(high, low)
(low, low)
<Phase 0> Stay (VN shows good performance)
(any, high)
<Phase 2-2> Reconfigure VN by attractor selection-based method
(low, low)
(high, low)
(high, low)
(any, high)
Labels (A, B) show transition conditions: A=confidence, B=service quality

Figure 4.4: State transition diagram for the VN reconfiguration framework

- We configure one of VN candidates g_i that works well for the traffic situation μ_i at this phase.

(Phase 2-2) Reconfigure the VN using the attractor selection-based method.

- The control phase changes to this phase when the confidence takes a large value and the service quality on the VN is low.
- We search for a good VNs using the attractor selection-based method [41] in this phase.

(Phase 0) Stay. (The VN shows good performance.)

- The control phase changes to this phase when the service quality on the VN is improved.
- We do not reconfigure the VN in this phase.

Table 4.2: Parameters of the USNET physical network topology

Parameter	Value
Number of nodes	24
Number of links	43
Number of BVTs	10
Bandwidth of each BVT	100 Gbps

4.5 Evaluation of the VN Reconfiguration Framework

In this section, we first investigate the characteristics of our VN reconfiguration framework, and then evaluate the advantages of our VN reconfiguration framework over an elastic optical network in environments where traffic demand fluctuates. In such environments, it is expected that optimization approaches that use the end-to-end traffic demand matrix will have difficulty in reconfiguring a VN in response to traffic changes, as we mentioned in Section 4.1. We therefore evaluate the advantage of our VN reconfiguration framework by comparing it with an approach using the attractor selection-based method [41] alone since neither method uses traffic demand matrix information. Thus, we refer to the attractor selection-based method as the reference method.

4.5.1 Evaluation Environments

We use an elastic optical network that has the USNET topology. Table 4.2 shows the parameters of the physical network topology. The number of nodes, each of which consists of an IP router and a bandwidth-variable wavelength cross-connect (BV WXC), is 24, and the number of links (i.e., bidirectional optical fibers) is 43. Here, all the IP routers are edge routers. Each BV WXC has 10 bandwidth-variable transponders (BVTs) that can offer up to 100 Gbps of bandwidth. The goal of the control is to make the maximum link utilization on a VN less than 0.5. We generate traffic demand matrices $\mathbf{T}_1, \cdots, \mathbf{T}_5$ which follow a log-normal distribution. We denote the amounts of incoming and outgoing traffic at edge routers by $\mathbf{E}_1, \cdots, \mathbf{E}_5$, which are pre-specified traffic situations $\boldsymbol{\mu}_1, \cdots, \boldsymbol{\mu}_5$, when the traffic demand matrices are $\mathbf{T}_1, \cdots, \mathbf{T}_5$, and calculate the configurations of VN candidates

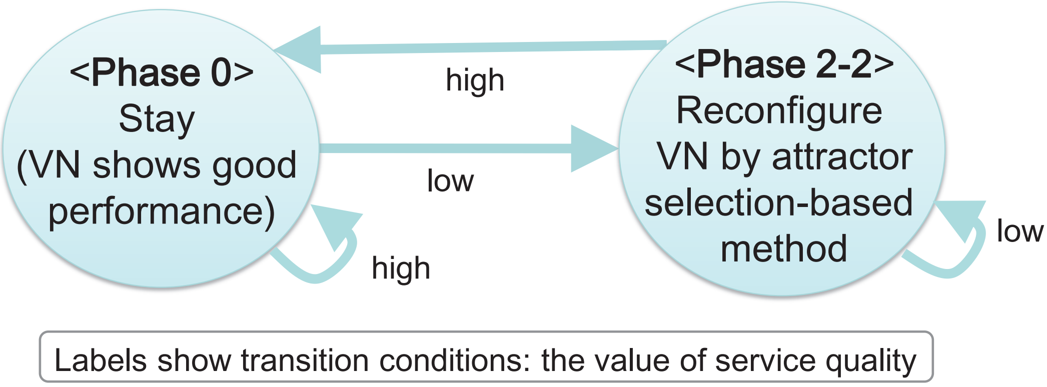

<Phase 0>
Stay
(VN shows good performance)
high
low
high
<Phase 2-2>
Reconfigure VN by attractor selection-based method
low
Labels show transition conditions: the value of service quality

Figure 4.5: State transition diagram for the attractor selection-based method

$g_1, \cdots, g_5$, which can accommodate $\mathbf{T}_1, \cdots, \mathbf{T}_5$. Specifically, we determine the virtual topology using the most subcarriers first (MSF) algorithm [57], and allocate frequency slots to lightpaths using the first-last fit algorithm [58]. We set c to 3, and set λ to 10.

For the evaluation, at every unit time step, the end-to-end traffic demand matrix is generated based on the normal distribution $N(\mathbf{T}_1, \Sigma)$ until time 100; then it is generated based on the normal distribution $N(\mathbf{T}_2, \Sigma)$ until time 200, where $\mathbf{T}_i = (T_{i,11}, \cdots, T_{i,NN})$ and $\Sigma = CV^2 \mathrm{diag}(T_{i,11}^2, \cdots, T_{i,NN}^2)$. N is the number of nodes, and CV is the coefficient of variation that represents the degree of traffic fluctuation. Refs. [67, 68] analyze real traffic data and fit the data to a traffic fluctuation model that follows a normal distribution. From these results, we find that the CV of real traffic is approximately within the range [0.5, 1.5]. The reference method changes its control phase based on the service quality on a VN, as shown in Fig. 4.5; the method searches for good VNs at Phase 2-2, then changes to Phase 0 when the performance of the VN gets improved.

4.5.2 Characteristics of the VN Reconfiguration Framework

Fig. 4.6 shows the transitions between control phases for each method when CV is 0.5. The figure clearly shows the behavior of our framework. Our method starts in Phase 1 and continues to stay in Phase 1 until the traffic situation is identified. At time 5, the

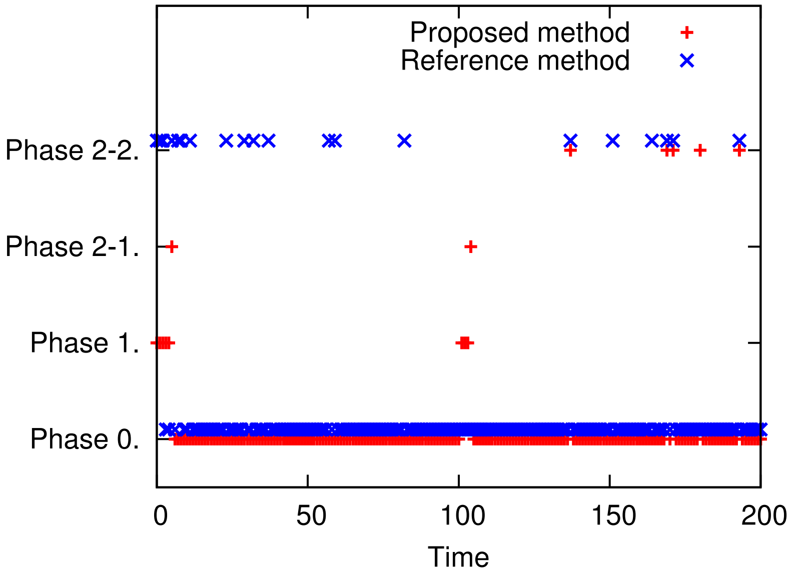


Figure 4.6: Transition of the control phase ($CV = 0.5$)

method shifts to Phase 2-1, and reconfigures a VN to the most promising one among the pre-prepared VN candidates. After that, the method enters Phase 0 since the VN can accommodate the traffic. Since the confidence in the identification of the current traffic situation becomes small at time 101, the method returns to Phase 1. At time 104, the method shifts to Phase 2-1 having detected the change of traffic situation. Note that our method can detect the change of traffic situation even though it accumulates past observations. After that, the method returns to Phase 0 since the VN can accommodate the traffic. Even when the method shifts to Phase 2-2 due to a traffic fluctuation (time 137), it is immediately returned to Phase 0 by the attractor selection-based method. This is because the VN is reconfigured to the most promising candidate and thereby the attractor selection-based method requires a little effort to find a good VN. In contrast, the reference method repeatedly changes control phase between Phase 2-2 and Phase 0. That is, although the reference method searches for a suitable VN for the current traffic situation and temporarily finds one, the VN cannot adapt to traffic changes after that.

4.5.3 Advantages of the VN Reconfiguration Framework

In this section, we evaluate the advantages of our VN reconfiguration framework. In operation, our VN reconfiguration framework aims to detect changes of traffic situation and configure a VN suitable for the current traffic situation. We therefore evaluate the advantages of our framework by considering performance after the traffic situation changes (i.e., after time 100). We executed 100 trials with different seeds to generate different traffic demand matrices.

First, we evaluate the required time for identification of the traffic situation. Fig. 4.7 shows the distribution of the required time for identification of the traffic situation. Here, we define the required time as the time from when the traffic situation changes (i.e., time 100) until the time when the confidence become stable at a large value and the control phase changes to Phase 2-1 for the first time. In Fig. 4.7, we can see that our VN reconfiguration framework can identify the traffic situation within 10 time steps in all the trials. Note that we confirmed that our framework had identified the current traffic situation as $\boldsymbol{\mu}_2$ in all trials. That is, our framework can correctly identify the traffic situation since the traffic demand matrix is generated by following $N(\mathbf{T}_2, \Sigma)$. We also find that the required time increases as the CV becomes larger. This is because it takes a longer time to accumulate evidence for the choice when the traffic demand fluctuates more largely and the observations are more likely to deviate from the pre-specified traffic situation.

Second, we evaluate the performance of the VN configured in Phase 2-1. Fig. 4.8 shows the distribution of the maximum link utilization of the VN configured in Phase 2-1 after the traffic situation changed (i.e., after time 100). We see that the VN configured in Phase 2-1 can, in most trials, achieve the goal of the control of making the maximum link utilization less than 0.5. However, the maximum link utilization of the VN tends to increase as the CV becomes larger. That is, although our VN reconfiguration framework can adapt to changes in traffic situations by configuring the most promising VN among the candidates in Phase 2-1 when the traffic fluctuation is small, our VN reconfiguration framework requires the search for a good VN in Phase 2-2 when the traffic fluctuation is large.

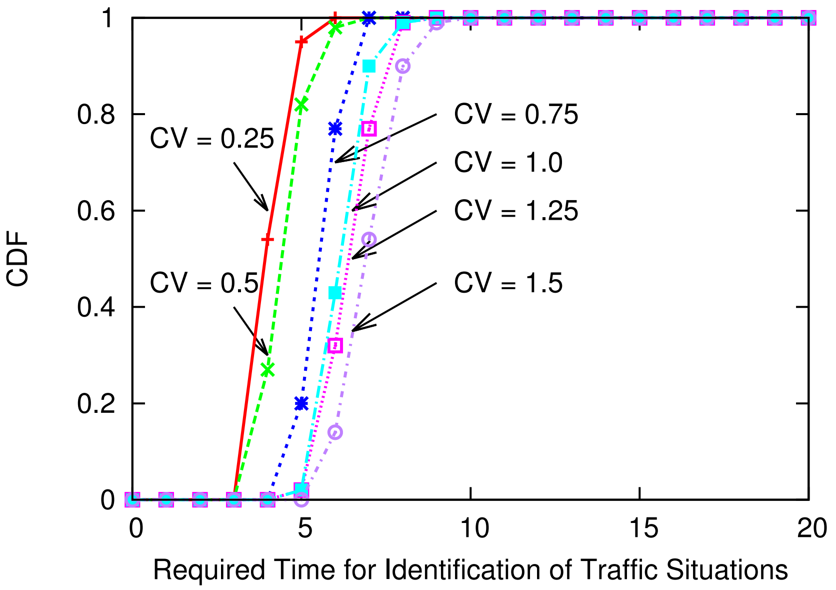


Figure 4.7: Cumulative distribution of the required time for identification of the traffic situation

Finally, we evaluate the stability of our VN reconfiguration framework. Fig. 4.9 shows the average of the total elapsed time spent in Phase 2-2 for each method after the traffic situation changed (i.e., after time 100). The horizontal axis shows the *CV* of the new traffic situation, and the vertical axis shows the average of the total elapsed time. In Fig. 4.9, we can see that the total elapsed time of our method is shorter than that of the reference method for all situations. That is, our method successfully decreases the number of VN reconfigurations needed to reach a VN suitable for the traffic situation. We also find that the total elapsed time of our method increases as *CV* becomes larger. This is because the VN configured in Phase 2-1 can adapt to traffic changes when the traffic fluctuation is small, and our VN reconfiguration framework requires the search of good VNs when the traffic fluctuation is large.

We also evaluated the advantages of our VN reconfiguration framework using the Simple CAIS Internet topology, which is used in Ref. [41], and obtained similar results to those shown in this section: that is, our VN reconfiguration framework can identify the traffic

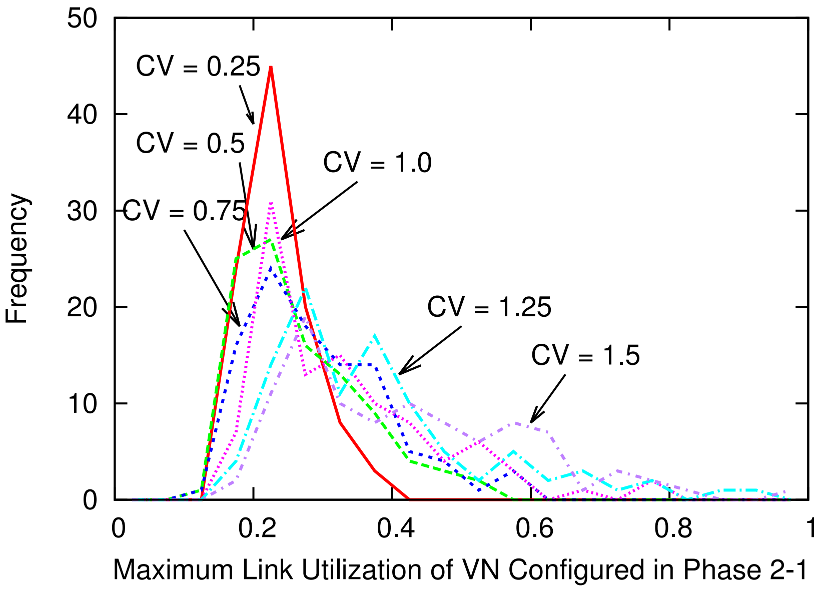


Figure 4.8: Distribution of the maximum link utilization of the VN configured in Phase 2-1

situation using the amounts of incoming and outgoing traffic at edge routers, and reduce the number of VN reconfigurations needed to reach a VN suitable for the traffic situation.

4.5.4 Effect of the Number of Choices

In this section, we discuss the effect of the number of choices (i.e., the number of pre-specified traffic situations). It is expected that our VN reconfiguration framework can adapt to more traffic situations when it can identify them. However, it is likely to be more difficult to distinguish between traffic situations as the number of them becomes larger. Thus, we evaluate the effect of the number of choices on our VN reconfiguration framework.

We denote the number of choices (i.e., the number of pre-specified traffic situations) by D, which is the parameter of evaluation in this section. That is, we generate traffic demand matrices $\mathbf{T}_1, \cdots, \mathbf{T}_D$ which follow a log-normal distribution. Then, we denote the amounts of incoming and outgoing traffic at edge routers by $\mathbf{E}_1, \cdots, \mathbf{E}_D$, which is the traffic situation $\boldsymbol{\mu}_1, \cdots, \boldsymbol{\mu}_D$, when the traffic demand matrix is $\mathbf{T}_1, \cdots, \mathbf{T}_D$, and calculate configurations of

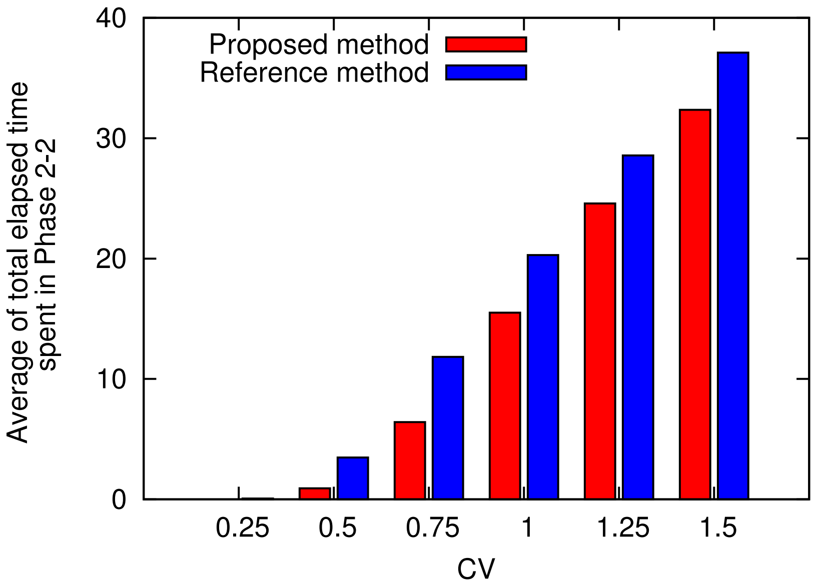


Figure 4.9: Average of the total elapsed time spent in Phase 2-2

the VN candidates $g_1, \cdots, g_D$ that can accommodate the traffic demands $\mathbf{T}_1, \cdots, \mathbf{T}_D$. We calculate the configuration of these VN candidates in the same way as in Section 4.5.1. For the evaluation, CV is 0.5, and the other parameters are similar to those in Section 4.5.1.

We consider the maximum time taken to identify the traffic situation for various numbers of choices. Fig. 4.10 shows the maximum time required to identify the traffic situation over 100 trials with different seeds used to generate different traffic demand matrices. We can see that it takes longer to identify traffic situations as their number becomes larger. Specifically, although the required time is about 10 time steps at most when the number of choices is 15 or less, the required time becomes longer when the number of choices becomes larger than 20. That is, there is a trade-off between the ability to identify more traffic situations and the ability to adapt to traffic changes in a shorter-time period. Thus, it is desirable to limit the number of pre-specified traffic situations to deploy our VN reconfiguration framework effectively in order to accommodate changing traffic demand; in this evaluation environment, the limit is 15 or less.

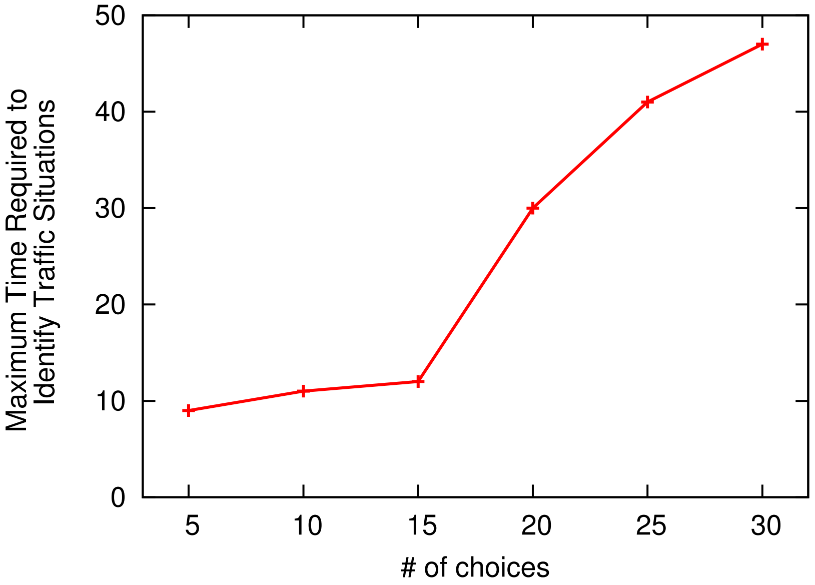


Figure 4.10: Maximum time required to identify the traffic situation

4.6 Virtual Network Reconfiguration Based on the Bayesian Attractor Model with Linear Regression

4.6.1 Overview of the Extended VN Reconfiguration Framework

We extend the VN reconfiguration framework in Section 4.4 to deal with the case where the identification of traffic situations fails. When the identification of traffic situations by the BAM-based method fails, we calculate and configure a new VN using linear regression [69] by following the steps below. Fig. 4.11 shows the outline of the VN reconfiguration method based on the BAM with linear regression.

Step 1 Fit the current traffic situation, denoted by $\boldsymbol{\mu}_{new}$, by linear regression using a set of pre-specified traffic situations $\boldsymbol{\mu}_1, \cdots, \boldsymbol{\mu}_D$.

Step 2 Calculate and configure a new VN, denoted by g_{new}, using the obtained regression coefficients $w_1, \cdots, w_D$.

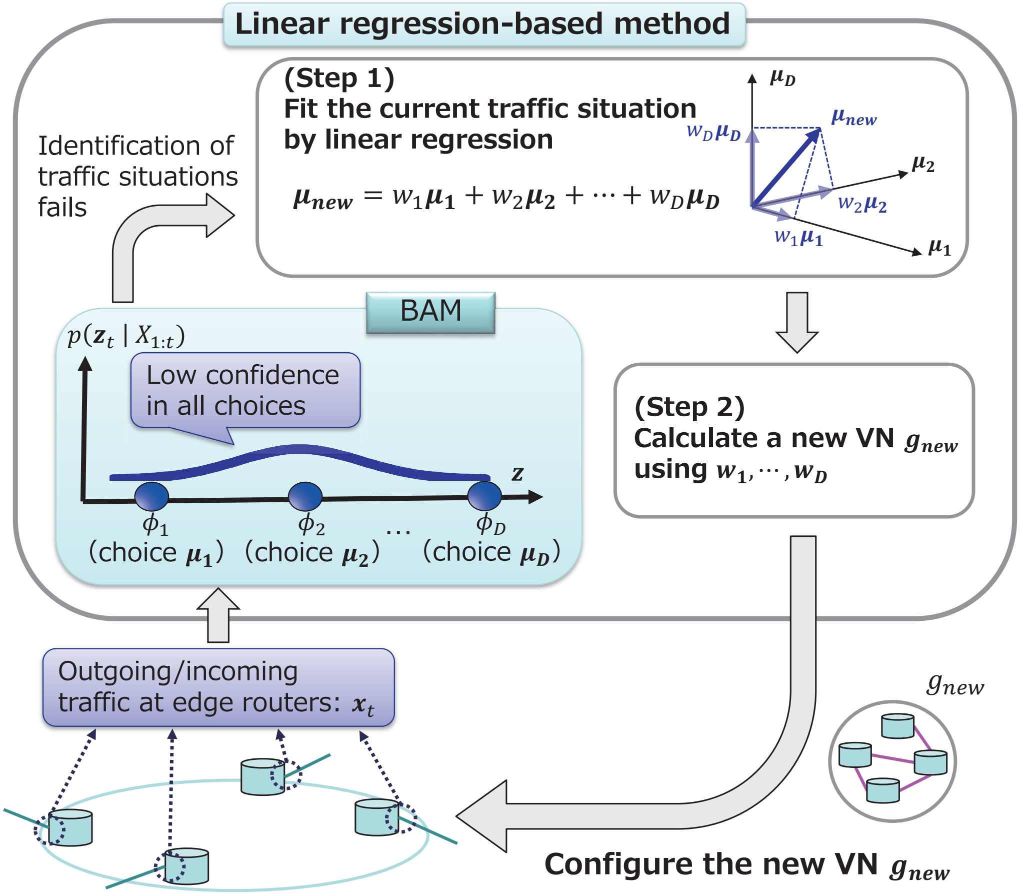

Linear regression-based method
(Step 1)
Fit the current traffic situation by linear regression
$\mu_{new} = w_1\mu_1 + w_2\mu_2 + \cdots + w_D\mu_D$
μ_D
μ_{new}
$w_D\mu_D$
μ_2
$w_2\mu_2$
$w_1\mu_1$
μ_1
Identification of traffic situations fails
BAM
$p(z_t \mid X_{1:t})$
Low confidence in all choices
z
ϕ_1
ϕ_2
ϕ_D
(choice μ_1)
(choice μ_2)
(choice μ_D)
(Step 2)
Calculate a new VN g_{new} using $w_1, \cdots, w_D$
Outgoing/incoming traffic at edge routers: x_t
g_{new}
Configure the new VN g_{new}

Figure 4.11: Outline of the VN reconfiguration based on the BAM with linear regression

Applying only the liner regression-based approach is not sufficient when the identification of traffic situations fails because the VN g_{new} may not be able to accommodate traffic demand. In this case, we apply the attractor selection-based VN reconfiguration method [41] to find a good VN. However, by introducing the linear regression-based method, it is expected that the number of VN reconfiguration to accommodate traffic demand can be reduced even when the identification of traffic situations fails, since we calculate the VN g_{new} that is suitable to some extent using the information obtained by fitting the current traffic situation. Note that, since this linear regression-based method utilizes only the information used in the above VN reconfiguration framework, such as observed amounts of incoming and outgoing traffic at edge routers and a set of pre-specified traffic situations $\boldsymbol{\mu}_1, \cdots, \boldsymbol{\mu}_D$, there is no information to be additionally observed to configure the VN g_{new}. We incorporate this linear regression-based method into the VN reconfiguration framework proposed in Section 4.4.

4.6.2 VN Reconfiguration Algorithm with Linear Regression

We explain the details of the linear regression-based VN reconfiguration algorithm.

(Step 1) Fit the traffic situation by linear regression

The BAM as a state-space representation has the output equation Eq. (4.3), where the observation value (observed amounts of incoming and outgoing traffic at edge routers in our VN reconfiguration framework) is represented by a linear sum of pre-specified traffic situations. We apply this equation to the approach for dealing with the case where the identification of traffic situations fails. That is, we fit the current traffic situation $\boldsymbol{\mu}_{new}$ by linear regression using a set of pre-specified traffic situations $\boldsymbol{\mu}_1, \cdots, \boldsymbol{\mu}_D$ to satisfy the equation

$$\boldsymbol{\mu}_{new} = w_1\boldsymbol{\mu}_1 + \cdots + w_D\boldsymbol{\mu}_D + \boldsymbol{\epsilon} = \sum_{i=1}^{D} w_i\boldsymbol{\mu}_i + \boldsymbol{\epsilon}, \tag{4.9}$$

where $w_1, \cdots, w_D$ are regression coefficients and $\boldsymbol{\epsilon}$ represents the error term. More precisely, we calculate regression coefficients $w_1, \cdots, w_D$ so that the residual sum of squares (RSS) defined by the equation,

$$\begin{aligned} RSS(w_i) \qquad &= \epsilon^T\epsilon = (\boldsymbol{\mu}_{new} - \hat{\boldsymbol{\mu}}_{new})^T(\boldsymbol{\mu}_{new} - \hat{\boldsymbol{\mu}}_{new}) \\ &= \left(\boldsymbol{\mu}_{new} - \textstyle\sum_{i=1}^{D} w_i\boldsymbol{\mu}_i\right)^T \left(\boldsymbol{\mu}_{new} - \textstyle\sum_{i=1}^{D} w_i\boldsymbol{\mu}_i\right), \end{aligned} \tag{4.10}$$

is minimized by the least squares method. In the BAM-based approach, the coefficient of each traffic situation $\boldsymbol{\mu}_i$, which is denoted by $w_i' = \sigma(z_i)$, is defined to satisfy $0 \leq w_i' \leq 1$. However, there are no constraints on the coefficient w_i when we fit the current traffic situation $\boldsymbol{\mu}_{new}$ by linear regression. Even when the traffic volume increases and thereby the coefficient w_i is greater than 1, it is not a problem in calculating a new VN g_{new} since we can consider that the traffic pattern, which is the relationship among the traffic volume at every edge routers, dose not change. Note that, in this study, we do not cover the case where the traffic volume increases so that the enhancement of the physical network equipment becomes necessary for accommodating traffic demand.

(Step 2) Calculate a new VN

Utilizing the obtained regression coefficients $w_1, \cdots, w_D$, we calculate a new VN g_{new} and configure it. In our algorithm, since the current traffic situation $\boldsymbol{\mu}_{new}$ is fitted by a linear sum of pre-specified traffic situations $\boldsymbol{\mu}_1, \cdots, \boldsymbol{\mu}_D$, we represents the current traffic demand matrix $\mathbf{T}_{new}$ as a linear sum of the corresponding traffic demand matrices $\mathbf{T}_1, \cdots, \mathbf{T}_D$, as shown in this equation

$$\mathbf{T}_{new} = w_1\mathbf{T}_1 + \cdots + w_D\mathbf{T}_D = \sum_{i=1}^{D} w_i\mathbf{T}_i. \tag{4.11}$$

We can use the information of the traffic demand matrices $\mathbf{T}_1, \cdots, \mathbf{T}_D$ because we retain this information to calculate VN candidates in the BAM-based method. Then, we calculate a new VN g_{new} by the heuristic algorithms [57,58] using the obtained traffic demand matrix $\mathbf{T}_{new}$ as an input.

When g_{new} cannot accommodate traffic demand, we apply the attractor selection-based VN reconfiguration method [41] to find a good VN. However, by introducing the linear regression-based method, it is expected that the number of VN reconfiguration to accommodate traffic demand can be reduced even when the identification of traffic situations fails, since we calculate the VN g_{new} that is suitable to some extent using the information obtained by fitting the current traffic situation.

4.7 Evaluation of the Extended VN Reconfiguration Framework

In this section, we evaluate the effectiveness of configuring a new VN g_{new} calculated by the linear regression-based method when the identification of traffic situations fails.

4.7.1 Evaluation Environments

The parameters of the physical network topology, the goal of the control, and the retained information by the BAM-based method, such as the pre-specified traffic demand matrices $\mathbf{T}_1, \cdots, \mathbf{T}_5$, and the corresponding traffic situations $\boldsymbol{\mu}_1, \cdots, \boldsymbol{\mu}_5$, and the VN candidates $g_1, \cdots, g_5$, are the same as shown in Section 4.5.1.

We generate 1,000 patterns of traffic demand matrix information $\mathbf{T}'$ with different seeds assuming unknown traffic fluctuations. Note that, the generated traffic demand matrices $\mathbf{T}'$ are different from the retained ones $\mathbf{T}_1, \cdots, \mathbf{T}_5$. For the evaluation, at every unit time step, the end-to-end traffic demand matrix is generated based on the normal distribution $N(\mathbf{T}_1, \Sigma)$ until time 50; then a traffic fluctuation occurs and the traffic demand matrix is generated based on the normal distribution $N(\mathbf{T}', \Sigma)$ until time 100, where $\mathbf{T}_i = (T_{i,11}, \cdots, T_{i,NN})$ and $\Sigma = 0.5^2 \mathrm{diag}(T_{i,11}^2, \cdots, T_{i,NN}^2)$.

4.7.2 Evaluation Results

We have evaluated the effectiveness of the linear regression-based method when the identification of traffic situations fails. Fig. 4.12 shows the breakdown of the simulation results, i.e., whether the identification of traffic situations succeeds or not, and whether the configured VN can accommodate traffic demand or not for 1,000 trials using different $\mathbf{T}'$. In Fig. 4.12, we can see that the identification of traffic situations succeeds in 30.9(=12.5+18.4) % of trials, and fails in 69.1 % of trials. To investigate the effectiveness of the linear regression-based method when the identification of traffic situations fails, we focus on the latter 69.1 % of trials below. In Fig. 4.12, we find that we can accommodate traffic demand in 65.7 % of trials when the identification of traffic situations fails, whereas we cannot accommodate

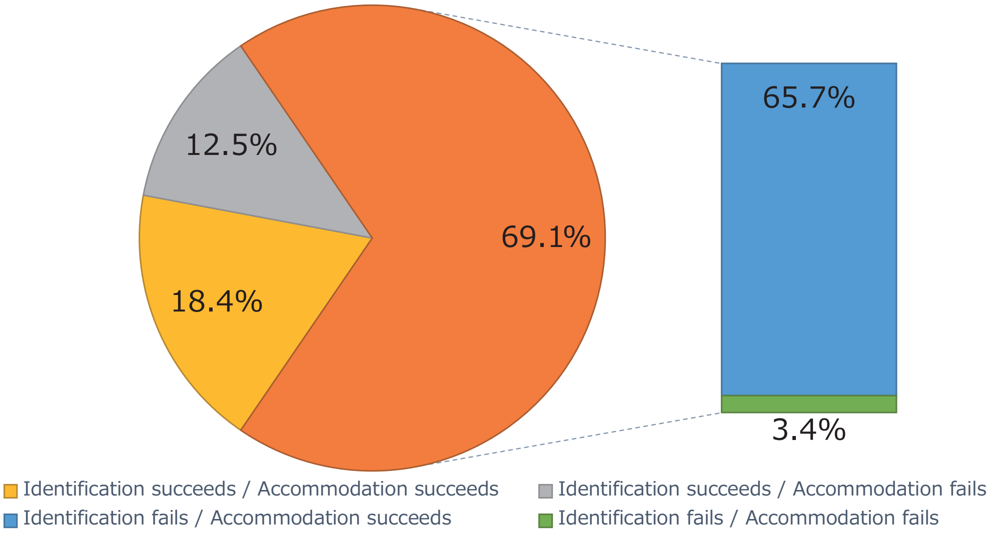


Figure 4.12: Breakdown of the simulation results

traffic demand in 3.4 % of trials. Thus, the linear regression-based method can configure a suitable VN in most cases when the identification of traffic situation fails. When we apply only the BAM-based method, we need to use the attractor selection-based method in 81.6(=100-18.4) % of trials. By incorporating the linear regression-based method into our VN reconfiguration framework, we need the attractor selection-based method only in 15.9(=12.5+3.4) % of trials. This indicates that we can effectively reduce VN reconfiguration necessary to reach a suitable VN for the current traffic situation.

4.8 Guideline to Select and Update a Set of Pre-specified Traffic Situations

4.8.1 Approach for Selecting a Set of Pre-specified Traffic Situations

To effectively deploy our linear regression-based method, it is important to appropriately select and update a set of pre-specified traffic situations. It is expected that we can obtain better VNs when the linear sum of pre-specified traffic situations expresses wider variety of traffic situations and makes the residuals smaller. Therefore, it is one of the guidelines to select D traffic situations $\boldsymbol{\mu}_1, \cdots, \boldsymbol{\mu}_D$ so that the liner sum of these traffic situations improve the ability to expresses various traffic situations. One way to improve the ability to express various traffic situations is to select a set of pre-specified traffic situations so that they have linear independence when we consider each traffic situation as a vector. This is because the dimension of the space spanned by the pre-specified traffic situation vectors becomes maximum when the set of selected traffic situation vectors has linear independence. Whether the set of selected traffic situation vectors has linear independence or not can be easily judged by checking whether the rank of the matrix $M = [\boldsymbol{\mu}_1, \cdots, \boldsymbol{\mu}_D]$ matches D or not.

Here, we evaluate the relationship between the ability to express various traffic situations and the performance of the linear regression-based VN reconfiguration method. Although the parameters for the evaluation are the same as shown in Section 4.7.1, we use 3 sets of pre-specified traffic situations below.

- 1st set: $\{\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \boldsymbol{\mu}_3, \boldsymbol{\mu}_4, \boldsymbol{\mu}_5\}$
- 2nd set: $\{\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \boldsymbol{\mu}_3, \boldsymbol{\mu}_4, (\boldsymbol{\mu}_1 + \boldsymbol{\mu}_2)/2\}$
- 3rd set: $\{\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \boldsymbol{\mu}_3, (\boldsymbol{\mu}_1 + \boldsymbol{\mu}_2)/2, (\boldsymbol{\mu}_2 + \boldsymbol{\mu}_3)/2\}$

The number of pre-specified traffic situation of each set is 5. The rank of the matrix M with each traffic situation as a column is 5 for the first set, 4 for the second set, 3 for the third set. That is, the first set has linear independence, the second and the third set is linear dependent. The first set is the same as used for the evaluation in Section 4.7.

Fig. 4.13 shows the ratio of trials where the identification of traffic situations fails and the configured VN by the linear regression-based method cannot accommodate traffic demand out of 1,000 trials, which we call failure rate, when using each set of pre-specified traffic situations. In Fig. 4.13, we can see that the failure rate is reduced by using the first set of traffic situations. Moreover, we can find that the failure rate is smaller in the case of using the second set with rank 4 than in the case of using the third set with rank 3.

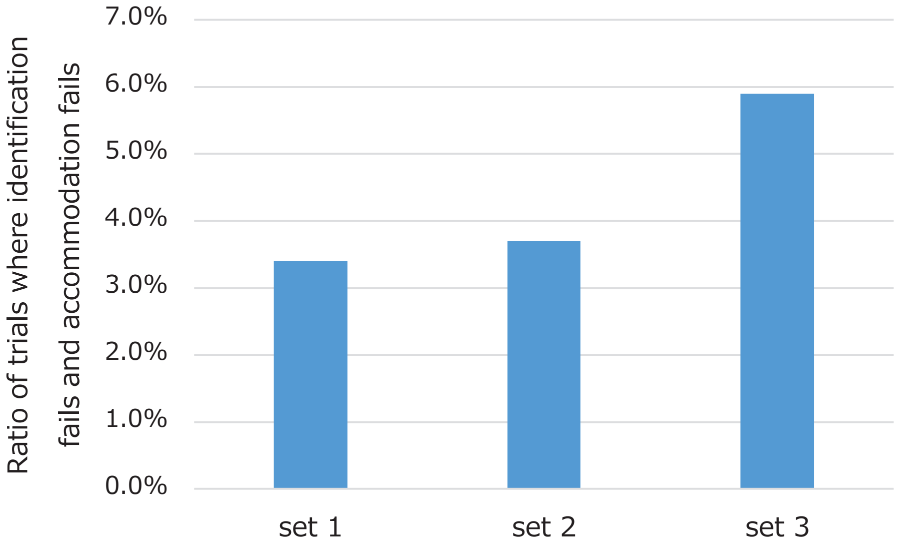


Figure 4.13: Effects of linear independence of a set of pre-specified traffic situations

In other words, the failure rate decreases as the rank of the matrix M with each selected traffic situation as a column becomes larger.

From the above evaluation, it is effective to maximize the rank of the matrix M with each selected traffic situation as a column, that is, it is effective to select a set of pre-specified traffic situations so that is has linear independence.

4.8.2 Approach for Updating a Set of Pre-specified Traffic Situations

We believe that it is effective to dynamically update a set of pre-specified traffic situations in order to accommodate changing traffic demand for the future by applying our VN reconfiguration framework. When the identification of traffic situations fails and our linear regression-based method can configure a VN suitable for the current traffic situation, it is not necessary to update the set of pre-specified traffic situations since our VN reconfiguration framework can deal with the traffic fluctuation using the retained traffic situations. In the case where the VN configured by the linear regression-based method cannot accommodate traffic demand, we apply the attractor selection-based method to find a good VN.

When a solution (i.e., a VN that can accommodate traffic demand) is found, we add the traffic situation at that time to the set of pre-specified traffic situations. However, since it takes a longer time to identify the traffic situations as the number of pre-specified traffic situations becomes larger, as shown in Section 4.5.4, it is not desirable the number of pre-specified traffic situations is too large. Therefore, it is necessary to delete some of pre-specified traffic situations from the set properly. We believe that it is sufficient to delete some of traffic situations from the set with a longer time period than to add new traffic situations. We therefore can use information of the current traffic demand matrix $\mathbf{T}_{now}$ when we delete some of traffic situations from the set. We denote the corresponding traffic situation bv $\boldsymbol{\mu}_{now}$ when the traffic demand matrix is $\mathbf{T}_{now}$. One guideline is to delete some of traffic situations from the set that can be represented by a linear sum of the pre-specified traffic situations $\boldsymbol{\mu}_1, \cdots, \boldsymbol{\mu}_D, \boldsymbol{\mu}_{now}$.

4.9 Conclusion

We have developed a virtual network reconfiguration framework based on the Bayesian Attractor Model (BAM), which does not use the end-to-end traffic demand matrix. Our VN reconfiguration framework first identifies the current traffic situation to the closest one among several pre-specified traffic situations using the BAM, and immediately changes the VN to the most suitable candidate for the identified traffic situation. Then, the framework observes the service quality on the VN and reconfigures the VN if necessary by the attractor selection-based method. Evaluation results showed that our framework can identify the traffic situation using the amounts of incoming and outgoing traffic at edge routers, which are more easily obtained than the traffic demand matrix, and reduces the number of VN reconfigurations needed to reach a VN suitable for the actual traffic situation.

Furthermore, we extended the above VN reconfiguration framework to deal with the case where the identification of traffic situations fails. When the identification fails, the extended framework fits the current traffic situation by linear regression using a set of pre-specified traffic situations, and calculates and configures a new VN using the obtained regression coefficients. Then, the extended framework observes the service quality on the VN and reconfigures the VN if necessary by the attractor selection-based method. Evaluation results showed that the linear regression-based method can configure a suitable VN in most cases when failing to identify the current traffic situation, which leads to reduction of the number of VN reconfiguration. We also investigate how to select and update a set of pre-specified traffic situations, and found that it is effective to select a set of pre-specified traffic situations so that the set has linear independence when we consider each of traffic situation as a vector.

In the future we will investigate how to select and update a set of pre-specified traffic situations that optimizes the performance of VNs configured by the extended VN reconfiguration framework. By updating the pre-specified traffic situations, it is expected that we can configure VNs that are suitable for more traffic situations and require fewer reconfigurations.

Chapter 5

Conclusion

The emergence of new network services and applications with wide range of required bandwidth has caused large traffic fluctuations. One approach for the network operator to accommodate traffic demand on an optical network is to construct a virtual network (VN) and reconfigure a VN following traffic changes. Our research group has previously proposed a VN reconfiguration approach based on attractor selection, which models the behavior where living organisms adapt to unknown changes in their surrounding environment and recover their condition, to adapt to traffic fluctuations. In this thesis, following the attractor selection-based VN reconfiguration approach, we proposed an attractor-based VN reconfiguration framework that quickly adapts to various traffic fluctuations with fewer VN reconfigurations.

First, we proposed a design method of attractors (i.e., VN candidates) in the attractor selection-based VN reconfiguration approach. In the attractor selection-based approach, since a VN is reconfigured guided by attractors, that is, a VN is reconfigured to have a network topology close to one of the VN candidates, it is crucial to design the attractors properly. The proposed method designs attractors with a wide variety characteristics so that the attractor selection-based VN reconfiguration can adapt to various traffic fluctuations. Our basic design approach is to prepare VN candidates which the bottleneck links (lightpaths) are different from each other. However, our exhaustive algorithm based on this approach has a problem of requiring large amounts of computational time for large-scale networks that have more than 10 nodes. To solve this problem, we also proposed a method that hierarchically contracts a network topology so that our algorithm can be applied to large-scale networks. Evaluation results showed that our method can design VN candidates that achieve better service quality on a VN than the randomly generated VN candidates, even when targeting for a 1000-node network. As a result, the VN reconfiguration using attractors obtained by our design method finds a solution in a shorter time against various traffic fluctuations.

Second, we proposed an attractor selection-based VN reconfiguration method for elastic optical path networks. Elastic optical path networks have been shown to be a promising candidate for future resource-efficient optical networks. Although elastic optical path networks can achieve higher utilization efficiency of spectrum resources by dividing spectrum resources into narrower frequency slots and providing the sufficient bandwidth, it is essential to tackle the problem of allocating spectrum resources under the spectrum contiguity and continuity constraints. We newly defined the potential bandwidth as a metric that reflects the bandwidth that can be additionally offered under the spectrum contiguity and continuity constraints. Then, our method reconfigures a VN based on attractor selection so that both the service quality on the VN and the potential bandwidth get improved. In addition, our method adjusts the bandwidth according to the link utilization of lightpaths that form the VN to provide the required bandwidth. Evaluation results showed that the proposed method can set aside about 50 % of resources for future use, while improving the service quality on a VN to the same extent as the existing heuristic method, considering the spectrum contiguity and continuity constraints.

By proposing the above methods, we established an attractor selection-based VN reconfiguration method for optical networks that quickly adapts to various traffic fluctuations. However, since the attractor selection-based VN reconfiguration approach gradually reconfigures a VN in the process of search for a solution (i.e., a VN that can accommodate traffic demand), simply applying the attractor selection-based approach may over-reconfigure a VN in nature, which disrupts network services accommodated on the VN. Thus, to reduce the number of VN reconfigurations, we introduced a cognitive mechanism that perceives current traffic situation and adapts to the situation. Specifically, we proposed another VN reconfiguration method based on the Bayesian Attractor Model (BAM), which models the human behavior of making appropriate decisions by recognizing the surrounding situation. The key idea of this method is to memorize a set of VN candidates, each of which works well for a pre-specified traffic situation, and then retrieve a suitable VN for the current traffic situation from this set. We used certain patterns of incoming and outgoing traffic at edge routers as the traffic situation, since this information can be obtained more easily than the traffic demand matrix. By identifying the current traffic situation using the BAM, this method retrieves the most promising VN. However, for the case where the retrieved VN cannot accommodate traffic demand, we applied the attractor selection-based VN reconfiguration method. That is, we established a VN reconfiguration framework that deals with known traffic situations by the BAM-based method and deals with unknown traffic situations by the attractor selection-based method. Evaluation results showed that the BAM-based method can identify the current traffic situation by observing the amounts of incoming and outgoing traffic at edge routers. As a result, our VN reconfiguration framework can reach a VN suitable for the current traffic situation with fewer VN reconfigurations.

Finally, we extended the above VN reconfiguration framework to deal with the case where the identification of the current traffic situation fails. When the identification fails, the BAM-based method cannot configure a promising VN. We therefore proposed a method that configures a promising VN when the identification fails, and incorporate this method into the above VN reconfiguration framework. Our method utilizes a set of pre-specified traffic situations, and the current traffic situation is fitted by linear regression when the identification fails. Then, our method configures a VN using the obtained regression coefficients. Evaluation results showed that the linear regression-based method can configure a suitable VN in most cases when failing to identify the current traffic situation. We also investigated how to select and update the set of pre-specified traffic situations, and found that it is effective to select a set of pre-specified traffic situations to have linear independence.

In summary, we proposed an attractor-based VN reconfiguration framework for optical networks in this thesis. The VN reconfiguration framework observes the amounts of incoming and outgoing traffic at edge routers and identifies the current traffic situation using the BAM-based method. When the identification succeeds, the BAM-based method retrieves and configures the most promising VN from a set of VN candidates. When the identification fails, the linear regression-based method fits the current traffic situation by linear regression and configures a promising VN using the obtained regression coefficients. In the case where the configured VN is not suitable for the current traffic situation, the attractor selection-based method searches for a solution. In the process of search of a solution, the VN is reconfigured so that both service quality on the VN and the potential bandwidth get improved, guided by the attractors obtained by the design method. In this way, the attractor-based VN reconfiguration framework quickly adapts to various traffic situations with fewer VN reconfigurations. One of our future research topics is to prove that our VN reconfiguration framework also adapts to network failures. Another future direction is to find an approach to reconfiguring multiple VNs, each of which is assigned to one network service and coordinates with other VNs using a BAM-based mechanism.

We believe that the above discussion in this thesis will contribute to the management of future optical networks.

Bibliography

[1] Cisco, Visual Network Index, “Forecast and Methodology, 2015-2020,” June 2016.

[2] A. Banerjee, Y. Park, F. Clarke, H. Song, S. Yang, G. Kramer, K. Kim, and B. Mukherjee, “Wavelength-division-multiplexed passive optical network (WDM-PON) technologies for broadband access: a review,” *Journal of optical networking*, vol. 4, pp. 737–758, Nov. 2005.

[3] G. Zhang, M. De Leenheer, A. Morea, and B. Mukherjee, “A survey on OFDM-based elastic core optical networking,” *IEEE Communications Surveys & Tutorials*, vol. 15, pp. 65–87, Feb. 2013.

[4] O. Gerstel, M. Jinno, A. Lord, and S. B. Yoo, “Elastic optical networking: A new dawn for the optical layer?,” *IEEE Communications Magazine*, vol. 50, pp. s12–s20, Feb. 2012.

[5] M. Jinno, H. Takara, B. Kozicki, Y. Tsukishima, Y. Sone, and S. Matsuoka, “Spectrum-efficient and scalable elastic optical path network: architecture, benefits, and enabling technologies,” *IEEE Communications Magazine*, vol. 47, pp. 66–73, Nov. 2009.

[6] N. M. K. Chowdhury and R. Boutaba, “Network virtualization: state of the art and research challenges,” *IEEE Communications magazine*, vol. 47, pp. 20–26, July 2009.

[7] N. M. K. Chowdhury and R. Boutaba, “A survey of network virtualization,” *Computer Networks*, vol. 54, pp. 862–876, Apr. 2010.

[8] A. Fischer, J. F. Botero, M. T. Beck, H. De Meer, and X. Hesselbach, "Virtual network embedding: A survey," *IEEE Communications Surveys & Tutorials*, vol. 15, pp. 1888–1906, Feb. 2013.

[9] D. Kreutz, F. M. Ramos, P. E. Verissimo, C. E. Rothenberg, S. Azodolmolky, and S. Uhlig, "Software-defined networking: A comprehensive survey," *Proceedings of the IEEE*, vol. 103, pp. 14–76, Jan. 2015.

[10] R. Dutta and G. N. Rouskas, "A survey of virtual topology design algorithms for wavelength routed optical networks," *Optical Networks*, vol. 1, pp. 73–89, May 1999.

[11] B. Ramamurthy and A. Ramakrishnan, "Virtual topology reconfiguration of wavelength-routed optical WDM networks," in *Proceedings of IEEE GLOBECOM*, vol. 2, pp. 1269–1275, Dec. 2000.

[12] A. Gençata and B. Mukherjee, "Virtual-topology adaptation for WDM mesh networks under dynamic traffic," *IEEE/ACM Transactions on Networking (TON)*, vol. 11, pp. 236–247, Apr. 2003.

[13] F. Ricciato, S. Salsano, A. Belmonte, and M. Listanti, "Off-line configuration of a MPLS over WDM network under time-varying offered traffic," in *Proceedings of IEEE INFOCOM*, vol. 1, pp. 57–65, June 2002.

[14] S. Gieselman, N. Singhal, and B. Mukherjee, "Minimum-cost virtual-topology adaptation for optical WDM mesh networks," in *Proceedings of International Conference on Communications*, vol. 3, pp. 1787–1791, June 2005.

[15] G. Agrawal and D. Medhi, "Lightpath topology configuration for wavelength-routed IP/MPLS networks for time-dependent traffic," in *Proceedings of IEEE GLOBECOM*, pp. 1–5, Nov. 2006.

[16] D. Banerjee and B. Mukherjee, "Wavelength-routed optical networks: Linear formulation, resource budgeting tradeoffs, and a reconfiguration study," *IEEE/ACM Transactions on Networking (TON)*, vol. 8, pp. 598–607, Oct. 2000.

[17] W. Wei, C. Wang, and X. Liu, “Adaptive IP/optical OFDM networking design,” in *Proceedings of Optical Fiber Communication Conference*, p. OWR6, Mar. 2010.

[18] T. Tanaka, A. Hirano, and M. Jinno, “Advantages of IP over elastic optical networks using multi-flow transponders from cost and equipment count aspects,” *Optics express*, vol. 22, pp. 62–70, Jan. 2014.

[19] Hong, Sangjin and Jue, Jason P and Zhang, Qiong and Wang, Xi and Cankaya, Hakki C and She, Qingya and Xie, Weisheng and Sekiya, Motoyoshi, “Virtual optical network provisioning over flexible-grid multi-domain optical networks,” in *Proceedings of IEEE GLOBECOM*, pp. 1–6, Dec. 2015.

[20] V. Gkamas, K. Christodoulopoulos, and E. Varvarigos, “A joint multi-layer planning algorithm for IP over flexible optical networks,” *IEEE/OSA Journal of Lightwave Technology*, vol. 33, pp. 2965–2977, July 2015.

[21] J. Zhang, Y. Zhao, X. Yu, J. Zhang, M. Song, Y. Ji, and B. Mukherjee, “Energy-efficient traffic grooming in sliceable-transponder-equipped IP-over-elastic optical networks [invited],” *IEEE/OSA Journal of Optical Communications and Networking*, vol. 7, pp. A142–A152, Jan. 2015.

[22] Assis, KDR and Peng, S and Almeida, RC and Waldman, H and Hammad, A and Santos, AF and Simeonidou, D, “Network virtualization over elastic optical networks with different protection schemes,” *IEEE/OSA Journal of Optical Communications and Networking*, vol. 8, pp. 272–281, Apr. 2016.

[23] Y. Ohsita, T. Miyamura, S. Arakawa, S. Ata, E. Oki, K. Shiomoto, and M. Murata, “Gradually reconfiguring virtual network topologies based on estimated traffic matrices,” *IEEE/ACM Transactions on Networking*, vol. 18, pp. 177–189, Feb. 2010.

[24] N. Fernández, R. J. D. Barroso, D. Siracusa, A. Francescon, I. de Miguel, E. Salvadori, J. C. Aguado, and R. M. Lorenzo, “Virtual topology reconfiguration in optical networks

by means of cognition: evaluation and experimental validation [invited]," *IEEE/OSA Journal of Optical Communications and Networking*, vol. 7, pp. A162–A173, Jan. 2015.

[25] F. Morales, M. Ruiz, L. Gifre, L. M. Contreras, V. López, and L. Velasco, "Virtual network topology adaptability based on data analytics for traffic prediction," *IEEE/OSA Journal of Optical Communications and Networking*, vol. 9, pp. A35–A45, Jan. 2017.

[26] Y. Zhang, M. Roughan, N. Duffield, and A. Greenberg, "Fast accurate computation of large-scale IP traffic matrices from link loads," *ACM SIGMETRICS Performance Evaluation Review*, vol. 31, pp. 206–217, June 2003.

[27] A. Medina, N. Taft, K. Salamatian, S. Bhattacharyya, and C. Diot, "Traffic matrix estimation: Existing techniques and new directions," *ACM SIGCOMM Computer Communication Review*, vol. 32, pp. 161–174, Aug. 2002.

[28] Y. Zhang, M. Roughan, C. Lund, and D. Donoho, "An information-theoretic approach to traffic matrix estimation," in *Proceedings of ACM SIGCOMM*, pp. 301–312, Aug. 2003.

[29] A. Soule, A. Nucci, R. L. Cruz, E. Leonardi, and N. Taft, "Estimating dynamic traffic matrices by using viable routing changes," *IEEE/ACM Transactions on Networking*, vol. 15, pp. 485–498, June 2007.

[30] H. Zhou, L. Tan, Q. Zeng, and C. Wu, "Traffic matrix estimation: A neural network approach with extended input and expectation maximization iteration," *Journal of Network and Computer Applications*, vol. 60, pp. 220–232, Jan. 2016.

[31] L. Nie, D. Jiang, and L. Guo, "A convex optimization-based traffic matrix estimation approach in IP-over-WDM backbone networks," *Journal of Network and Computer Applications*, vol. 50, pp. 32–38, Apr. 2015.

[32] Y. Koizumi, T. Miyamura, S. Arakawa, E. Oki, K. Shiomoto, and M. Murata, "Adaptive virtual network topology control based on attractor selection," *IEEE/OSA Journal of Lightwave Technology*, vol. 28, pp. 1720–1731, June 2010.

[33] Y. Minami, S. Arakawa, Y. Koizumi, T. Miyamura, K. Shiomoto, and M. Murata, "Adaptive virtual network topology control in WDM-based optical networks," in *Proceedings of INTERNET*, pp. 49–54, Sept. 2010.

[34] C. Furusawa and K. Kaneko, "A generic mechanism for adaptive growth rate regulation," *PLoS Computational Biology*, vol. 4, p. e3, Jan. 2008.

[35] Y. Baram, "Orthogonal patterns in binary neural networks," *NASA Technical Memorandum No. 100060*, Mar. 1988.

[36] Toshihiko Ohba, Shin'ichi Arakawa, Yuki Koizumi, and Masayuki Murata, "Scalable design method of attractors in noise-induced virtual network topology control," *IEEE/OSA Journal of Optical Communications and Networking*, vol. 7, pp. 851–863, Sept. 2015.

[37] Toshihiko Ohba, Shin'ichi Arakawa, Yuki Koizumi, and Masayuki Murata, "Hierarchical design of an attractor structure for VNT control based on attractor selection," in *Proceedings of IEEE Consumer Communications and Networking Conference*, pp. 330–336, Jan. 2015.

[38] Toshihiko Ohba, Shin'ichi Arakawa, Yuki Koizumi, and Masayuki Murata, "Design and control of an attractor structure for virtual network topology control based on attractor selection," *Technical Report of IEICE(PN2012-79)*, vol. 112, Mar. 2013.

[39] Toshihiko Ohba, Shin'ichi Arakawa, Yuki Koizumi, and Masayuki Murata, "Evaluation of diversity of attractors for virtual network topology control based on attractor selection," *Technical Report of IEICE(PN2013-15)*, vol. 113, pp. 41–46, Aug. 2013.

[40] Toshihiko Ohba, Shin'ichi Arakawa, Yuki Koizumi, and Masayuki Murata, "Hierarchical design of an attractor structure with topological diversity for VNT control based on attractor selection," *Technical Report of IEICE(PN2014-18)*, vol. 114, pp. 43–48, Sept. 2014.

[41] Toshihiko Ohba, Shin'ichi Arakawa, and Masayuki Murata, "Virtual network reconfiguration in elastic optical path networks for future bandwidth allocation," *IEEE/OSA Journal of Optical Communications and Networking*, vol. 8, pp. 633–644, Sept. 2016.

[42] Toshihiko Ohba, Shin'ichi Arakawa, and Masayuki Murata, "Noise-induced virtual network topology control for elastic optical networks," *Technical Report of IEICE(PN2015-33)*, vol. 115, pp. 55–60, Nov. 2015.

[43] Toshihiko Ohba, Shin'ichi Arakawa, and Masayuki Murata, "A bayesian-based approach for virtual network reconfiguration in elastic optical path networks," in *Proceedings of Optical Fiber Communication Conference*, pp. Th1J–7, Mar. 2017.

[44] Toshihiko Ohba, Shin'ichi Arakawa, and Masayuki Murata, "A bayesian-based virtual network reconfiguration in elastic optical path networks," *Technical Report of IEICE(PN2016-33)*, vol. 116, pp. 45–50, Nov. 2016.

[45] Toshihiko Ohba, Shin'ichi Arakawa, and Masayuki Murata, "Virtual network reconfiguration based on bayesian attractor model with linear regression," *Technical Report of IEICE(PN2017-37)*, vol. 117, pp. 57–63, Nov. 2017.

[46] S. Bitzer, J. Bruineberg, and S. J. Kiebel, "A bayesian attractor model for perceptual decision making," *PLoS Computational Biology*, vol. 11, p. e1004442, Aug. 2015.

[47] J. Wu, "A survey of WDM network reconfiguration: Strategies and triggering methods," *Computer Networks*, vol. 55, pp. 2622–2645, May 2011.

[48] D.-R. Din and C.-W. Chou, "Virtual-topology adaptation for mixed-line-rate optical WDM networks under dynamic traffic," in *Proceedings of International Conference on Computer Communications and Networks*, pp. 1–6, Aug. 2014.

[49] X. Zhang, H. Wang, and Z. Zhang, "Survivable green IP over WDM networks against double-link failures," *Computer Networks*, vol. 59, pp. 62–76, Feb. 2014.

[50] I. Ari, B. Hong, E. L. Miller, S. A. Brandt, and D. D. Long, "Managing flash crowds on the Internet," in *Proceedings of Modeling Analysis and Simulation of Computer and Telecommunication Systems*, pp. 246–249, Oct. 2003.

[51] A. Schaeffer-Filho, P. Smith, A. Mauthe, and D. Hutchison, "Network resilience with reusable management patterns," *IEEE Communications Magazine*, vol. 52, pp. 105–115, July 2014.

[52] R. Rojas, *Neural networks: a systematic introduction.* Springer, July 1996.

[53] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2008, Oct. 2008.

[54] B. Chatterjee, N. Sarma, and E. Oki, "Routing and spectrum allocation in elastic optical networks: A tutorial," *IEEE Communications Surveys & Tutorials*, vol. 17, pp. 1776–1800, May 2015.

[55] J. L. Vizcaíno, Y. Ye, and I. T. Monroy, "Energy efficiency analysis for flexible-grid OFDM-based optical networks," *Computer Networks*, vol. 56, pp. 2400–2419, July 2012.

[56] D. Leung, S. Arakawa, M. Murata, and W. D. Grover, "Re-optimization strategies to maximize traffic-carrying readiness in WDM survivable mesh networks," in *Proceedings of Optical Fiber Communication Conference*, vol. 3, p. OWG6, Mar. 2005.

[57] K. Christodoulopoulos, I. Tomkos, and E. Varvarigos, "Elastic bandwidth allocation in flexible OFDM-based optical networks," *Journal of Lightwave Technology*, vol. 29, pp. 1354–1366, Mar. 2011.

[58] R. Wang and B. Mukherjee, "Spectrum management in heterogeneous bandwidth networks," in *Proceedings of IEEE GLOBECOM*, pp. 2907–2911, Dec. 2012.

[59] M. Klinkowski, M. Ruiz, L. Velasco, D. Careglio, V. Lopez, and J. Comellas, "Elastic spectrum allocation for time-varying traffic in flexgrid optical networks," *IEEE Journal on Selected Areas in Communications*, vol. 31, pp. 26–38, Jan. 2013.

[60] H. R. Heekeren, S. Marrett, and L. G. Ungerleider, "The neural systems that mediate human perceptual decision making," *Nature reviews neuroscience*, vol. 9, pp. 467–479, June 2008.

[61] P. R. Fard, H. Park, A. Warkentin, S. J. Kiebel, and S. Bitzer, "A bayesian reformulation of the extended drift-diffusion model in perceptual decision making," *Frontiers in computational neuroscience*, vol. 11, May 2017.

[62] G. E. Box and G. C. Tiao, *Bayesian inference in statistical analysis*, vol. 40. John Wiley & Sons, 2011.

[63] J. J. Hopfield, "Neurons with graded response have collective computational properties like those of two-state neurons," *Proceedings of the national academy of sciences*, vol. 81, pp. 3088–3092, May 1984.

[64] S. S. Haykin *et al.*, *Kalman filtering and neural networks.* Wiley Online Library, 2001.

[65] S. Bitzer, H. Park, F. Blankenburg, and S. J. Kiebel, "Perceptual decision making: drift-diffusion model is equivalent to a bayesian model," *Frontiers in human neuroscience*, vol. 8, p. 102, Feb. 2014.

[66] D. Knuth, *The art of computer programming, Volume 2, Seminumerical Algorithms.* Addison-Wesley, 1969.

[67] A. Gunnar, M. Johansson, and T. Telkamp, "Traffic matrix estimation on a large IP backbone: a comparison on real data," in *Proceedings of ACM SIGCOMM conference on Internet measurement*, pp. 149–160, Oct. 2004.

[68] A. Nucci, A. Sridharan, and N. Taft, “The problem of synthetically generating IP traffic matrices: initial recommendations,” *ACM SIGCOMM Computer Communication Review*, vol. 35, pp. 19–32, July 2005.

[69] G. A. Seber and A. J. Lee, *Linear regression analysis*, vol. 936. John Wiley & Sons, 2012.