

**Master's Thesis**

Title

**Proposal, Evaluation and Experiment of Large-scaled Virtual  
Network Topology Control Method Based on Attractor Selection**

Supervisor

Professor Masayuki Murata

Author

Koji Mizumoto

February 10th, 2014

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

Master's Thesis

Proposal, Evaluation and Experiment of Large-scaled Virtual Network Topology Control Method  
Based on Attractor Selection

Koji Mizumoto

**Abstract**

Now that the Internet plays an important role as an infrastructure, adaptive control of communication network to environmental changes such as traffic changes or node failures and providing planned network quality is desired. Simultaneously, since the amount of traffic transferred in the network is increasing drastically, the network is responsible for accommodating huge amount of traffic. One approach for accommodating large amount of traffic and achieving adaptive control of the network is to construct a Virtual Network Topology (VNT) on Wavelength Division Multiplexing (WDM) network and reconstruct VNTs according to network environment changes. WDM network offers a huge amount of bandwidth thanks to wavelength-division multiplexing, which multiplexes optical signals that operate on different wavelengths, where traffic is transferred by using a VNT that is constructed by establishing a set of lightpaths between network nodes. Research on VNT control methods that reconstruct VNTs against environmental changes and moves to preferable VNTs for current environment have been conducted. However, existing VNT control methods reconstruct VNTs elaborately supposing some kinds of traffic changes or single node failure. These methods may deteriorate the network condition considerably against unexpected traffic changes such as remarkable traffic fluctuations or multiple node failures, whereas it has high adaptability to modeled traffic changes or modeled node failures. Therefore for adapting to a wider variety of traffic changes or node failures and achieving effective transport of traffic, we proposed attractor selection based VNT control method where attractor selection is a model of behavior of organisms during adaptation to unknown changes in their surrounding environment and recovery of their state. We showed that our method is adaptive to traffic demand changes and link failures, and is able to configure suitable VNTs for the traffic demand after deterioration caused by large fluctuation in traffic demand. However, our method has large computational overhead, with

calculation time of  $O(N^4)$  ( $N$  is the number of nodes) and memory of  $O(N^4)$  for calculating a VNT.

In this paper, we propose a new method for reducing the computational overhead of our previous method. The proposed method can handle large networks that have many nodes. By reducing the number of variables used for calculating VNTs, the proposed method relaxes computational overhead to calculation time of  $O(N^3)$  and memory of  $O(N^3)$ . We conducted evaluation with computer simulation, which shows that the proposed method requires only 0.03 sec for a 1000-node network, whereas our previous method requires about 1000 sec and the proposed method maintains the previous method's adaptability to environmental changes. We also verified the feasibility of the proposed method through experiments with a seven-router network. The results of the experiments show that our method recovers the network condition after deterioration caused by environmental changes.

### **Keywords**

Wavelength Division Multiplexing, VNT Control, Attractor Selection

# Contents

- 1 Introduction** **6**
  
- 2 VNT Control Based on Attractor Selection** **9**
  - 2.1 Attractor Selection . . . . . 9
  - 2.2 VNT Control Method Based on Attractor Selection . . . . . 10
  - 2.3 Computational Overhead of the VNT Control Method . . . . . 12
  
- 3 VNT Control Method for Large-scaled Optical Networks** **13**
  - 3.1 Proposed Method . . . . . 13
  - 3.2 Computational Overhead of Proposed Method . . . . . 17
  
- 4 Evaluation with Computer Simulation** **18**
  - 4.1 Evaluation of Computational Overhead . . . . . 18
  - 4.2 Evaluation of Adaptability . . . . . 20
  
- 5 Experimental Verification** **31**
  - 5.1 Experimental Setup . . . . . 31
  - 5.2 Experimental Scenario . . . . . 35
  - 5.3 Experimental Results . . . . . 38
  
- 6 Conclusion** **44**
  
- Acknowledgement** **45**
  
- References** **46**

## List of Figures

1	Reconstruction of a VNT in IP over WDM Network . . . . .	8
2	Difference of the regulatory matrix $W$ between our previous method and the proposed method . . . . .	15
3	Candidates of lightpaths in a 4-node network . . . . .	15
4	Illustrative example of our proposed method . . . . .	16
5	Calculation time of a VNT . . . . .	19
6	Memory consumption . . . . .	19
7	Distribution of maximum link utilization in a 50-node network . . . . .	22
8	Distribution of maximum link utilization in a 1000-node network . . . . .	23
9	Distribution of convergence time for a 50-node network . . . . .	26
10	Distribution of convergence time for a 1000-node network . . . . .	27
11	Effect of noise strength ( $\delta = 200$ ) . . . . .	30
12	Effect of the activity ( $\sigma = 0.20$ ) . . . . .	30
13	Experimental environment . . . . .	32
14	Physical topology of experiments . . . . .	33
15	VNT control server . . . . .	33
16	Scenario A: Traffic change . . . . .	36
17	Scenario B: Single OXC failure . . . . .	37
18	Scenario C: Multiple OXC failures . . . . .	38
19	Changes of maximum link utilization: Scenario A . . . . .	39
20	Changes of maximum link utilization when VNT loses connectivity . . . . .	40
21	Changes of maximum link utilization: Scenario B (single OXC failure) . . . . .	41
22	Changes of maximum link utilization: Scenario B. Activity calculation is modified. . . . .	42
23	Changes of maximum link utilization: Scenario C (multiple OXC failures). Activity calculation is modified. . . . .	43

## List of Tables

1	Mean measurement error of link utilization ( $\mu$ ) and standard deviation of link utilization ( $\sigma$ ) (averaged over 100 trials) . . . . .	34
2	Processing time for re-constructing routing table (averaged over five trials) . . . . .	35

# 1 Introduction

Now that the Internet plays an important role as an infrastructure, adaptive control of communication network to environmental changes such as traffic changes or node failures and providing planned network quality is desired. Simultaneously, since the amount of traffic transferred in the network is increasing drastically, the network is responsible for accommodating huge amount of traffic. These days, there are various applications such as P2P (Peer to Peer) network, VoIP (Voice over Internet Protocol) and VOD (Video On Demand) and it is pointed out that due to these applications, amount of traffic in the network increases and traffic changes rapidly and unpredictably [1, 2]. Therefore, it is necessary to control the network adaptively to traffic changes caused by these applications or unknown future applications. It is also important to sustain the network against node failures caused by, e.g., natural disasters to keep services since we cannot predict all failures.

WDM (Wavelength Division Multiplexing) network is an expected substrate network accommodating huge amount of traffic by using wavelength division multiplexing, which multiplexes several optical signals operating on different wavelengths into a fiber. One approach for accommodating traffic on Internet Protocol over WDM (IP over WDM) network is to construct a VNT (Virtual Network Topology) that consists of optical signals called lightpaths. Figure 1 shows IP over WDM network. IP over WDM network has two layers composed of WDM network and IP network. WDM network consists of OXCs (Optical Cross Connects) that switches optical signals and fibers. In IP network, VNT control constructs a VNT that consists of lightpaths connected between IP routers via OXCs in WDM network. There is traffic demand between IP routers and traffic is transferred through transmitters and receivers on IP routers and lightpaths. Since traffic demand is changed from time to time, it is necessary to construct a suitable VNT for current environment with limited transmitters and receivers. VNT control methods that establish or tear down lightpaths dynamically and reconstruct VNTs according to environmental changes and moves to a preferable VNT for current environment are considered [3–16].

Many VNT control methods have been proposed to adapt to traffic changes. For example, the VNT control method that constructs a static VNT based on previously obtained traffic demand matrices or assumption of traffic demand is proposed [17]. Since the VNTs calculated by the method is optimal for set of certain traffic demand, this method cannot get rid of unexpected

traffic demand changes. VNT control methods that reconstruct VNTs dynamically based on periodical measurement of network condition or detection of degradation of network condition adapt to traffic demand changes by reconstructing VNTs according to traffic changes [18–20]. However, most of the existing VNT control methods suppose periodic and gradual traffic demand changes and do not suppose large and unpredictable traffic changes or node failures. To adapt to node failures, there are some VNT control methods. For example, the VNT control method that minimizes the risk of disaster with the objective function of minimizing disconnection of weighted lightpaths is proposed in [21]. The VNT control method considering cascade failure is also proposed [22]. However, the methods in [21, 22] first enumerates a set of failure scenarios and then design VNT that works under the failures. These methods have no adaptability to environmental changes since VNT may not work against network failures that are not enumerated beforehand. Therefore, developing a VNT control method that reconstructs VNTs adaptively to various environmental changes such as traffic changes or node failures is necessary.

We previously proposed a VNT control method based on attractor selection for adapting to a wider variety of changes in traffic and achieving effective transport of traffic [23]. Attractor selection is a model of behavior of organisms during adaptation to unknown changes in their surrounding environment and recovery of their state. The fundamental concept underlying attractor selection is that a system adapts to environmental changes by selecting a suitable attractor for current surrounding environment. We showed that our method is adaptive to traffic demand changes and link failure [23], and is able to construct suitable VNTs for the traffic demand after a large fluctuation in traffic demand [24]. However, our method has large computational overhead, with calculation time of  $O(N^4)$  ( $N$  is the number of nodes) and memory of  $O(N^4)$  for updating the variables for calculating VNTs. It thus takes about 1000 sec to calculate the VNT for a 1000-node network compared with only 0.01 sec for the 50-node network that is treated in [23, 24].

As the Internet has become a social platform, the number of Internet users is increasing dramatically and amount of traffic is also increasing with new applications. One conceivable way to handle this traffic growth would be to replace existing nodes with higher performance nodes that are able to quickly handle heavy traffic. However, it is not necessarily advantageous to rely on state of the art nodes since there will be limitations on the capabilities of those nodes and capital cost and power consumption also needs to be taken into account. Although it is not feasible to immediately replace all existing nodes with higher performance nodes, it may be desirable to

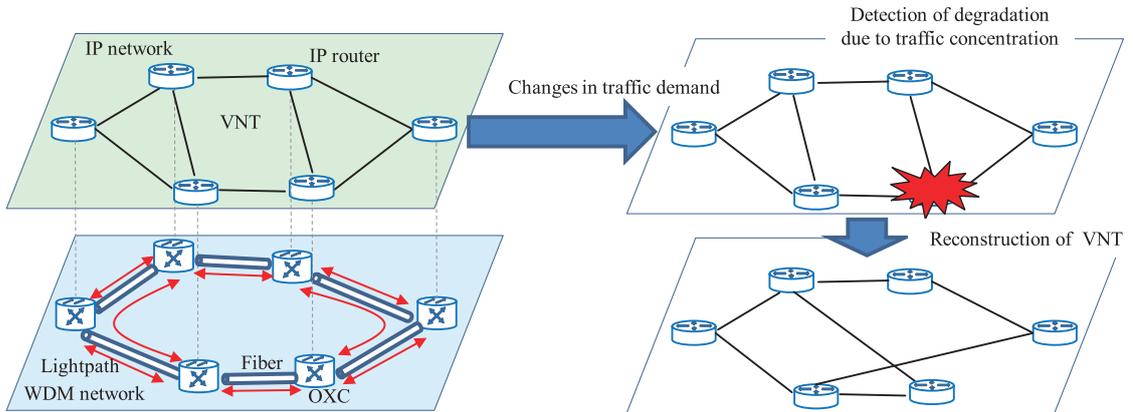


Figure 1: Reconstruction of a VNT in IP over WDM Network

deploy inexpensive, low-power nodes that have lower performance and to then add nodes as the amount of traffic increases. The design of VNTs for large-scale networks is therefore an intriguing problem.

In this paper we propose a large-scaled VNT control method based on attractor selection for reducing the computational overhead of our previous method. The proposed method can handle large networks that have many nodes. By reducing the number of variables used for calculating VNTs, the proposed method relaxes computational overhead to calculation time of  $O(N^3)$  and memory of  $O(N^3)$ . We conducted evaluation with computer simulation, which shows that the proposed method requires only 0.03 sec for a 1000-node network, whereas our previous method requires about 1000 sec, and the proposed method maintains the previous method's adaptability to environmental changes. We also verified feasibility of the proposed method through experiment with seven-node network and show that the proposed method operates on actual machines.

The remainder of this paper is organized as follows. Section 2, introduces the concept of attractor selection which is the key mechanism in our VNT control method and describes our previous VNT control method in order to explain how attractor selection is applied to VNT control. We also see the computational overhead of the method quantitatively in Section 2. Then, we propose a large-scaled VNT control method based on attractor selection in Section 3. We give the evaluation with computer simulation in Section 4 and discuss the results and distributed VNT control manner. We conduct the experimental verification in Section 5. We conclude the paper in Section 6.

## 2 VNT Control Based on Attractor Selection

Here we introduce our original VNT control method based on attractor selection. First, we explain the attractor selection model, which is a key mechanism for VNT control. Second, we explain how to apply attractor selection to VNT control. Finally, we show computational overhead of the method quantitatively.

### 2.1 Attractor Selection

Attractor selection is a model of behavior of organisms during adaptation to unknown changes in their surrounding environment and recovery of their state. The fundamental concept underlying attractor selection is that a system adapts to environmental changes by selecting a suitable attractor for current surrounding environment, where attractors are part of the equilibrium points in the phase space in which the system conditions are preferable. This selection mechanism is based on deterministic and stochastic behaviors, which are controlled by a simple feedback of current system conditions. Stochastic behavior are made dominant in the search for suitable conditions for adapting to the environmental changes as a response to rapid or unpredictable traffic changes. When the system is predominantly controlled by stochastic behavior, the system state fluctuates randomly due to noise and the system searches for a new attractor. Once the state of the system has recovered and the system has gotten close to an attractor, the deterministic behavior takes control of the system again. Deterministic behavior are made dominant and drives the system toward an attractor when current system conditions are suitable for the environment. Attractor selection thus adapts to environmental changes by selecting attractors by using stochastic behavior, deterministic behavior, and simple feedback.

A system driven by attractor selection can be described as

$$\frac{d\mathbf{x}}{dt} = \alpha \cdot f(\mathbf{x}) + \eta, \quad (1)$$

where  $\mathbf{x} = (x_1, \dots, x_i, \dots, x_n)$  ( $n$  is the number of variables) are variables that represent the state of the system,  $\alpha$  is called the activity and represents feedback of the system state,  $f(\mathbf{x})$  represents the deterministic term, and  $\eta$  represents the stochastic term. The state of the system is thus determined by  $\alpha$ ,  $f(\mathbf{x})$ , and  $\eta$ . When the system conditions are suitable for current environment,  $\alpha$  is set to a large value, and the deterministic term,  $f(\mathbf{x})$ , controls the system and drives the system toward

an attractor. When  $\alpha$  is small, the stochastic term,  $\eta$ , controls the system and the system state fluctuates randomly as the system searches for a new attractor. A system driven by attractor selection thus flexibly and adaptively responds to environmental changes by selecting between deterministic and stochastic behavior depending on the activity.

## 2.2 VNT Control Method Based on Attractor Selection

To apply attractor selection model to VNT control, we map the variables in Equation (1) by associating  $x_i$  with the state of a possible lightpath  $l_i$ ,  $\mathbf{x}$  with the state of a VNT, and  $\alpha$  with the condition of the IP network. This gives the following equation for  $x_i$ ,

$$\frac{dx_i}{dt} = \alpha \cdot \left( \zeta \left( \sum_j W_{ij} x_j \right) - x_i \right) + \eta, \quad (2)$$

where  $\zeta(\sum_j W_{ij} x_j) - x_i$  represents the deterministic term  $\eta$  represents the stochastic term which is set to white Gaussian noise, and  $\zeta(z) = \tanh((\mu/2)z)$  ( $\mu$  is a parameter) is the sigmoidal regulation function. Just like attractor selection found in biological systems, our method configures the suitable VNT for current environment by selecting between deterministic and stochastic behavior according to feedback from the IP network.

Whether a lightpath  $l_i$  is established depends on the value of  $x_i$ , which takes values of between -1.0 and 1.0, with  $l_i$  established if and only if  $x_i$  is greater than or equal to 0.0. An established lightpath  $l_i$  is torn down if  $x_i$  is less than 0.0. Note that our method sometimes fails to establish lightpaths due to constraints on the number of transmitters or receivers. In this case, we establish lightpaths in descending order of  $x_i$ .

### 2.2.1 Activity

We use the maximum link utilization on the IP network as a metric indicating the state of the IP network and convert maximum link utilization into activity as follows

$$\alpha = \frac{1}{1 + \exp(\delta \cdot (u_{max} - \zeta))}, \quad (3)$$

where  $\delta$  represents the gradient of this function and the constant  $\zeta$  is the threshold for  $\alpha$ . If the maximum link utilization,  $u_{max}$ , is greater than  $\zeta$ ,  $\alpha$  rapidly approaches 0 due to the poor state of the IP network. The system is then governed by noise and searches for a new attractor. When

the maximum link utilization is less than  $\zeta$ ,  $\alpha$  increases rapidly due to the good state of the IP network.

To determine the maximum link utilization, we collect the traffic volume on all links and select the maximum values. This information is easily and directly retrieved by using the Simple Network Management Protocol (SNMP).

### 2.2.2 Attractor Structure

The regulatory matrix  $\mathbf{W}$  contains elements  $W_{ij}$  from Equation (2) and is an important parameter since it determines the locations of attractors in phase space. Since our method selects one attractor and constructs the VNT corresponding to the selected attractor, the definition of  $\mathbf{W}$  is a challenge. To define arbitrary attractors in phase space, we use knowledge of the Hopfield neural network [25], which has a formula structure similar to Equation (2). The dynamics of our method are expressed by Equation (2). From the perspective of dynamical systems,  $\alpha$  is a constant that determines the convergence speed. The noise  $\eta$  is Gaussian white noise with a mean value of zero. These values do not affect equilibrium points in phase space, which are the attractors in our method. The equilibrium points are therefore determined by the following differential equation:

$$\frac{dx_i}{dt} = \zeta \left( \sum_j W_{ij} x_j \right) - x_i.$$

This is the same formula as a continuous Hopfield network [25], and we therefore use knowledge of associative memory to store arbitrary attractors in phase space.

Suppose we store  $K$  VNTs as attractors.  $\mathbf{W}$  is then defined as

$$\mathbf{W} = \mathbf{X}^+ \mathbf{X}, \quad (4)$$

where  $\mathbf{X}$  represents a matrix in which the  $s$  th row is the  $s$  th attractor.  $\mathbf{X}^+$  represents the pseudo inverse matrix of  $\mathbf{X}$ .

### 2.2.3 Dynamic Reconfiguration of Attractor Structure

If Equation (4) is used to define the attractor structure,  $\mathbf{x}$  tries to converge to any of the stored attractors. However, it does not always converge to an attractor due to the stochastic term. The system might find a suitable VNT for current environment, but that VNT might not be an attractor. If the VNT is one of the attractors, the system converges to the VNT and achieves a good

condition. Thus, when the configured VNT is preferable for current environment, we reconfigure the attractor structure so that the configured VNT is one of the attractors. The VNT suitable for current environment is therefore added to the attractor structure. Adding suitable VNTs to the attractors gives the system adaptability to a wide variety of traffic demand and quick convergence to attractors. One conceivable approach for adapting to a wide variety of traffic demand is to add VNTs that are suitable for current environment to the set of attractors whenever the system finds such VNTs. However, this approach is almost impossible due to the memory capacity limitations of the Hopfield network [25], in which the number of points that a system can memorize is up to 15 % of the number of variables  $|\mathbf{x}|$ , that is,  $N^2 \times 0.15$  in an  $N$  node network. By incorporating new VNTs suitable for current network environment into the attractor structure, we attain adaptability to a wide variety of traffic demand with only a limited number of attractors held in the attractor structure.

To achieve this, we simply use a first-in first-out (FIFO) policy for managing attractors. When we add a new VNT to attractors, we remove the oldest attractor from the set of attractors. Whether or not a configured VNT is to be added to the attractors is decided based on a target maximum link utilization value  $\theta$ . Our control objective is to configure a VNT that has maximum link utilization less than  $\theta$ . When a configured VNT achieves the control objective, the VNT is considered to be a suitable VNT for current environment and is added to the attractors. However, such configured VNTs differ every time due to the stochastic term. Adding these VNTs every time would result in many attractors that are close to one of the stored attractors, leading to a lack of diversity of attractors. To prevent this, we add a VNT at time  $t$  to the attractors only if  $u_{max}(t) > \theta$  and  $u_{max}(t+1) \leq \theta$ . Note that  $u_{max}(t)$  represents the maximum link utilization of the VNT at time  $t - 1$ .

We use this simple first-in, first-out (FIFO) policy and the target maximum link utilization for reconfiguration of attractor structure. Other strategies may also be possible which we intend to examine in the future.

### 2.3 Computational Overhead of the VNT Control Method

Here, we evaluate the computational overhead of the method from the calculation time and memory consumption point of view.

### 2.3.1 Calculation Time

When a VNT is calculated, the values of  $x_i$  in Equation (2) are updated by using Euler's method. The calculation for updating  $x_i$  is given by

$$x_i(t+1) = \alpha \cdot \left( \zeta \left( \sum_j W_{ij} x_j(t) \right) - x_i(t) \right) + \eta + x_i(t),$$

where  $x_i(t)$  represents the value of  $x_i$  at time  $t$ . This equation indicates that the calculation time depends on  $\sum_j W_{ij} x_j$ , that is, the calculation time for updating  $x_i$  correspond to  $|\mathbf{x}|$ , which represents the number of variables. The calculation time of updating  $\mathbf{x}$  is therefore  $O(|\mathbf{x}|) \times |\mathbf{x}|$  since each  $x_i$  takes  $O(|\mathbf{x}|)$  to update its value and there are  $|\mathbf{x}|$  variables. Since  $x_i$  is assigned to all possible  $N^2$  ( $N$  is the number of nodes) lightpaths in the case of VNT control method, the calculation time of the method is  $O(N^2) \times N^2$ .

### 2.3.2 Memory Consumption

The memory consumption for calculating a VNT is decided by the number of elements that the regulatory matrix  $\mathbf{W}$  has and it is  $O(N^4)$  since the dimensions of  $\mathbf{W}$  is  $N^2 \times N^2$ .

## 3 VNT Control Method for Large-scaled Optical Networks

Here, we propose large-scaled VNT control method based on attractor selection for reducing the computational overhead of our previous method.

### 3.1 Proposed Method

The calculation time of updating  $x_i$  depends on  $\sum_j W_{ij} x_j$  as described in Section 2.3. We propose large-scaled VNT control method by cutting the number of calculations in  $\sum_j W_{ij} x_j$ . We explain how to reduce calculation time in detail below.

$\sum_j W_{ij} x_j$  represents the effect of other lightpaths  $x_j$  on  $x_i$  when updating its value. Each  $x_i$  searches for attractors by taking into account of effect of  $x_j$ . We reduce the calculation time by reducing the number of variables  $x_j$  that have an effect on  $x_i$ , which mean that we reduce the number of lightpaths  $l_j$  that have an effect on  $l_i$ . Suppose that  $\mathbf{l}^k$  represents the set of lightpaths that originate from node  $k$  and  $\mathbf{x}^k$  represents the set of variables for lightpaths  $l_i (\in \mathbf{l}^k)$ . We

think the situation that proposed method uses only  $\mathbf{x}^k$ , when updating  $x_i(\in \mathbf{x}^k)$ , where whether  $l_i(\in \mathbf{l}^k)$  is established or torn down is affected by  $l_j(\in \mathbf{l}^k)$ . The reduction appears in the form of miniaturization of regulatory matrix  $\mathbf{W}$ . Figure 2 shows the part in  $\mathbf{W}$  which is used to calculate a VNT. In this figure, the matrix is  $n^2$  by  $n^2$ , in which  $n$  is the number of variables.  $W_{ij}$  represents the effect between  $x_i$  and  $x_j$ . Rectangular box in the matrix on the left represents the part of the matrix that is used for updating  $x_i$  in the previous method, i.e., all elements in  $i$  th row are used when  $x_i$  is updated. On the other hand, in the proposed method on the right, the part of the matrix that is used for updating  $x_i$  is reduced. Slashed boxes represent eliminated part that are no longer used to calculate a VNT and only  $\mathbf{w}^k$  that is set of  $W_{ij}(l_i, l_j \in \mathbf{l}^k)$ , is used for updating  $x_i$ . In the proposed method, since only lightpaths originating from the same node as lightpath  $l_i$  are took into account when  $x_i$  is updated, the effect from lightpaths originating from different nodes is eliminated, whereas all elements in  $i$  th row in the matrix are used in the previous method.

Below, illustrative example is explained. Figure 3 shows lightpaths that are candidate to be established in a four node network. Directional lightpaths can be established between any two nodes. The difference of set of  $x_j$  that are used for updating  $x_i$  between our previous method and the proposed method in a four node network is illustrated in Figure 4. The lighpath that corresponds to  $x_i$  is depicted as a red arrow form node A to node B. The lighpaths that correspond to  $x_j$  are depicted as green dotted arrow. Only lightpaths that originate from node A affect  $l_{AB}$  in the proposed method, whereas all lightpaths affect  $l_{AB}$  in the previous method. In this way, the proposed method achieves a reduction in computational overhead.

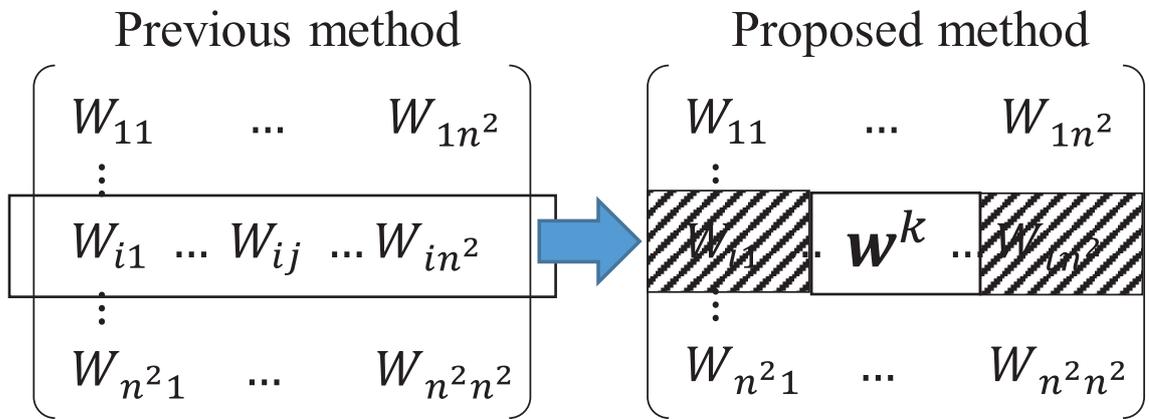


Figure 2: Difference of the regulatory matrix  $\mathbf{W}$  between our previous method and the proposed method

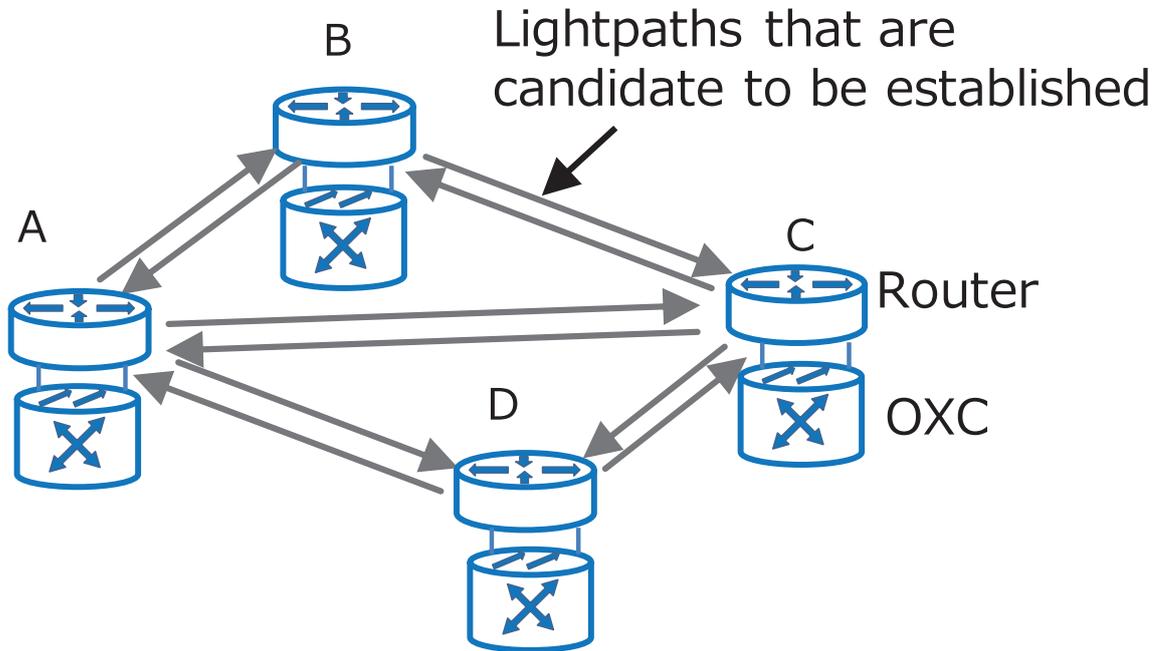
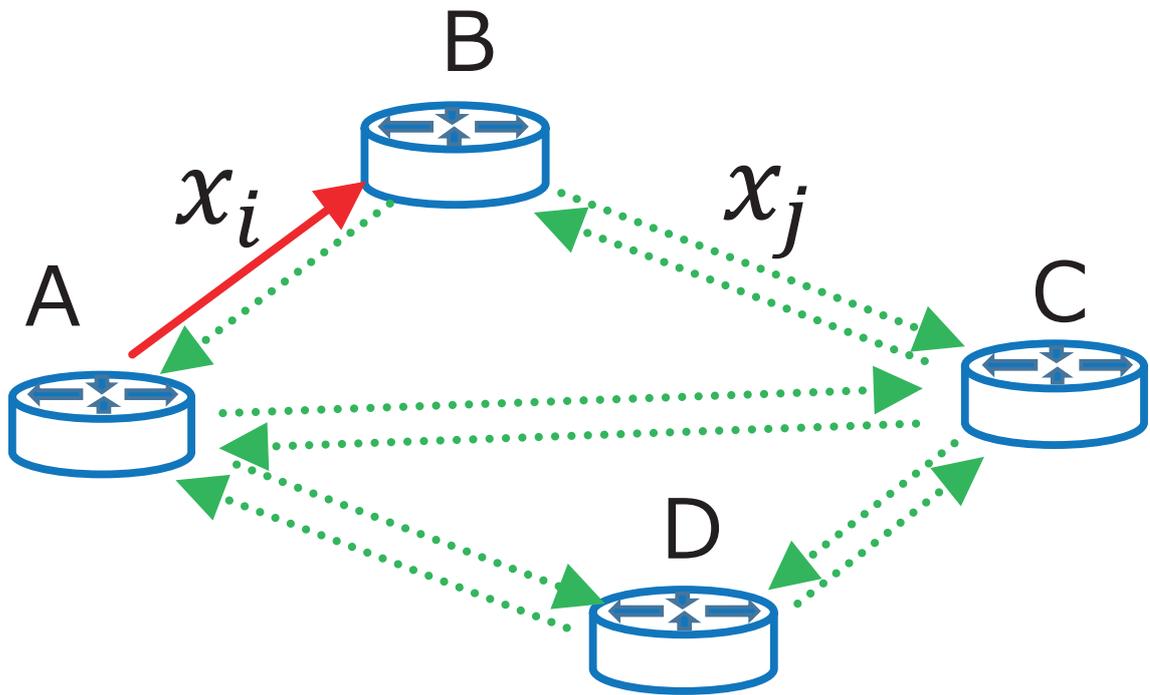
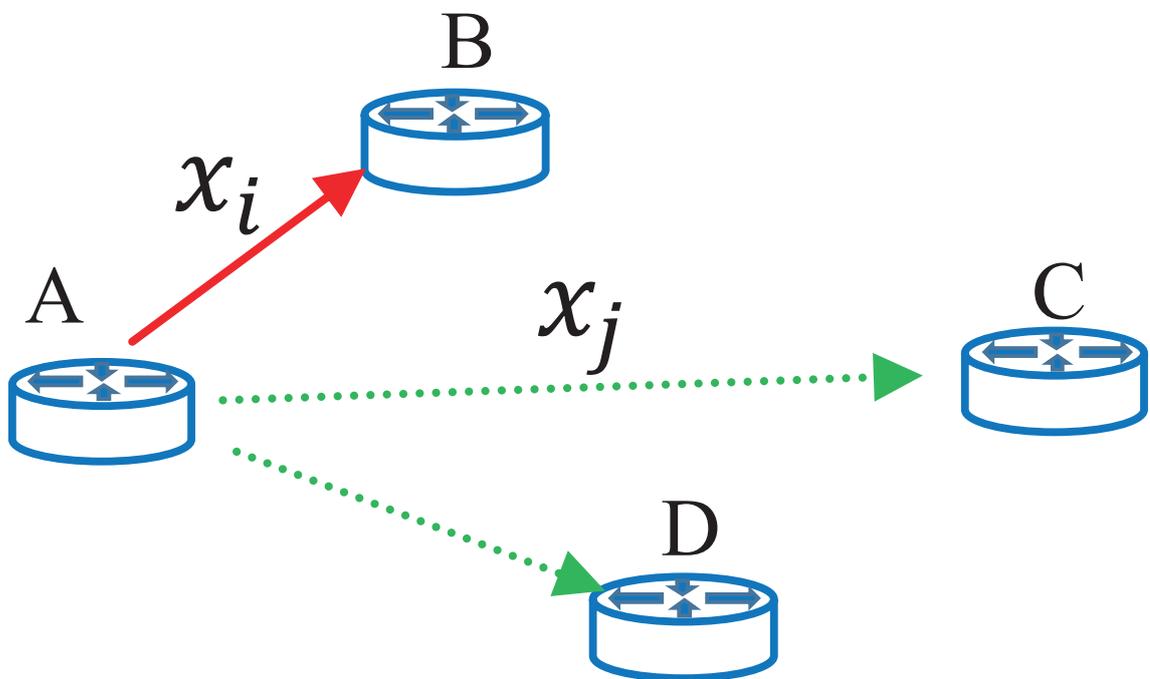


Figure 3: Candidates of lightpaths in a 4-node network



(a) Previous method



(b) Proposed method

Figure 4: Illustrative example of our proposed method

The proposed method limits the variables that have an effect; in other words,  $l_i$  is influenced by only lightpaths originating from the same node as  $l_i$  as explained above. Since each  $l_i$  takes only subset of all lightpaths into account to update its value, it is possible that a configured VNT might not be suitable from the global point of view even though it is good from the local point of view. How the number of variables that affect other variables is reduced is therefore important for constructing appropriate VNTs as a whole. To reduce variables, we define the magnitude of influence between lightpaths by using physical distance. The effect that one lightpath has on another lightpath is more important if the distance between the two lightpaths is shorter, less important if the distance is longer and most important if two lightpaths originate from the same node. We use the effect of only lightpaths originating from the same node since that is the most important effect.

The condition of the IP network is used as feedback that defines the activity in our method. Even if a VNT is good locally but is poor globally, the poor condition of the IP network is reflected in the activity, the stochastic term becomes dominant, and the system searches for a suitable VNT for current environment.

## 3.2 Computational Overhead of Proposed Method

We explain the computational overhead of proposed method.

### 3.2.1 Calculation Time

The number of variables used when proposed method updates  $x_i$  is  $N$ . Thus, calculation time of updating  $x_i$  is  $O(N)$  and the total calculation time is  $O(N) \times N^2$  since there are  $N^2$  variables. Note that variables  $x_i$  and  $x_j$  can be calculated in parallel if lightpaths  $l_i$  and  $l_j$  originate from different nodes. The time to calculate a VNT is thus reduced to  $O(N) \times N$ .

### 3.2.2 Memory Consumption

The memory consumption for calculating a VNT is decided by the number of elements that the regulatory matrix  $\mathbf{W}$  has. In proposed method, only  $w^k$  is necessary for calculating  $x_i$  in  $i$  th row in  $\mathbf{W}$ . The memory consumption for each row in  $\mathbf{W}$  is reduced to  $O(N)$ , and the number of rows is  $N^2$ . Therefore, the memory consumption is  $O(N^3)$ .

## 4 Evaluation with Computer Simulation

We evaluate the computational complexity and adaptability of our method by computer simulation.

### 4.1 Evaluation of Computational Overhead

Quantitatively, in Section 2.3, 3.2, we already show the computational overhead of previous method and proposed method. Here, we ran computer simulations to evaluate the computational overhead. We used a machine that had 64 GB of memory and two Intel (R) Xeon (R) X5670 CPUs (2.93 GHz, 3.06 GHz), each of which had 6 cores. We wrote the simulator in C# and used parallel programming in .NET Framework.

#### 4.1.1 Calculation Time

Figure 5 shows the results for the time to calculate a VNT for networks having various numbers of nodes. The horizontal axis shows the number of nodes and the vertical axis shows the time taken to calculate a VNT. “Proposed method” represents the results of proposed method and “Previous method” represents the results of the previous method. Note that we use the fitted function for the previous method when the number of nodes is greater than 350 since it is impossible to evaluate the method due to a lack of memory. As shown in Figure 5, proposed method reduces the time taken to calculate a VNT from the previous method. The difference in calculation time between two methods increases as the number of nodes increases. For a 100 node topology, the calculation times of the proposed method and the previous method are 0.0008 sec and 0.32 sec, respectively, with both methods calculating a VNT in less than 1 sec. However, for the 1000-node topology, only proposed method is able to calculate a VNT in less than 1 sec. The calculation time of proposed method is 0.23 sec whereas that of the previous method is more than 1000 sec, or roughly 15 min.

In a real network, it may be desirable to reconstruct VNTs in short time intervals to adapt to rapid and unpredictable environmental changes. We aim to reconstruct VNTs in a few tens of seconds’ interval against rapid and unpredictable environmental changes. For reconstructing VNTs, there are three processes. Firstly, we collect network information, secondly calculate a VNT, and finally establish or tear down lightpaths. Only link utilization is used as network information in the proposed method. Link utilization is easily retrieved by SNMP, so we can collect network

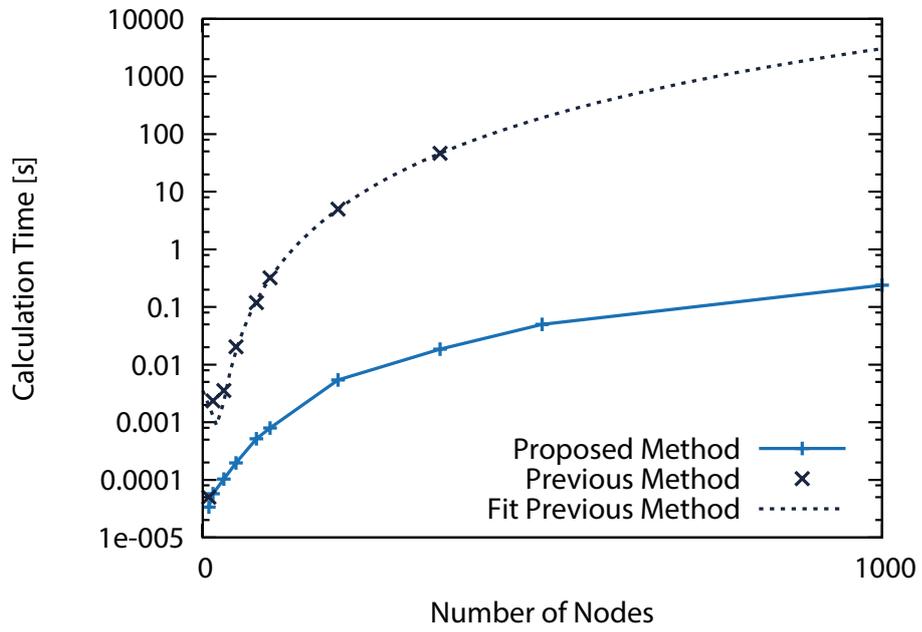


Figure 5: Calculation time of a VNT

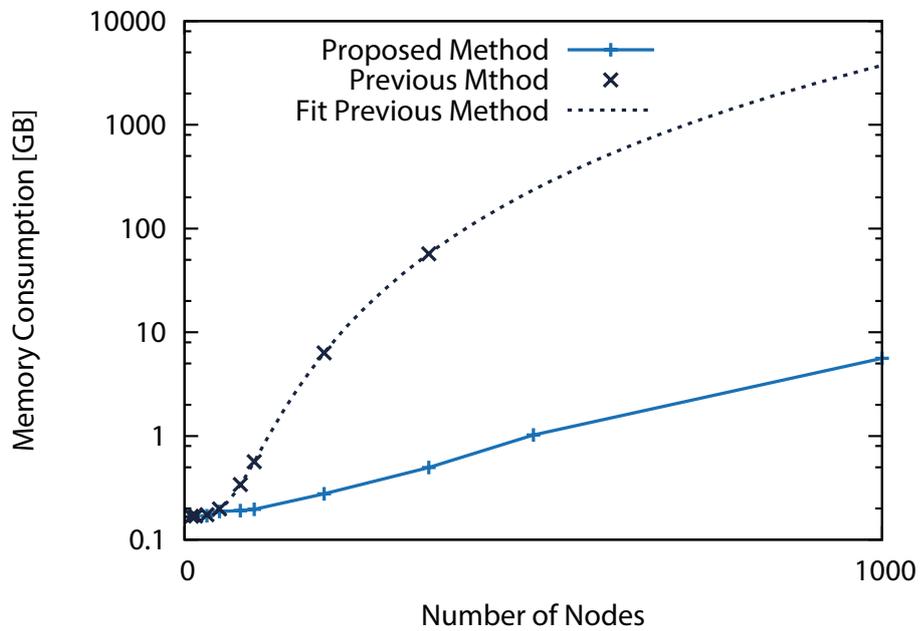


Figure 6: Memory consumption

information in several tens of seconds. Proposed method calculates a VNT in less than 1 sec even when the number of nodes in the network is 1000 and establishing and tearing down lightpaths are completed in several seconds. Therefore, it is possible to reconstruct VNTs for a 1000-node network in a few tens of seconds with proposed method compared.

#### 4.1.2 Memory Consumption

In addition to calculation time, memory consumption should be taken into consideration since large memory consumption is necessary for the regulatory matrix  $\mathbf{W}$ . Figure 6 shows the results for memory consumption that is necessary for calculating a VNT for networks having various numbers of nodes. As shown in Figure 6, proposed method reduces the memory consumption from the previous method. The memory consumption in proposed method is 6 GB in a 1000-node topology while that in the previous method is 5 TB, whereas the difference of the memory consumption between two methods is small in small node topologies. Even in 1000-node topology, proposed method can be handled by generic computer.

## 4.2 Evaluation of Adaptability

We evaluate the adaptability of proposed method to variations in traffic demand. Our control objective is to construct a VNT that has maximum link utilization less than the target link utilization ( $\theta=0.4$ ). The evaluation is made in terms of two metrics, the maximum link utilization and convergence time. Convergence time is defined as the number of VNT reconstructions before achieving the control objective.

The physical topologies in the simulations are generated randomly. Each node in each network has the same number of transmitters and receivers. Link capacity is 100 Gbps. Traffic demand matrices are generated randomly. The elements of the matrices follow a log-normal distribution of standard deviation 1 according to the observation in [26]. The IP network transports traffic on a VNT by following shortest path routing. we set the constant values in the definition of Equation 2 and 3 to  $\mu = 20$ ,  $\delta = 50$  and  $\zeta = 0.5$ .

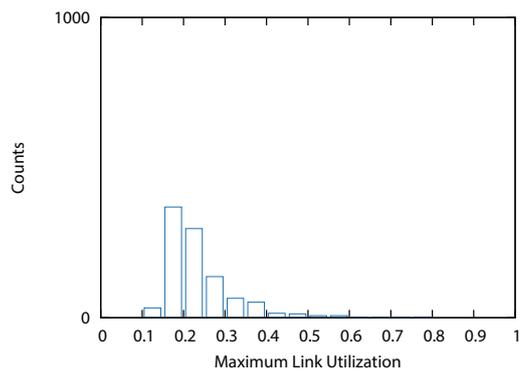
### 4.2.1 Comparison Methods

We use two existing heuristic VNT control methods for comparison. One of the heuristic VNT design algorithms is called the minimum delay logical topology design algorithm (MLDA) that was proposed in [18]. MLDA constructs VNTs on the basis of a given traffic demand matrix. The main objective of MLDA is to minimize the maximum link utilization. The basic idea behind MLDA is to place lightpaths between nodes in descending order of traffic demand. The other VNT control method is the increasing multi-hop logical topology design algorithm (I-MLTDA) [19, 20]. The main objective of I-MLTDA is to maximize multi-hop traffic, that is, to maximize accommodated traffic on the VNT by minimizing traffic on multi-hop lightpaths. To determine where to establish lightpaths, I-MLTDA uses the hop counts on the constructed VNT and traffic demand matrix information. Let  $H_{sd}$  and  $\Delta_{sd}$  denote the minimum number of hops needed to send traffic from a source  $s$  to a destination  $d$  and the amount of traffic from  $s$  to  $d$ , respectively. Note that I-MLTDA re-computes  $H_{sd}$  each time in order to select lightpaths. I-MLTDA places lightpaths in descending order of weight  $\Delta_{ij} \cdot (H_{ij} - 1)$ . Since the largest  $\Delta_{ij} \cdot (H_{ij} - 1)$  corresponds to the largest amount of multi-hop traffic that can be carried in one hop as opposed to using multiple hops, this results in maximization of the accommodated traffic on the resulting VNT.

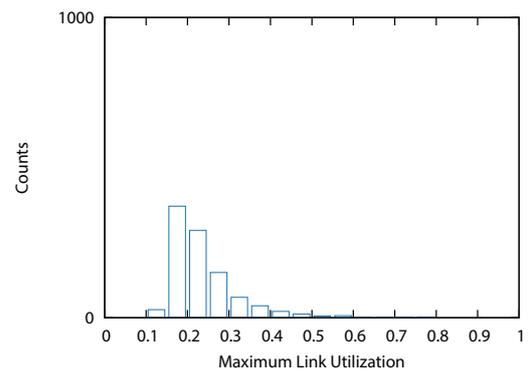
While most current heuristic VNT control methods, including MLDA and I-MLTDA, use information from the traffic demand matrix, proposed method uses only link utilization information. It is generally difficult to retrieve information from the traffic demand matrix [27, 28]. Several methods have therefore been proposed for estimating traffic matrix information from link load information [28]. However, the estimation accuracy of the methods depends on the type of traffic, as discussed in [27]. Therefore, to evaluate the potential adaptability of current heuristic VNT control methods, MLDA and I-MLTDA, we do not use traffic estimation methods, and they are given real traffic matrix information.

### 4.2.2 Maximum Link Utilization

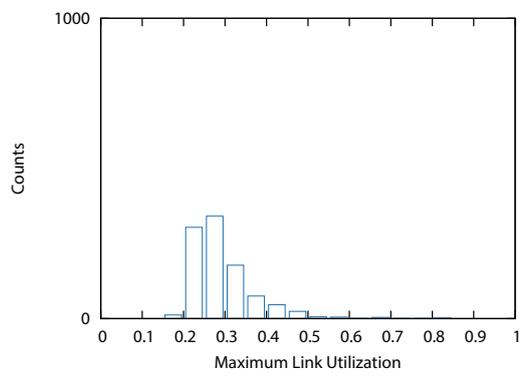
Figure 7 shows a histogram of the maximum link utilization of VNTs constructed by each method when given a 50-node network and 1000 patterns of traffic demand matrices. The histograms of the attractor selection based VNT methods summarize the results when those methods converge to attractors. Note that we set the maximum number of reconstruction trials to 1000 since our



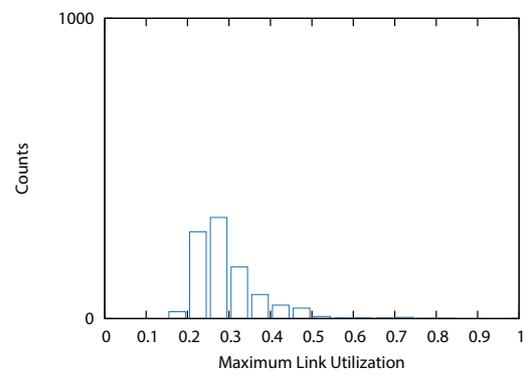
(a) MLDA



(b) I-MLTDA

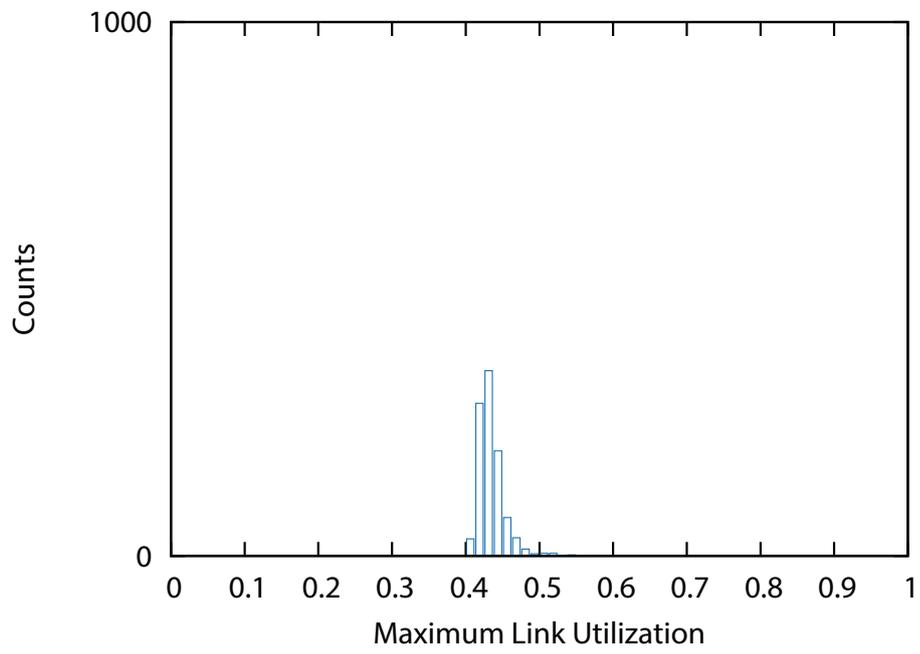


(c) Proposed method

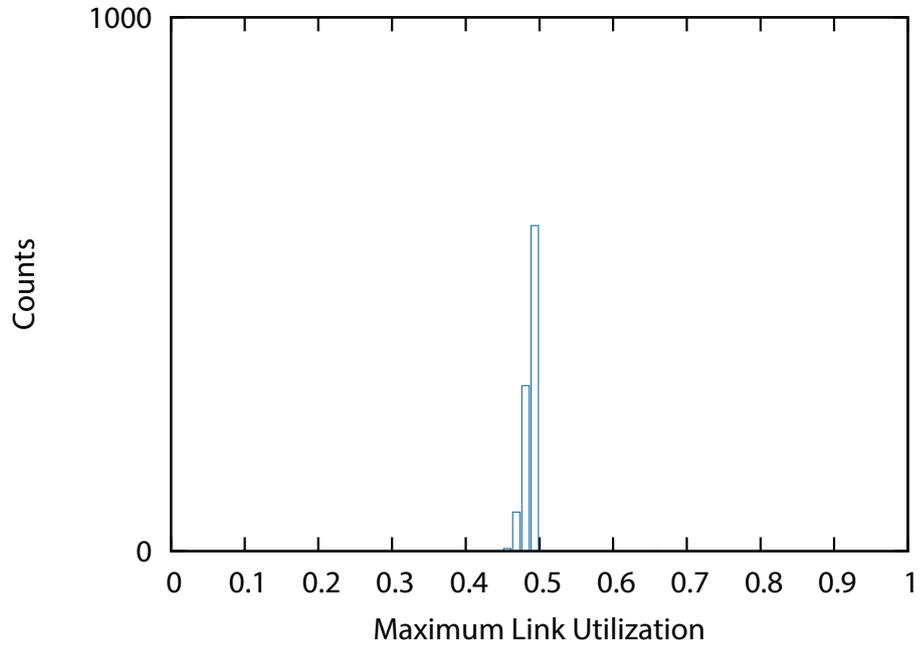


(d) Previous method

Figure 7: Distribution of maximum link utilization in a 50-node network



(a) MLDA



(b) Proposed method

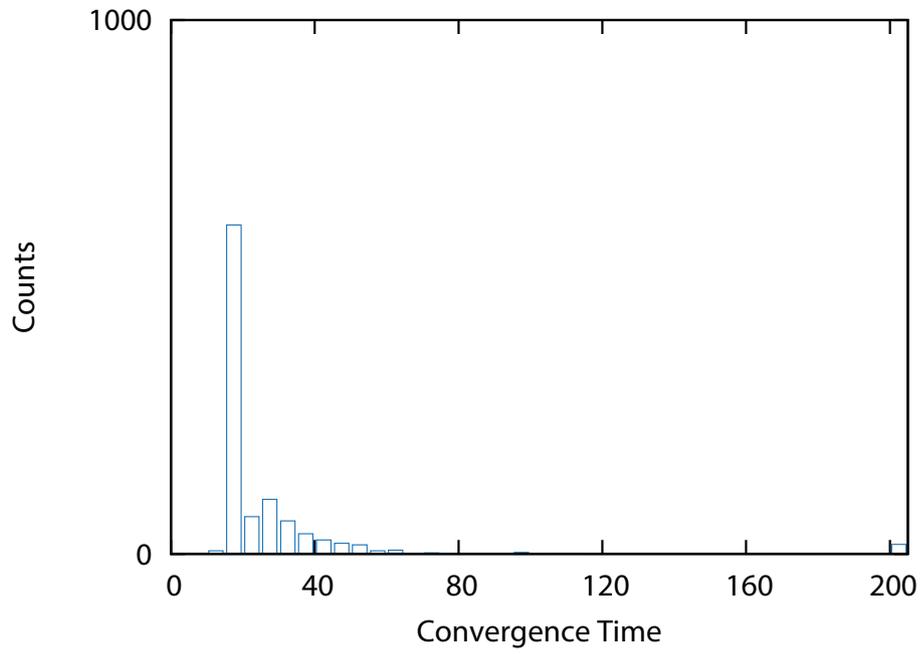
Figure 8: Distribution of maximum link utilization in a 1000-node network

methods sometimes fail to achieve the control objective. We use the maximum link utilization after 1000 trials when the methods fail to achieve the control objective. We adjust traffic demand volume so that the maximum link utilization of VNTs calculated by MLDA or I-MLTDA is less than target maximum link utilization ( $\theta = 0.5$ ) although it sometimes exceeds  $\theta$ . The results of MLDA and I-MLTDA are shown in Figures 7(a) and 7(b). The maximum link utilization is less than  $\theta$  most of the time and the control objectives of MLDA and I-MLTDA are achieved in 981 and 982 of the traffic patterns respectively. Figures 7(c) and 7(d) show the results of proposed method and our previous method. Although both methods achieve the control objective against most patterns of traffic demand, the number of cases in which the control objective is achieved by proposed method is 980 which is less than the 981 of our previous method. This degradation may have been caused by reducing the number of variables that influence have an effect in proposed method. However, the difference in the number of traffic patterns where the control objective is achieved between the two methods is small and proposed method inherits the adaptability to traffic demand variation from the previous method. The reason why proposed method almost has the same adaptability to variation in traffic demand as the previous method in spite of the reduction of the computational overhead is thanks to stochastic term. In proposed method, to reduce computational overhead, variables are updated by using part of the regulatory matrix, i.e., the effect from the lightpaths originating from the same node. Thus, inappropriate VNTs as a whole may be configured. However, proposed method achieves the control objective as same times as the previous method, and this may be caused by that random search with stochastic term explores widely in a phase space and the VNT achieving the control objective is found. Note that the maximum link utilization of both MLDA and I-MLTDA is less than that of our attractor selection-based methods. This is because MLDA and I-MLTDA use information from the traffic demand matrices and therefore configure suboptimal VNTs.

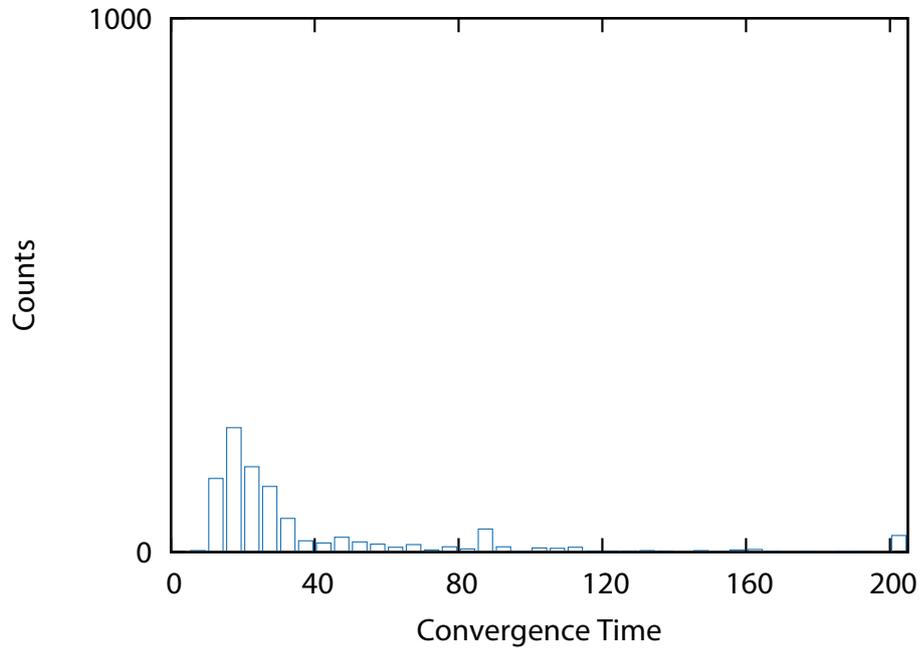
Figure 8 shows histograms of the maximum link utilization for a given 1000-node topology and 1000 patterns of traffic demand matrices for proposed method and MLDA. For a network of this size, we could not evaluate our previous method due to lack of memory, and could not evaluate I-MLTDA due to the computational complexity. Most of the maximum link utilization values of VNTs calculated by MLDA take values between 0.4 and 0.5 as shown in Figure 8(a). This implies that there are fewer VNTs that achieve the objective unlike the evaluation of the 50-node topology. Figure 8(b) shows the results of proposed method, which achieves the objective against most

patterns of traffic demand. The maximum link utilization mainly takes values of between 0.45 and 0.5. The number of patterns for which proposed method achieves the control objective is 998, which is greater than 985 of MLDA. This implies that proposed method achieves the control objective against more traffic demand matrices even though the maximum link utilization of VNTs tends to be greater in proposed method than MLDA. Hence, proposed method may construct VNTs that achieve the control objective against traffic demand matrices for which the VNTs constructed by MLDA do not achieve the objective. Note that the situation did not occur in our simulation where a VNT calculated by MLDA achieved the objective against some traffic demand pattern for which the VNT calculated by proposed method did not achieve the objective.

We also evaluated the methods using a 2000 node topology. The results exhibited the same tendencies as the results for the 1000-node topology, and so we omit the results.



(a) Proposed method



(b) Previous method

Figure 9: Distribution of convergence time for a 50-node network

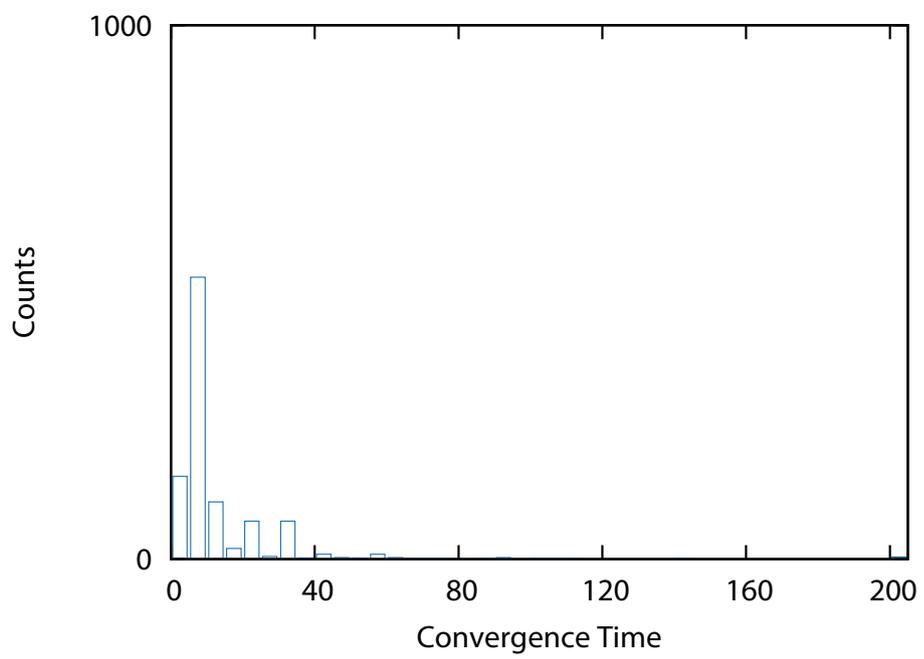


Figure 10: Distribution of convergence time for a 1000-node network

### 4.2.3 Convergence Time

Figure 9 shows histograms of the convergence time. The horizontal axis represents the convergence time with the histograms plotted up to a time of 200. The bin to the right of the time 200 represents the number of times the control objective failed to be achieved after reconstructing VNTs 200 times. As can be seen, both of the convergence time of proposed method and previous method are less than 40 for most cases of traffic demand patterns, Whereas there is some difference which may be caused by stochastic behavior. Figure 10 shows the results for a 1000-node topology. Proposed method achieves the control objective in fewer than 40 reconfiguration trials in many cases. This implies that proposed method finds a suitable VNT for current environment and converges to it even if the number of node increases and the phase space becomes large.

It is desirable to converge to an attractor quickly in order to adapt to environmental changes in a short amount of time. To achieve short time convergence, it is important to add suitable VNTs for current environment to the attractor structure. It becomes easier and faster to converge to attractors if there are more suitable VNTs for current environment. We still do not have guidelines for which VNTs to use as attractors. The problem of how to construct the attractor structure is left as work for the future. Dealing with cases in which our method is unable to achieve the objective is also a challenge. Our method continues to search for a VNT until a suitable VNT for current environment can be found even if there is no VNT that achieves the control objective. To achieve fast convergence and to avoid iteration of reconstruction, our method needs some mechanism for determining whether there exist VNTs that can achieve the objective. In [29], fast convergence is achieved by using dynamic target maximum link utilization.

### 4.2.4 Effect of Noise Strength

Stochastic behavior, that is noise, in attractor selection has an important role in achieving the adaptability to variations in traffic demand. So, we evaluate the effect of noise strength. Since noise is given as white Gaussian noise of mean 0 and standard deviation  $\sigma$ , we give different  $\sigma$ . In this evaluation, the number of reconfiguration is at most 200. Figure 11 shows the operational ratio vs. the noise strength  $\sigma$ . The horizontal axis is  $\sigma$  and the vertical axis is the operational ratio. When  $\sigma$  is 0, a VNT will not be reconfigured since there is no effect of noise. Control objective cannot be achieve against deterioration after traffic changes. When  $\sigma$  is 0.05, the effect of noise

is too small to achieve the control objective, though VNTs are reconfigured. When  $\sigma$  is 0.10, the noise become strong enough to search for suitable VNT for current environment and achieve the control objective in many cases. The operational ratio is high when  $\sigma$  is between 0.10 and 0.20. This is the adequate strength for searching VNT and converging a preferable VNT for current environment. When  $\sigma$  is 0.25, the noise it too strong to converge VNT. From the above results,  $\sigma$  should be big enough to search for VNT and small enough to converge VNT.

#### **4.2.5 Effect of Activity**

We evaluate the effect of activity  $\alpha$ , by using various  $\delta$  in Equation(3). Parameter  $\delta$  represents the gradient of  $\alpha$ . Since the gradient of  $\alpha$  is steep with large  $\delta$ , our method becomes sensitive to the changes in the maximum link utilization. Figure 12 shows the operational ratio vs.  $\delta$ . In this evaluation, the number of reconfiguration is at most 200. The horizontal axis is  $\delta$  and the vertical axis is the operational ratio. When  $\delta$  is between 50 and 200, the control objective is achieved in most cases. However, operational ratio is low, when  $\delta$  is 10 and 25. This is because our method becomes insensitive to the deterioration of condition, that is maximum link utilization, and VNT will not be reconfigured. From the above results,  $\delta$  should be big enough to reconfigure VNT when the condition get worse.

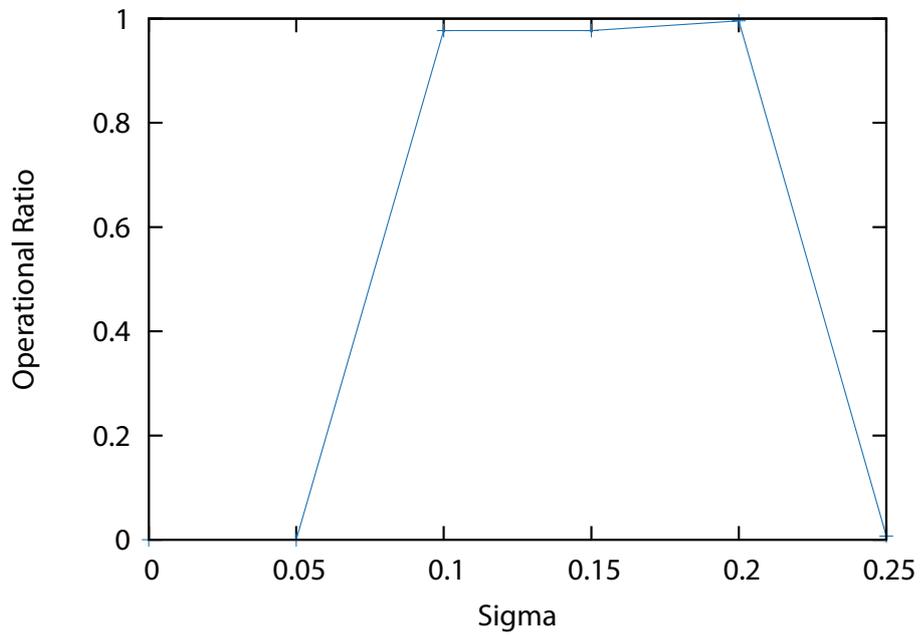


Figure 11: Effect of noise strength ( $\delta = 200$ )

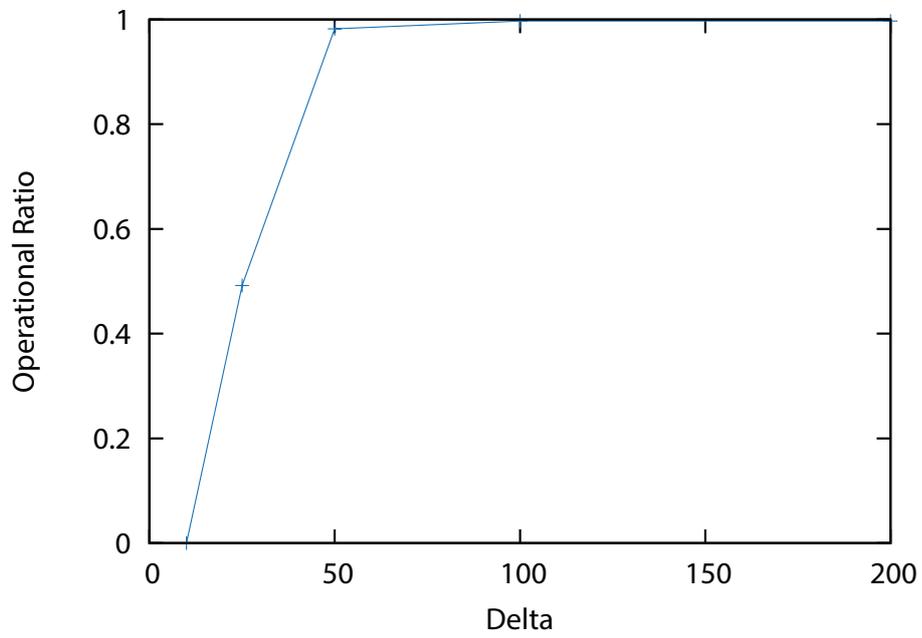


Figure 12: Effect of the activity ( $\sigma = 0.20$ )

## 5 Experimental Verification

We verified feasibility of the proposed method with experiment. First, we explain a network used for the experiments and experimental scenarios. After that, we show our experimental results.

### 5.1 Experimental Setup

Our experimental environment consists of physical topology and a control server. We explain the physical topology and the control server in following sections.

#### 5.1.1 Physical Topology of Experiments

Figure 13 shows equipment used for our experiments. There are seven IP routers and one OXC and they are connected via fibers. We use Juniper MX5 and MX10 for the routers and Glimmerglass System 100 for the OXC. Figure 14 illustrates the physical topology of our experiments. The OXC is logically divided into 10 OXCs. Single line between routers and OXCs in Figure 14 represents a fiber supporting full duplex transmission. In our experiment, each router establishes lightpaths with other routers by using multiple fibers where a fiber accommodates only one lightpath instead of multiplexing multiple lightpaths. We give a static route of each lightpath established between routers. The line speed of router interfaces is 1 Gbps, so the bandwidth of a lightpath is 1 Gbps. We use OSPF (Open Shortest Path First) protocol for routing protocol.

#### 5.1.2 Control Server

Figure 15 shows the overview of VNT control with the control server based on attractor selection. The control server repeatedly performs the following three steps,

- (1) Retrieves link utilization information from IP routers by using SNMP (Simple Network Management Protocol),
- (2) Converts the maximum link utilization to the activity. Our VNT control method then calculates a VNT.
- (3) Sends a request to establish or tear down lightpaths that correspond to the calculated VNT.

The request of establishing a lightpath will be rejected when the resources of the physical network are insufficient.

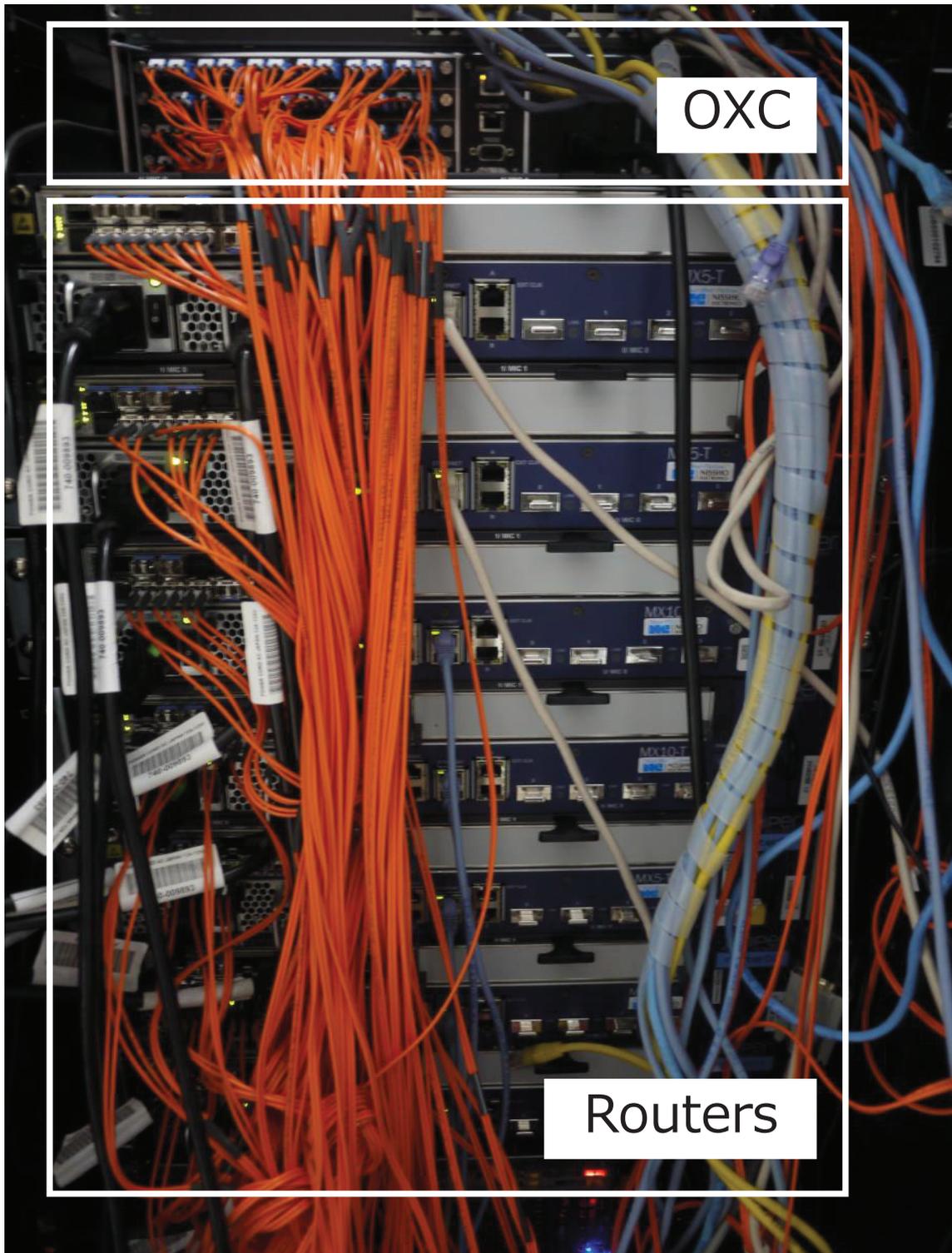


Figure 13: Experimental environment

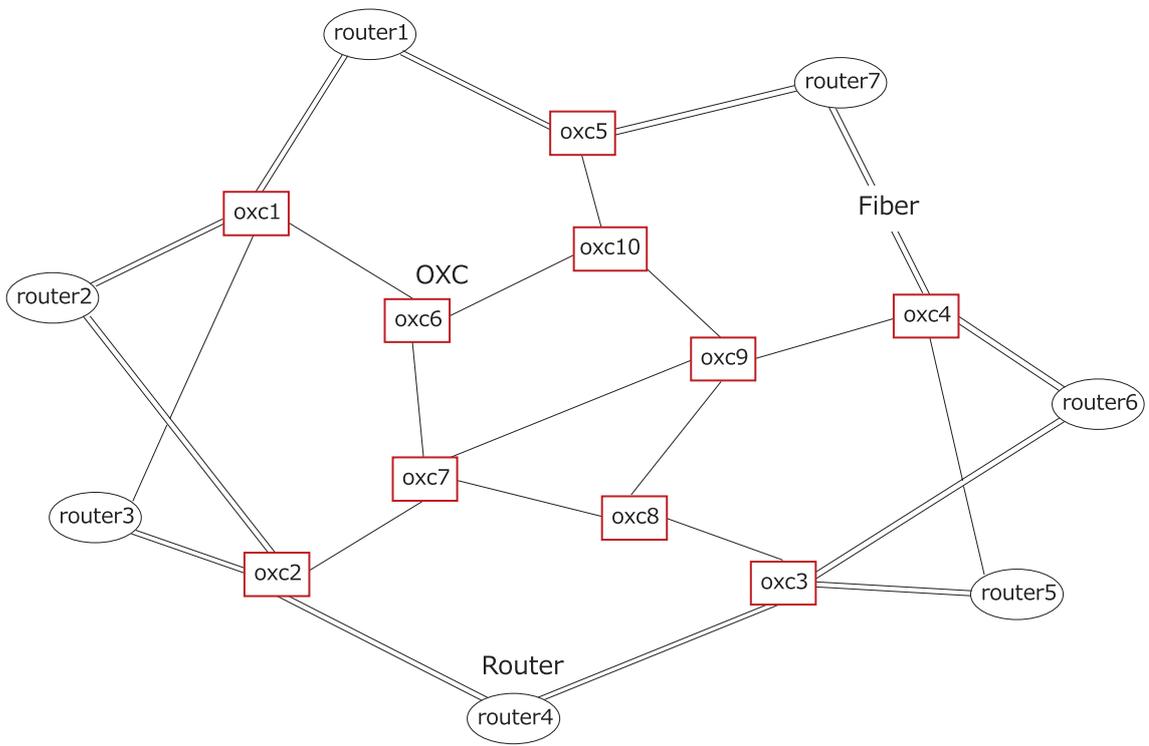


Figure 14: Physical topology of experiments

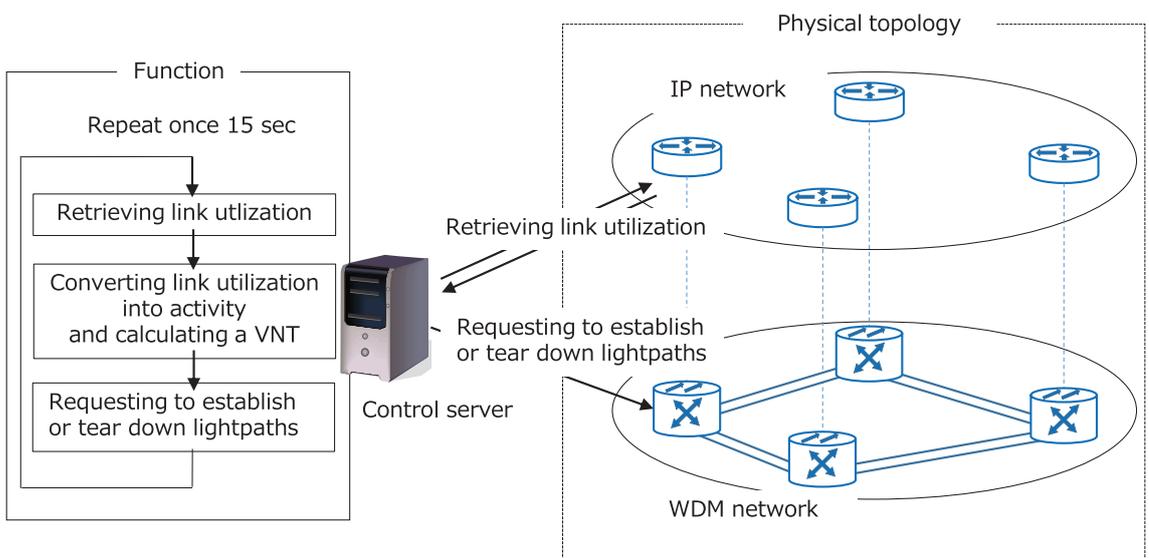


Figure 15: VNT control server

Table 1: Mean measurement error of link utilization ( $\mu$ ) and standard deviation of link utilization ( $\sigma$ ) (averaged over 100 trials)

Measurement time (sec)	$\mu$	$\sigma$
5	0.069	0.084
10	0.072	0.049
15	0.023	0.028
30	0.022	0.026
60	0.014	0.007

### 5.1.3 Calculation of Link Utilization

We calculate link utilization by dividing amount of traffic during a certain period of time by the time. The proposed method can reconstruct VNTs in short interval if it can get feedback in short measurement time. However, too much shorten the time leads to error of link utilization. We investigated the error of link utilization with different measurement time to know the lower bound of the time keeping the error small. Table 1 shows the error of link utilization with different measurement time. We measured a link, on which two 400 Mbps flows were transferred and took the average over 100 trials. As show in the table, the longer the measurement time is, smaller average of error and standard deviation of link utilization are. The error of measurement of 15 sec is small and almost as same as that of measurement of 30 sec whereas the error is big comparatively when the measurement time is 5 sec or 10 sec. Therefore, in following experiments, we set 15 seconds for the time.

When VNT is actually reconfigured, routing table of IP network is re-constructed by OSPF protocol. Traffic is not transferred during the reconstruction. We need to ignore the reconstruction time for calculating link utilization to remove the error since we calculate link utilization by dividing amount of traffic during a certain period of time by the time. We investigated the processing time of re-constructing routing table. We reconfigured VNTs 5 times and analyzed the routing log in router. The log showed that the time for a router to complete to re-construct routing table after VNT was reconfigured was 10.97 sec averaged over five trials. It was almost same as 11.32 sec of the interval when no traffic was transferred. This indicates that re-constructing routing table causes the interval. Figure 2 shows the main processing events and its processing time for completing to

Table 2: Processing time for re-constructing routing table (averaged over five trials)

Event	Processing time (sec)
Establishment of neighbor state	0.83
Selection of designated router	4.06
Re-creation of routing table	6.08
Total time	10.97

re-construct routing table. When a lightpath is established between routers, these routers establish neighbor state in OSPF, which takes about 0.83 sec. After establishing neighbor state, these routers select designated router which needs for exchanging LSA (Link State Advertisement), which takes about 4.06 s. Finally, each router exchanges link state information and re-construct routing table, which takes about 6.08 sec. As shown in Table 2, total time of re-constructing routing table is about 11 sec. Therefore, in our experimental environment, we ignore the reconstruction time, 11 sec, for calculating the link utilization when VNT is actually reconfigured.

## 5.2 Experimental Scenario

We verify the feasibility of the proposed method with the scenario that the propose method recovers the condition after getting the feedback of poor condition, which is caused by environmental changes such as traffic changes or node failures (OXC failures). We explain three scenarios with different environmental changes, 1) traffic change, 2) single OXC failure, 3) multiple OXC failures.

### 5.2.1 Scenario A: Traffic Change

A first scenario involves traffic increase as the environmental changes (Figure 16). Detailed description of Scenario A is,

- (1) at time 5, a 400 Mbps UDP flow starts to be transferred from router 7 to router 5,
- (2) at time 10, a 400 Mbps UDP flow starts to be transferred from router 1 to router 5,
- (3) after (1) and (2), link utilization of the lightpath between router 7 and router 5 becomes 0.8,
- (4) the maximum link utilization is fed back as activity to the proposed method,

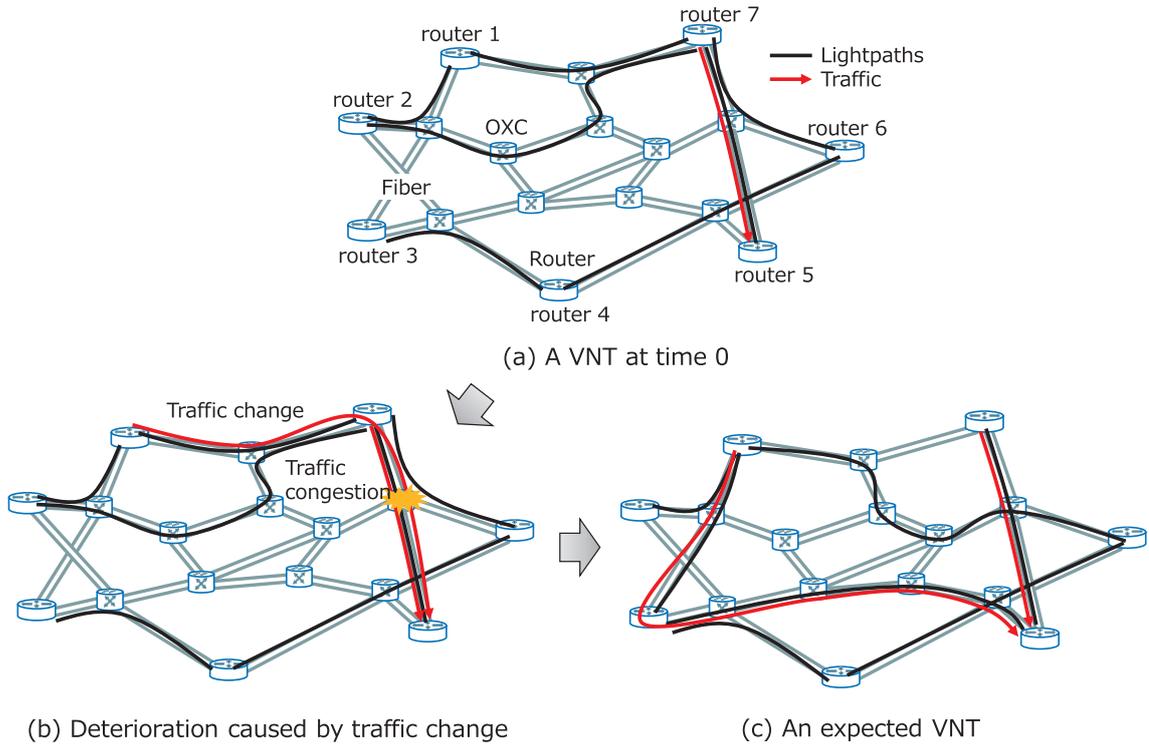


Figure 16: Scenario A: Traffic change

(5) then, the proposed method searches for a VNT which accommodates changed traffic.

We expect that the proposed method converges to the VNT that accommodates two UDP flows. An expected VNT is depicted in Figure 16 (c).

### 5.2.2 Scenario B: Single OXC Failure

A second scenario involves single OXC failure as the environmental changes (Figure 17). We make OXCs be failed by shutting down ports of OXCs through operation console. Detailed description of Scenario B is,

- (1) at time 5, two 400 Mbps UDP flows start to be transferred. The first flow is transferred from router 2 to router 1 and the second flow is transferred from router 4 to router 6,
- (2) at time 10, OXC 1 breaks down,
- (3) the route of the flow from router 2 to router 1 is changed by OSPF protocol,
- (4) after that, link utilization of the lightpath between router 4 and router 6 becomes 0.8,

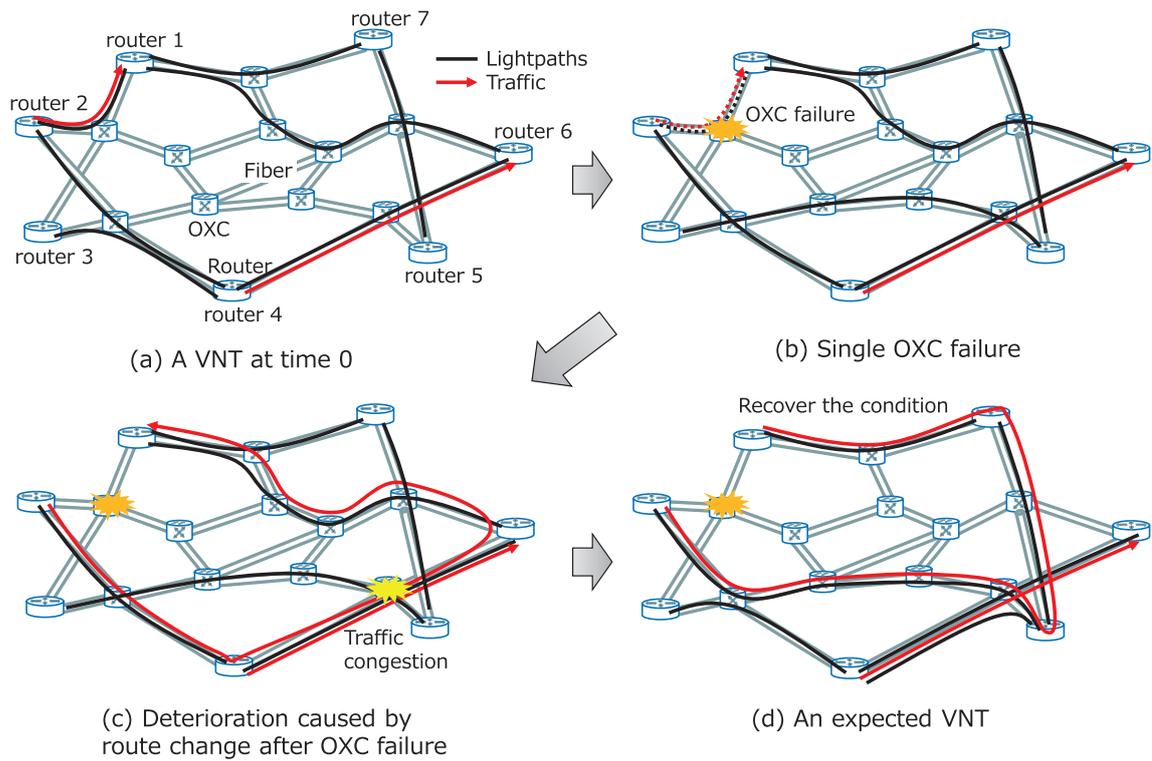


Figure 17: Scenario B: Single OXC failure

- (5) the maximum link utilization is fed back as activity to the proposed method,
- (6) then, the proposed method searches for a VNT which accommodates changed traffic.

We expect that the proposed method converges to the VNT which that accommodates two UDP flows. An expected VNT is depicted in Figure 17 (d).

### 5.2.3 Scenario C: Multiple OXC Failures

A third scenario involves multiple OXC failures as the environmental changes (Figure 18). Detailed description of Scenario C is,

- (1) at time 5, two 400 Mbps UDP flows start to be transferred. The first flow is transferred from router 6 to router 2 and the second flow is transferred from router 7 to router 2,
- (2) at time 10, OXC 6 and OXC 7 break down,
- (3) the route of the two flows from router 6 to router 2 and from router 7 to router 2 is changed by OSPF protocol,

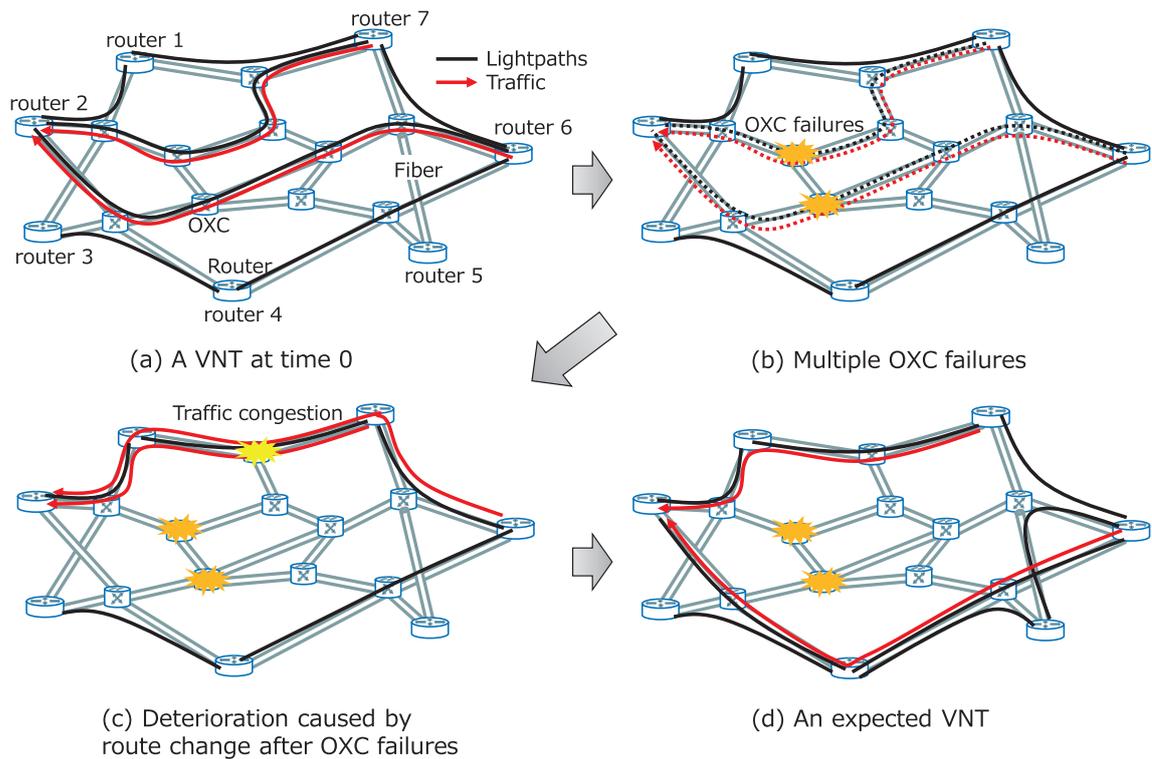


Figure 18: Scenario C: Multiple OXC failures

- (4) after that, link utilization of the lightpath between router 1 and router 2, and router 7 and router 1 becomes 0.8,
- (5) the maximum link utilization is fed back as activity to the proposed method,
- (6) then, the proposed method searches for a VNT which accommodates changed traffic,

We expect that the proposed method converges to the VNT that accommodates two UDP flows. An expected VNT is depicted in Figure 18 (d).

### 5.3 Experimental Results

Figure 19 shows the result of Scenario A. In the figure, horizontal axis represents the elapsed time after starting the experiment. The figure has two vertical axes: “Maximum Link Utilization” and “Discarded Packets [Mbps]”. “Maximum Link Utilization” represents the maximum link utilization and “Discarded Packets [Mbps]” represents the rate of discarded packets in routers. We plot the rate of discarded packets since packets will be discarded if a router do not know the forwarding

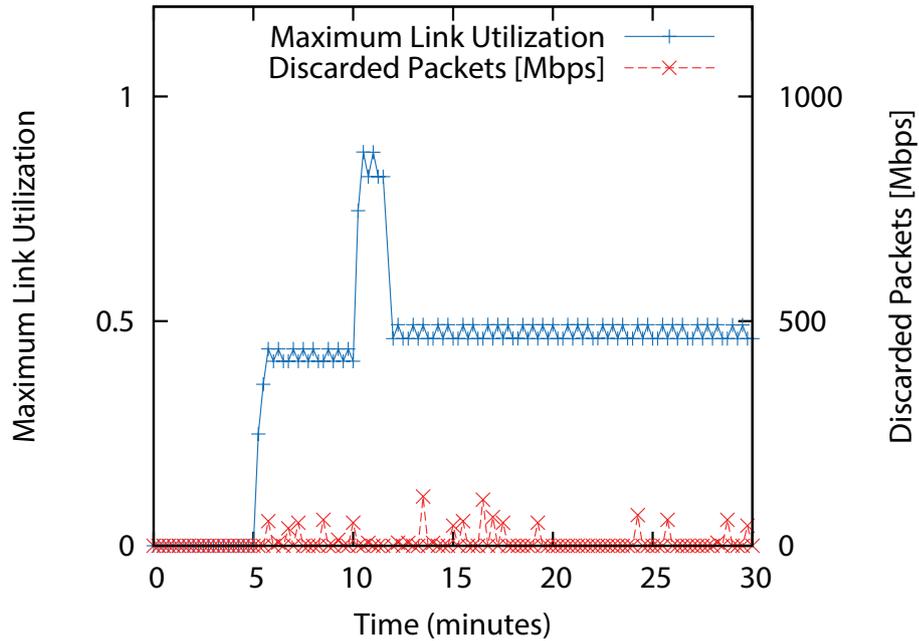


Figure 19: Changes of maximum link utilization: Scenario A

address of the packets on constructed VNT. Note here that although information of discarded packets is easily and quickly retrieved by SNMP in general, in our experiments, we calculate amount of discarded packets by subtracting the amount of packets arriving at a router from the amount of packets departing packets from the router during the measurement interval, which were retrieved by SNMP, since the routers, Juniper MX5 and MX10, do not support SNMP statistic of discarded packets and it takes more time to retrieve the statistic from the router's original management information. As shown in Figure 19, maximum link utilization increases to around 0.4 at time 5, when the proposed method do not search VNTs since less than 0.5 of maximum link utilization is interpreted as good condition of IP network. When traffic change occurs at time 10, the maximum link utilization becomes about 0.8 and the proposed method receives feedback of poor condition of IP network. Then, our method starts to search a suitable VNT for current environment. At time 12, maximum link utilization becomes low and the condition gets recovered. The proposed method converges to the VNT after time 12 as we see that the maximum link utilization is stable after time 12.

When we conducted experiments with Scenario B, we found that a router sometimes lost connectivity with other routers as show in Figure 20. In the Figure 20, at time 5, two UDP flows

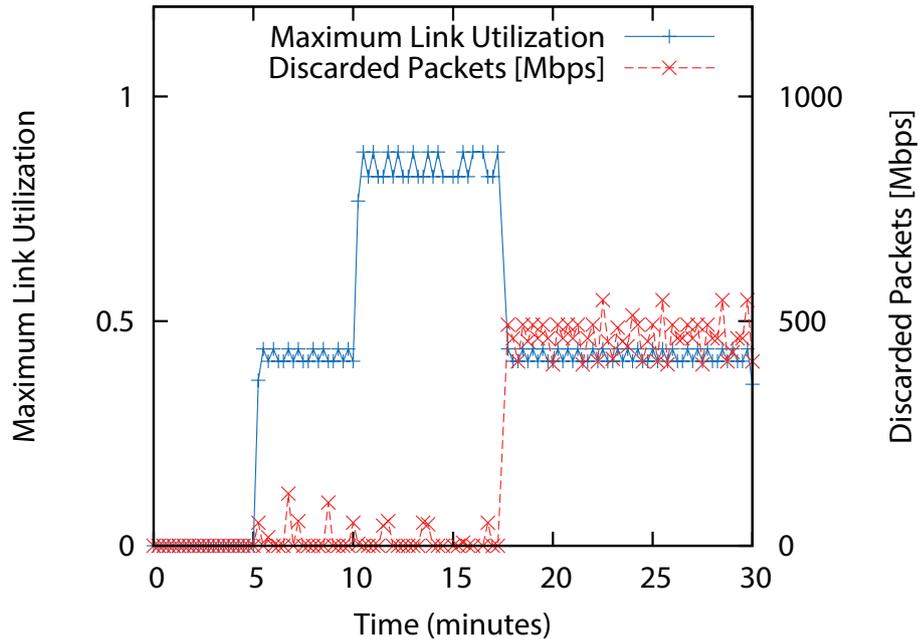


Figure 20: Changes of maximum link utilization when VNT loses connectivity

started to be transferred and the maximum link utilization became around 0.4. At time 10, an OXC broke down, the maximum link utilization became around 0.8 and our method started to search a suitable VNT for current environment. Then, at time 18 the maximum link utilization became around 0.4 and our method converged to the VNT after time 18. However, in the Figure 20, we can see the increase of the rate of discarded packets after time 18. It indicates that some router lost connectivity after time 18 and discarded packets arriving the router. A router may lose connectivity with other routers when failures occur since we assume that failures are not to be detected. Note here that if we assume that failures can be detected, the topology after the failures can be fed into our method. With this case, the packets arriving at the router were discarded, and thereby the maximum link utilization becomes low. Our method may converge to the VNT even though the packets are discarded. To prevent this problem, we additionally use information of the rate of discarded packets to calculate the activity. That is, when the rate of discarded packets exceeds a threshold (100 Mbps) we regard that connectivity between routers is lost in the VNT and set the value of activity to zero.

Figure 21 shows the result of experiment with Scenario B. As shown in Figure 21, maximum link utilization increases to around 0.4 at time 5. When single node failure occurs at time 10, the

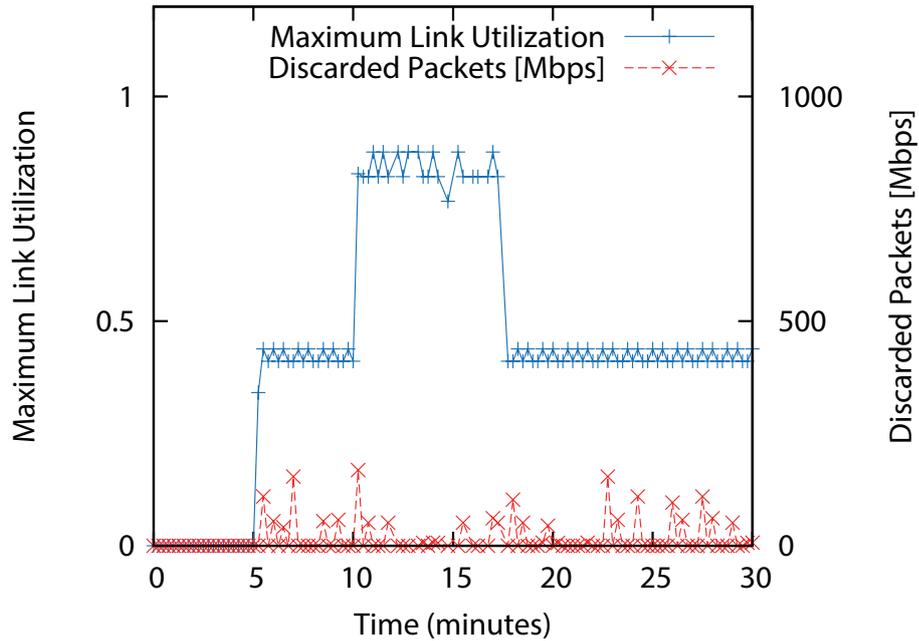


Figure 21: Changes of maximum link utilization: Scenario B (single OXC failure)

maximum link utilization becomes about 0.8 and the proposed method receives feedback of poor condition of IP network. Then, our method starts to search a suitable VNT for current environment. At time 18, maximum link utilization becomes low and the condition gets recovered. The proposed method converges to the VNT after time 12 as we see that the maximum link utilization is stable after time 18.

Figure 22 shows another result of the experiment with Scenario B. In this experiment, we give a different VNT at time 0. As shown in Figure 22, traffic change occurs at time 5 and maximum link utilization increases to about 0.4. When node failure occurs at time 10, maximum link utilization becomes about 0.8 and the proposed method starts to search a suitable VNT for current environment. At time around 17, the maximum link utilization gets low while the rate of discarded packets increased. The increase of the rate of discarded packets is fed back, activity is set to zero and the proposed method search a suitable VNT for current environment. Though the maximum link utilization increases again at time around 22 as the result of searching, the rate of discarded packets decreases and the maximum link utilization becomes around 0.4 at time around 24.

Figure 23 shows the result of experiment with Scenario C. The result is similar to Figure 22.

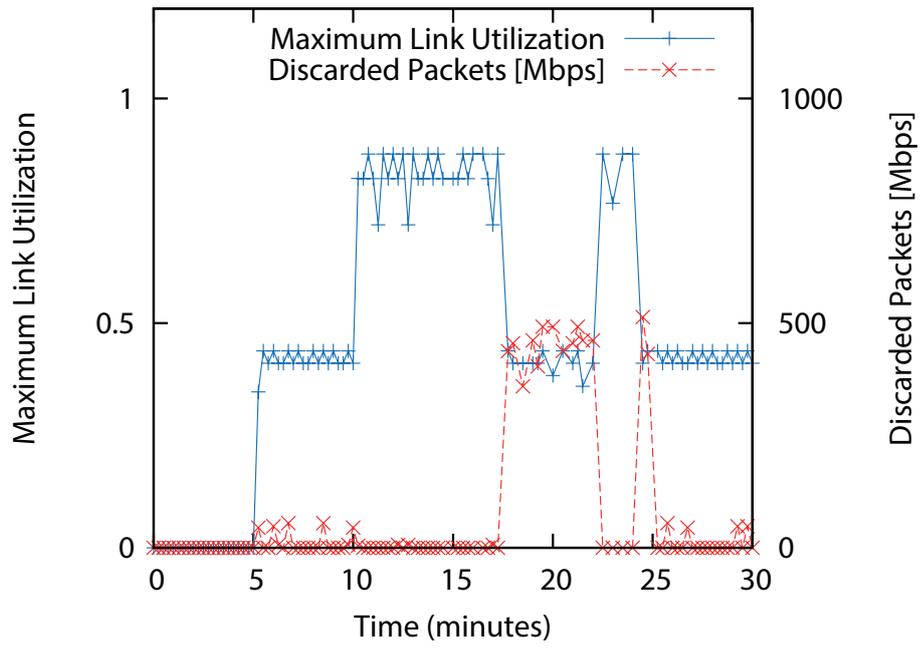


Figure 22: Changes of maximum link utilization: Scenario B. Activity calculation is modified.

After deterioration of network condition at time 10, the proposed method searches a suitable VNT for current environment and finally finds the VNT at time around 23 and converges to the VNT.

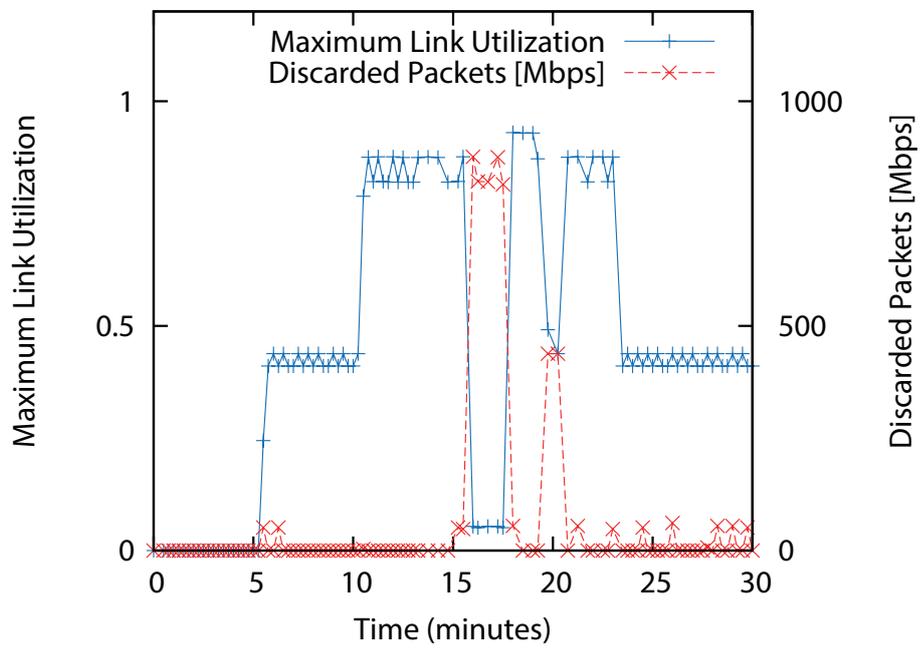


Figure 23: Changes of maximum link utilization: Scenario C (multiple OXC failures). Activity calculation is modified.

## 6 Conclusion

In this paper, we proposed a VNT control method based on attractor selection in WDM networks that is an extension of our previous method. Our aim was to apply our method to larger networks. The computational complexity of our proposed method was reduced to  $O(N^3)$  from the  $O(N^4)$  of the previous method. The proposed method could use parallel computing, and the calculation time was reduced to  $O(N^2)$ . Simulation results show that the proposed method calculates VNTs very quickly. The calculation time was 0.03 sec for a 1000-node network. Our proposed method also showed adaptability to various patterns of traffic demand and achieved the control objective at a rate similar to that of our previous method. However, our proposed method sometimes failed to achieve the objective due to excessively high traffic demand. Our proposed method keeps searching for a suitable VNT until it is able to converge to one, and we leave as future work the problem of dealing with long iterations of reconfiguration. We also verified the feasibility of the proposed method through experiment with seven-router network. When we conducted experiments with OXC failures, we found that a router sometimes lost connectivity with other routers. With this case, the packets arriving at the router were discarded, and thereby the maximum link utilization became low, i.e., activity became high. To prevent this, when the rate of discarded packets exceeds a threshold, we regard that connectivity between routers is lost in the VNT and set the value of activity to zero. The results of the experiments show that the proposed method recovers the network condition after deterioration caused by environmental changes such as traffic changes or node failures. Simultaneously, we got knowledge of necessity of considering the effect of VNT reconstruction on routing protocol and the error of the network information we retrieve from the network.

## **Acknowledgement**

I would like to express my sincere gratitude to my supervisor, Professor Masayuki Murata of Osaka University, for his innumerable help, continuous support, encouragement and tremendous advice . I especially would like to express my deepest appreciation to my supervisor, Associate Professor Shin'ich Arakawa of Osaka University for his elaborated guidance and invaluable discussion that made my research life unfogettable. Without his support, all of my work would not have achieved. I would like to acknowledge and thank Assistant Professor Yuki Koizumi of Osaka University for his valuable cooperation in my experiments. He supported my work and gave me valuable help and guidance for using experimental equipments. I am grateful to Assistant Professor Yuichi Ohsita of Osaka University for his helpful comments and feedback. I also feel thankful Mr. Takuya Iwai, Mr. Naotaka Onzuka and Mr. Yoshinobu Shijo. They helped me set up experimental equipments and gave me invaluable advice for my experiments. Finally, I truly thank my friends and colleagues in the Department of Information Networking, Graduate School of Infomation Science and Technology of Osaka University for their support.

## References

- [1] Y. Liu, H. Zhang, W. Gong, and D. Towsley, "On the interaction between overlay routing and underlay routing," in *Proceedings of IEEE INFOCOM*, vol. 4, pp. 2543–2553, Mar. 2005.
- [2] Y. Koizumi, S. Arakawa, and M. Murata, "Stability of virtual network topology control for overlay routing services," *Journal of Optical Networking*, vol. 7, pp. 704–719, July 2008.
- [3] S. Huang, M. Xia, C. U. Martel, and B. Mukherjee, "A multistate multipath provisioning scheme for differentiated failures in telecom mesh networks," *Journal of Lightwave Technology*, vol. 28, pp. 1585–1596, Oct. 2010.
- [4] M. F. Habib, M. Tornatore, F. Dikbiyik, and B. Mukherjee, "Disaster survivability in optical communication networks," *Computer Communications*, vol. 36, pp. 630–644, Jan. 2013.
- [5] J. Wu, "A survey of WDM network reconfiguration: Strategies and triggering methods," *Computer Networks*, vol. 55, pp. 2622–2645, May 2011.
- [6] F. Xu, M. Peng, A. Rayes, N. Ghani, and A. Gumaste, "Multi-failure post-fault restoration in multidomain DWDM networks," in *Proceedings of Optical Fiber Communication Conference and Exposition*, pp. 1–3, Mar. 2011.
- [7] M. Habib, M. Tornatore, M. De Leenheer, F. Dikbiyik, and B. Mukherjee, "Design of disaster-resilient optical datacenter networks," *Journal of Lightwave Technology*, vol. 30, pp. 2563–2573, Aug. 2012.
- [8] Y. Xin, G. Rouskas, and H. Perros, "On the physical and logical topology design of large-scale optical networks," *Journal of Lightwave Technology*, vol. 21, pp. 904–915, Apr. 2003.
- [9] J. Li, G. Mohan, E. Tien, and K. Chua, "Dynamic routing with inaccurate link state information in integrated IP-over-WDM networks," *Computer Networks*, vol. 46, pp. 829–851, Dec. 2004.
- [10] T. Ye, Q. Zeng, Y. Su, L. Leng, W. Wei, Z. Zhang, W. Guo, and Y. Jin, "On-line integrated routing in dynamic multifiber IP/WDM networks," *Journal on Selected Areas in Communications*, vol. 22, pp. 1681–1691, Nov. 2004.

- [11] S. Arakawa, M. Murata, and H. Miyahara, "Functional partitioning for multi-layer survivability in IP over WDM networks," *IEICE Transactions on Communications*, vol. E83-B, pp. 2224–2233, Oct. 2000.
- [12] N. Ghani, S. Dixit, and T. Wang, "On IP-over-WDM integration," *Communications Magazine*, vol. 38, pp. 72–84, Mar. 2000.
- [13] M. Kodialam and T. Lakshman, "Integrated dynamic IP and wavelength routing in IP over WDM networks," in *Proceedings of IEEE INFOCOM*, pp. 358–366, Apr. 2001.
- [14] J. Comellas, R. Martínez, J. Prat, V. Sales, and G. Junyent, "Integrated IP/WDM routing in GMPLS-based optical networks," *Network Magazine*, vol. 17, pp. 22–27, Mar./Apr 2003.
- [15] K. Lee and M. Shayman, "Rollout algorithms for logical topology design and traffic grooming in multihop WDM networks," in *Proceedings of IEEE GLOBECOM*, vol. 4, Dec. 2005.
- [16] Q. Rahman, A. Sood, Y. Aneja, S. Bandyopadhyay, and A. Jaekel, "Logical topology design for WDM networks using tabu search," in *Proceedings of the 13th International Conference on Distributed Computing and Networking*, pp. 424–427, Jan. 2012.
- [17] 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*, pp. 57–65, June 2002.
- [18] R. Ramaswami and K. N. Sivarajan, "Design of logical topologies for wavelength-routed optical networks," *Journal on Selected Areas in Communications*, vol. 14, pp. 840–851, June 1996.
- [19] E. Leonardi, M. Mellia, and M. A. Marsan, "Algorithms for the Logical Topology Design in WDM All-Optical Networks," *Optical Networks*, vol. 1, pp. 35–46, 2000.
- [20] D. Banerjee and B. Mukherjee, "Wavelength-routed optical networks: Linear formulation, resource budgeting tradeoffs, and a reconfiguration study," *Transactions on Networking*, vol. 8, pp. 598–607, Apr. 2000.

- [21] F. Dikbiyik, A. S. Reaz, M. De Leenheer, and B. Mukherjee, "Minimizing the disaster risk in optical telecom networks," in *Proceedings of Optical Fiber Communication Conference and Exposition*, pp. 1–3, Mar. 2012.
- [22] C. C. Meixner, F. Dikbiyik, M. Tornatore, C. N. Chuah, and B. Mukherjee, "Disaster-resilient virtual-network mapping and adaptation in optical networks," in *Proceedings of Optical Network Design and Modeling*, pp. 107–112, Apr. 2013.
- [23] Y. Koizumi, T. Miyamura, S. Arakawa, E. Oki, K. Shiimoto, and M. Murata, "Adaptive virtual network topology control based on attractor selection," *Journal of Lightwave Technology*, vol. 28, pp. 1720–1731, June 2010.
- [24] Y. Minami, Y. Koizumi, S. Arakawa, T. Miyamura, K. Shiimoto, and M. Murata, "Benefits of virtual network topology control based on attractor selection in WDM networks," *International Journal on Advances in Internet Technology*, vol. 4, pp. 79–88, Sept. 2011.
- [25] Y. Baram, "Orthogonal patterns in binary neural networks," *NASA Technical Memorandum No. 100060*, Mar. 1988.
- [26] A. Nucci, A. Sridharan, and N. Taft, "The problem of synthetically generating IP traffic matrices: initial recommendations," *Computer Communication Review*, vol. 35, no. 3, pp. 19–32, 2005.
- [27] A. Callado, C. Kamienski, G. Szabó, B. Gero, J. Kelner, S. Fernandes, and D. Sadok, "A survey on internet traffic identification," *Communications Surveys & Tutorials*, vol. 11, pp. 37–52, 2009.
- [28] Y. Zhang, M. Roughan, N. Duffield, and A. Greenberg, "Fast accurate computation of large-scale IP traffic matrices from link loads," in *Proceedings of ACM SIGMETRICS*, vol. 31, pp. 206–217, June 2003.
- [29] S. Kamamura, Y. Koizumi, D. Shimazaki, T. Miyamura, S. Arakawa, K. Shiimoto, A. Hiramatsu, and M. Murata, "Attractor selection-based virtual network topology control with dynamic threshold reconfiguration for managed self-organization network," in *Proceedings of the 24th International Teletraffic Congress*, pp. 1–6, Sept. 2012.