Scalable Design Method of Attractors in Noise-Induced Virtual Network Topology Control

Toshihiko Ohba, Shin'ichi Arakawa, Yuki Koizumi, and Masayuki Murata

***Abstract*—One approach to accommodating IP traffic on a wavelength-routed network is to construct a virtual network topology (VNT) and reconfigure the VNT according to traffic changes. Our research group has proposed a VNT control method based on attractor selection, which is a model of the behavior by which living organisms adapt to unknown changes in their surrounding environment. Guided by attractors, the VNT control method searches for a solution: a VNT that can accommodate the IP traffic. Attractors are a subset of the equilibrium points in the solution space and correspond to VNT candidates. It is crucial to design the attractors properly, since they define the attractive states in the VNT control. If the VNT candidates are not designed properly, it takes a long time for the method to find a solution. This paper therefore proposes a method for designing VNT candidates. Our basic approach is to prepare VNT candidates in which the bottleneck links (lightpaths) are different from each other. However, our exhaustive algorithm based on this approach has the problem of requiring large amounts of computational time for large-scale networks. We therefore propose a method that hierarchically contracts a network topology to enable application of our algorithm to large-scale networks. Evaluation results show that the VNT control method using attractors obtained by our method achieves shorter convergence times.**

***Index Terms*—Attractor selection; Virtual network topology; VNT control; Wavelength division multiplexing; Wavelength-routed network.**

I. 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. Much research has investigated methods for accommodating IP traffic over WDM networks [1–3]. IP over WDM networks consist of two layers, the WDM network and the IP network (Fig. 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 topology (VNT) 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 controlling the VNT, that is, by reconfiguring the VNT according to the traffic changes such that the VNT can accommodate the changing traffic demand.

Many studies have been devoted to developing methods for accommodating IP traffic on a VNT according to traffic changes [4–6]. References [4–6] propose methods for configuring a VNT by solving a mixed integer linear program, which aims to minimize the maximum link utilization of the VNT, the packet delay, or the number of resources that form the VNT (e.g., wavelengths and router ports). These methods use information in the form of traffic demand matrices. The existing heuristic approaches such as the Increasing Multi-hop Logical Topology Design Algorithm (I-MLTDA) and the Minimum-delay Logical topology Design Algorithm (MLDA) [7] also use the information of traffic demand. However, it generally takes a long time to retrieve the traffic demand information matrix. The methods proposed in Refs. [4–7], therefore, reconfigure the VNT based on long-term measurements of traffic demand.

However, when traffic demand fluctuates rapidly, it is difficult for methods that design the VNT based on traffic demand matrices to reconfigure the VNT 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. [8,9] 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. [10]. This method reconfigures the VNT 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 VNT to be able to accommodate changing traffic demand. It is therefore important to devise a method for

Manuscript received March 19, 2015; revised July 2, 2015; accepted July 14, 2015; published August 5, 2015 (Doc. ID 236183).

The authors are with the Graduate School of Information Science and Technology, Osaka University, Suita, Osaka 565-0871, Japan (e-mail: t-ohba@ist.osaka-u.ac.jp).

http://dx.doi.org/10.1364/JOCN.7.000851

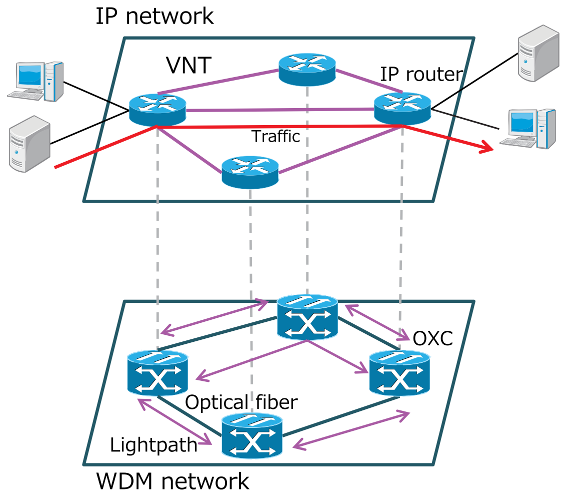

Fig. 1. IP over WDM network.

adaptively reconfiguring the VNT in response to traffic changes that occur over a short period of time.

Our research group has proposed a VNT control method that is adaptive to traffic changes and accommodates IP traffic effectively [11,12]. This method is based on a dynamical system called the attractor selection model, which models the behavior by which living organisms adapt to unknown changes in their surrounding environment and recover their condition. In our VNT control method, attractors, which are a subset of the equilibrium points in the solution space, correspond to VNT candidates. The basic mechanism in this VNT control comprises deterministic and stochastic behaviors. Here, the deterministic behavior represents a VNT control directed to attractors, and the stochastic behavior represents a randomized factor to change VNT configurations. These behaviors are controlled by feedback indicating a condition of the IP network. Here, the condition can be a communication quality of the IP network such as, for example, maximum utilization of virtual links. That is, our VNT control method does not collect the traffic demand matrices, but instead collects the condition of the network and controls the VNT 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 min or less, than the traffic demand matrices used by existing VNT control methods. When there are large fluctuations in traffic demand, our VNT control method searches for a solution: a VNT 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. [11,12] that this method can reconfigure a VNT adaptively in response to fluctuations in the network environment such as traffic changes and node failure.

In our VNT control method based on attractor selection, it is crucial to design attractors properly, since attractors define the attractive states of the VNT control. In Refs. [11,12], we used randomly generated VNT candidates. However, if the VNT candidates are not designed properly, it takes a long time for the VNT control method to find a solution. For example, assuming that VNT candidates are tuned for only certain patterns of traffic demand, our VNT control 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 VNT candidates as attractors. However, the number of VNT candidates that can be kept as attractors is limited to 10%–15% of the number of possible lightpaths according to the properties of the Hopfield network [13]. The Hopfield network is one well-known model for storing and reading bit patterns using a weighted matrix. So, we use the properties of the Hopfield network to estimate the limitation. We therefore propose a new method for designing VNT candidates. Our approach classifies various VNT candidates in 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 that 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 VNT candidates that can accommodate various patterns of traffic demand, various kinds of VNTs can be searched by the attractor selection. Our VNT control method can thus find a solution within a short period of time. In other words, this makes our VNT control method more adaptive to traffic changes.

The rest of this paper is organized as follows. In Section II, we explain our VNT control method based on attractor selection. We then propose a method for designing VNT candidates and evaluating our algorithm in Section III. We also propose a method for hierarchically designing VNT candidates for large-scale networks in Section IV. In Section V, we evaluate the method for hierarchically designing VNT candidates and the VNT control method using the attractors obtained by this method. We conclude this paper in Section VI.

II. VNT Control Based on Attractor Selection

We will start by explaining the VNT control method based on attractor selection proposed in [11,12]. In this paper, traffic is assumed to flow between IP routers via the shortest path of the VNT. We refer to link utilization on the VNT simply as link utilization.

A. Overview of VNT Control Based on Attractor Selection

Dynamic systems that are driven by the attractor selection model adapt to unknown changes in their surrounding environments [14]. In the attractor selection model, 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 model comprises both deterministic behavior and stochastic behavior. The behavior of a dynamic system driven by attractor selection is described as follows:

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

The state of the system is represented by $\mathbf{x} = (x_1, \ldots, x_i, \ldots, x_n)$ (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 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 VNT control method considers the state of the system $\mathbf{x}$ as the state of all possible lightpaths that form the VNT and uses the condition of the IP network as the activity. Our VNT control method then configures the VNT so that the condition of the IP network improves when the condition of the IP network becomes poor due to fluctuations in traffic demand.

B. Dynamics of VNT Control

Our VNT control 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

$$\frac{dx_i}{dt} = \alpha \cdot \left(\varsigma\left(\sum_j W_{ij} x_j \right) - x_i \right) + \eta. \tag{2}$$

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, and 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), 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.

1) Activity: Our VNT control 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 VNT control methods. We convert the maximum link utilization on the IP network, $u_{\max}$, into the activity α using Eq. (3) below. The activity is in the range of [0, γ]. The constant number θ is the threshold for the VNT control. When the maximum link utilization is more than the threshold θ, the activity rapidly approaches zero and our VNT control method searches for a solution that improves the condition of the IP network. The constant δ determines the gradient of the function. The conversion equation is

$$\alpha = \frac{\gamma}{1 + \exp(\delta \cdot (u_{\max} - \theta))}. \tag{3}$$

2) Regulatory Matrix: We set the regulatory matrix so that it has a set of VNT candidates as attractors. That is, we set the regulatory matrix $\mathbf{W}$ so that $d\mathbf{x}/dt$ in Eq. (1) is equal to zero when the VNT reconfigured by our VNT control method $\mathbf{x} = (x_1, \ldots, x_i, \ldots, x_n)$ is one of the attractors. To store attractors in the regulatory matrix, we use a method to decide the regulatory matrix using the pseudoinverse matrix, which is shown in Ref. [15]. Specifically, assuming that we set m VNT candidates as attractors and one of the candidates is represented by $\mathbf{x}^{(k)} = (x_1^{(k)}, \ldots, x_i^{(k)}, \ldots, x_n^{(k)}) (1 \le k \le m)$, the regulatory matrix that has m attractors is

$$\mathbf{W} = \mathbf{X}^{+}\mathbf{X}, \tag{4}$$

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

III. Attractor Design Method

A. Attractor Design Problem

We suppose that we are designing attractors (i.e., VNT candidates) for a network with n nodes. Although the size of the solution space is 2^{n^2}, the number of VNT candidates that can be kept as attractors is limited to about 10%–15% of the number of possible lightpaths, n^2 [13]. Moreover, VNT candidates should be diverse enough to allow them to adapt to various fluctuations in traffic demand. Therefore, the problem in properly designing the VNT candidates to use as attractors comes down to the problem of selecting $0.1n^2$ VNT candidates that have a wide diversity from within the solution space.

For this problem, we focus on the characteristics of the VNT candidates. Since the VNT configured by our VNT control 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 VNT candidates. It is important that the attractors have different characteristics in order to produce a diverse range of VNT candidates. We therefore take the approach of classifying VNT candidates in groups based on their characteristics and selecting one attractor from each of the VNT candidate groups. By preparing a limited number of VNT candidates with diverse characteristics, attractor selection is able to search various kinds of VNTs. As a result, our VNT control method finds a solution quickly, making our VNT control method more adaptive to traffic changes.

B. Exhaustive Algorithm for Designing Attractors With Different Characteristics

We develop an algorithm that selects attractors for our VNT control 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 VNT candidates of the VNT g.
2) Classify the enumerated VNT candidates based on their characteristics.
3) Select an attractor from each group of VNT candidates.

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

1) Enumeration of VNT Candidates: We enumerate isomorphic VNTs of g. The isomorphic VNTs are generated by exchanging all the nodes of the VNT g. Figure 2 illustrates examples of isomorphic VNTs. In Fig. 2, the VNT g_1 consists of five nodes $N0$, $N1, \ldots, N4$ and the VNTs g_2 and g_3 are isomorphic VNTs of g_1. The isomorphic VNT g_2 is generated by shifting $N0$ of the VNT g_1 to $N1$, $N1$ to $N2$, $N2$ to $N3$, $N3$ to $N4$, and $N4$ to $N0$. The isomorphic VNT g_3 is generated by shifting $N0$ of the VNT g_1 to $N4$, $N4$ to $N3$, $N3$ to $N2$, $N2$ to $N1$, and $N1$ to $N0$. However, VNT 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 VNT candidates is at most $n!$.

In Fig. 2, we assume that the VNT 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 double-lined link between nodes $N3$ and $N4$. Since the VNT g_1 is configured by a heuristic method based on the traffic demand matrix T_1, the VNT 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, … , 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 VNT g_1 to $N1$, $N1$ to $N2$, $N2$ to $N3$, $N3$ to $N4$, and $N4$ to $N0$ to generate the isomorphic VNT g_2, the traffic load on the link between the nodes $N2$ and $N3$ becomes the highest and the VNT g_2 can accommodate T_2. That is, it is expected that any of the isomorphic VNTs 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 VNT g and the enumerated VNT candidates.

2) Classification of VNT Candidates: We classify the VNT candidates that belong to G in 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 VNT candidates. We then classify VNT candidates that have different bottleneck links from each other into different groups, as shown in Fig. 3. The bottleneck link is the link that has the largest value of edge betweenness centrality among the links that form a VNT candidate. When each of the VNT candidates that have been selected as attractors have different bottleneck links, it is expected that any of the VNT candidates selected as attractors will be able to accommodate various patterns of traffic demand. Note that in our VNT control 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 VNT candidates that have the same bottleneck links in the same group. The following gives a formal definition of the VNT candidate groups.

- $p = (s, d)$: 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 VNT candidate g_i

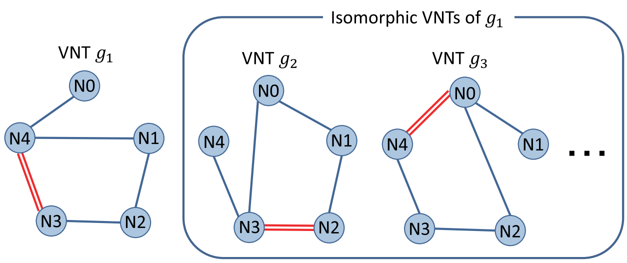

Fig. 2. Example of isomorphic VNTs.

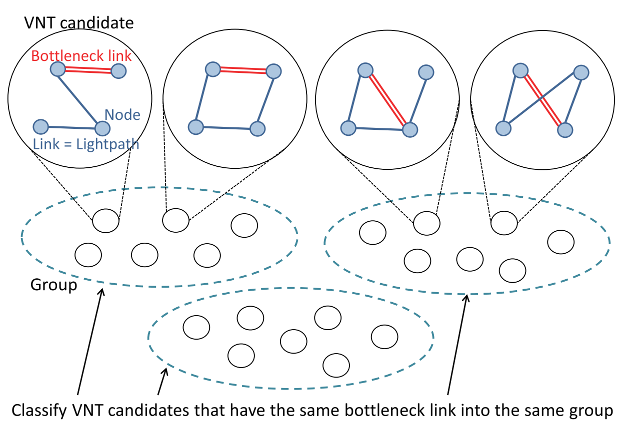

Fig. 3. Classification of VNT candidates.

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

$$G_p = \{g_i | g_i \in G, C(g_i, l_p) = \max_q C(g_i, l_q)\}. \qquad (5)$$

In this way, we divide the set of VNT candidates enumerated in Subsection III.B.1. The number of groups is at most n^2, since the number of possible lightpaths is n^2. However, since the number of VNT candidates that can be kept as attractors is $0.1n^2$, we further merge the VNT candidate groups.

We merge VNT 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. Figure 4 illustrates the condition used to merge VNT 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 VNT candidates that belong to groups $G_{(s,a)}$ and $G_{(a,d)}$ as having similar characteristics. Based on this heuristic, we merge the VNT candidates groups as follows:

$$G_{(s,d)} \leftarrow G_{(s,a)} \cup G_{(a,d)} \cup G_{(s,d)}, \qquad (6)$$

where the degree of node a is low. In Eq. (6), we also treat the VNT candidates in group $G_{(s,d)}$, which has the bottleneck link $l_{(s,d)}$, as having similar characteristics to groups $G_{(s,a)}$ and $G_{(a,d)}$. The reason is that it is likely that the traffic load on link $l_{(s,d)}$ is high when links $l_{(s,a)}$ and $l_{(a,d)}$ are bottleneck links. We select nodes a, s, and d in ascending order of degree, since the correlation of the traffic loads on links $l_{(s,a)}$ and $l_{(a,d)}$ is high when the degree of node a is low. However, since each group has different VNT candidates, we select nodes a, s, and d based on the average value of degree among all the VNT candidates in the group. We repeatedly merge the VNT candidate groups until the number of VNT candidate groups is about $0.1n^2$.

3) Selection of Attractors From VNT Candidate Groups: We finally select one attractor from each of the VNT candidate groups. We select the VNT candidate that has the lowest maximum value of edge betweenness centrality among the VNT candidate 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.

C. Effect of Design Approach: Engineered or Random

In this section, we evaluate the performance of the VNT candidates obtained by the algorithm described in Subsection III.B. 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 VNT candidates using I-MLTDA [7] as the heuristic method with a traffic demand matrix with elements that follow a log-normal distribution. We obtain 10 VNT candidates by following the algorithm in Subsection III.B. For the evaluation, we use 1000 patterns of traffic demand between each node pair according to a lognormal distribution. We compare this to a method that constructs VNT candidates by establishing lightpaths between node pairs chosen in a uniformly random manner. This is because we used randomly generated VNT candidates in Refs. [11,12]. The number of VNT candidates is the same for all methods.

Figure 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 VNT candidates for each traffic pattern. The horizontal axis shows the maximum link utilization of the VNT candidates by our method, and the vertical axis shows that of the VNT candidates by the method for comparison. In Fig. 5, we can see that the maximum link utilization of the VNT candidates by our method is lower for more traffic patterns than the other method. Specifically, the VNT 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 VNT candidates that reduce the traffic load for a wider variety of traffic demand than

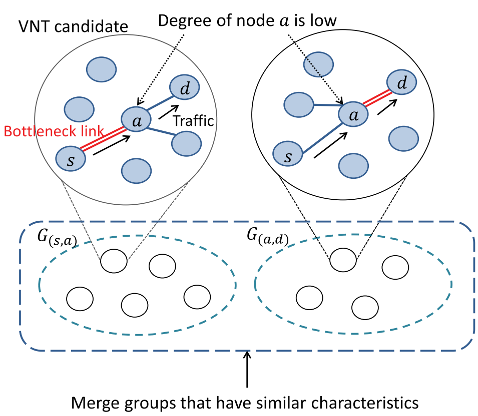

Fig. 4. Merging of VNT candidate groups.

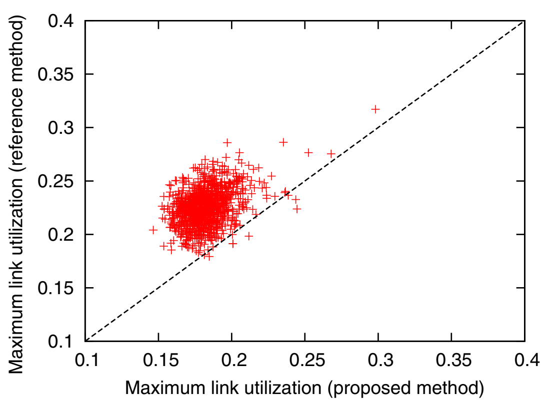

Fig. 5. Distribution of maximum link utilization: 10 nodes, 5 ports.

the other method. Thus, our approach allows us to design better attractors by classifying VNT candidates in groups based on their characteristics and selecting an attractor from each group.

IV. Scalable Design Method of Attractors

A. Scalability Problem of the Exhaustive Algorithm

Although we can design better attractors based on our approach, as shown in Subsection III.C, the algorithm described in Subsection III.B requires a large amount of computational time for large-scale networks. This is because the number of enumerated VNT candidates increases explosively as the number of nodes n increases. Using an ordinary PC, we can design VNT candidates for 10-node networks within 10 min of calculation. However, the calculation time increases exponentially as the number of nodes increases. The calculation time is 2 h for an 11-node network, and 24 h for a 12-node network. We therefore take the approach of contracting the physical network topology and applying the algorithm in Subsection III.B to the contracted network topology. Specifically, we divide the physical network topology into clusters where each cluster has several nodes, as shown in Fig. 6. We reduce the number of nodes in the network topology by treating the clusters as nodes, and apply the algorithm in Subsection III.B to the contracted network topology.

B. Algorithm for Designing Attractors Hierarchically

This section gives an outline of our method for designing VNT candidates hierarchically (see Fig. 7).

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

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

Step 3. Construct VNT candidates in the upper layers by following the algorithm in Subsection III.B.

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

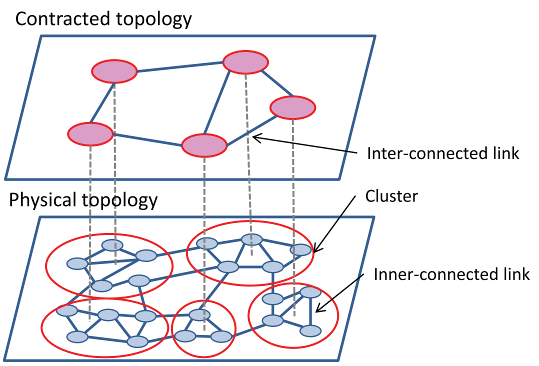

Fig. 6. Contraction of the physical network topology.

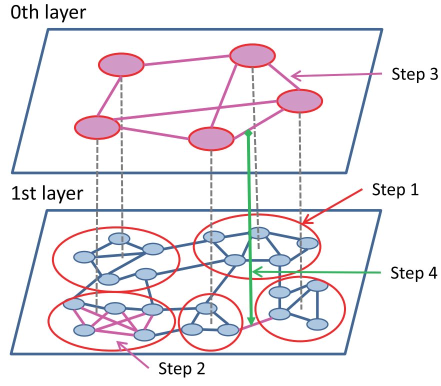

Fig. 7. Outline of the method for designing attractors hierarchically.

We explain the details of the algorithm below.

Step 1. Cluster division of a physical network topology:

We divide the physical network topology into c clusters. When the number of vertices in a cluster is more than c, we divide the cluster into clusters recursively until the number of vertices 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. 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 between clusters are sparsely connected by optical fibers.

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

We construct VNT candidates inside clusters in the bottom layer. We construct a VNT candidate that has a full-mesh topology or a star topology with several hub

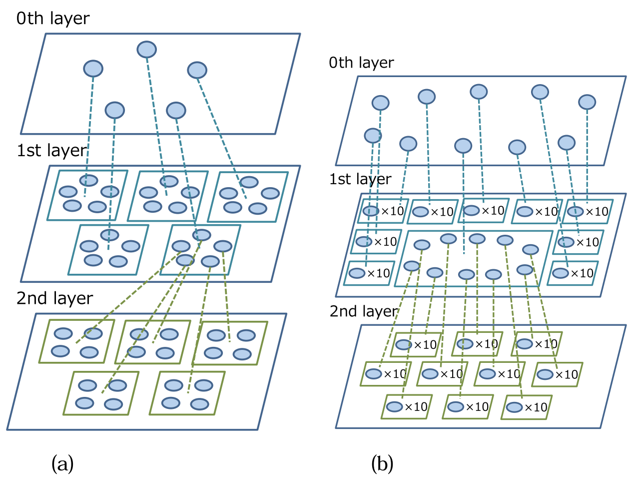

Fig. 8. Clusters in the networks. (a) 100-node network and (b) 1000-node network.

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 VNT candidates in upper layers:

We design VNT candidates with a diverse range of characteristics in the upper layer, which has c nodes (clusters), by following the algorithm in Subsection III.B. However, we do not apply the merge procedure of the algorithm, since the number of VNT 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 VNT candidate groups is at most $c(c-1)/2$, which is the number of the combination of nodes in the upper layer. We construct VNT candidates in the upper layers as follows.

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

Step 3-2. Classify the enumerated VNT 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 VNT 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

$$P_{u,v} = (k_u k_v)^{-1}. \tag{7}$$

Equation (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.

V. Evaluation of Scalable Design Method

A. Performance of VNT Candidates Obtained by the Scalable Design Method

In this section, we evaluate the performance of the VNT candidates obtained by the method in Section IV. We first consider a 100-node network where each node has 32 router ports. We consider the three-layer network shown in Fig. 8(a), where clusters in the same layer have the same number of nodes. This is because we intend 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 VNT candidates. The reason why the number of VNT candidates is seven is that the enumerated VNT candidates are classified into seven groups; only seven lightpaths become bottleneck links among the enumerated VNT candidates. In the middle layer, since there are seven VNT candidates in each cluster and the number of clusters is five, the maximum number of VNT candidates becomes 7^5, counting all combinations. However, the number is too large to be kept as attractors. Therefore, we use the same VNT candidate in all clusters in the middle layer. As a result, we take seven VNT candidates for the middle layer. The VNT 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 VNT candidates in the top layer, seven VNT candidates in the middle layer, and one VNT candidate in the bottom layer. Finally, we obtain 49 VNT candidates.

Table I shows the number of ports used in each layer in the VNT candidates. In the top layer, ports are used to establish lightpaths between nodes that belong to different clusters in the middle layer. Each of the VNT 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 VNT 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 middle layer. In the bottom layer, ports are used to establish lightpaths between nodes inside the same clusters. In the bottom layer, there are 25 clusters, and each cluster consists of four nodes. The VNT candidates in a cluster in the bottom layer have a full-mesh

TABLE I
Number of Ports Used in Each Layer: 100-Node Network

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

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 1000 patterns of traffic demand between each node pair according to a log-normal distribution. We assume that the transmission capacity of all lightpaths is equal throughout 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 Subsection III.C. We make a comparison with the method of constructing VNT candidates by establishing lightpaths between node pairs chosen in a uniformly random manner, because we used randomly generated VNT candidates in Refs. [11,12]. The number of VNT candidates is the same for all methods.

Figure 9 shows the distribution of maximum link utilization for each traffic pattern. This figure shows the lowest value of maximum link utilization among the VNT candidates for each traffic pattern. In Fig. 9, we can see that the maximum link utilization of the VNT candidates by our method is less than that of the VNT candidates by the other method for all traffic patterns. This shows that the method in Section IV can design VNT candidates that better reduce traffic loads in response to various traffic demand conditions compared with the other method. That is, the method in Section IV can design better VNT candidates than the other method for larger networks where we cannot apply the algorithm in Subsection III.B.

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. 8(a), each node in the top layer has 20 routers, while each node in the middle layer has four routers. Thus, traffic loads on links in upper layers become higher in general. Actually, in obtaining Fig. 9, we used more lightpaths in the upper layers, as shown in Table I. 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, VNT candidates obtained by our method can accommodate traffic effectively.

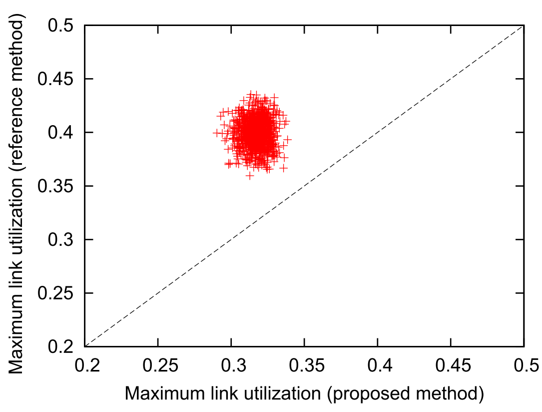

Fig. 9. Distribution of maximum link utilization: 100 nodes, 32 ports.

Further, we evaluate the performance of VNT candidates when applying our method to larger-scale networks. We use a 1000-node network where each node has 64 router ports. We consider the three-layer network shown in Fig. 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 VNT candidates. In the middle layer, since there are 26 VNT candidates in each cluster and the number of clusters is 10, the maximum number of VNT 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 VNT candidate in all clusters in the middle layer. As a result, we take 26 VNT candidates for the middle layer. The VNT 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 VNT candidates in the top layer, 26 VNT candidates in the middle layer, and one VNT candidate in the bottom layer. Finally, we obtain 676 VNT candidates.

Table II shows the number of ports used in each layer in the VNT candidates. In the top layer, each of the VNT 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 $50 \times ((14 \times 2)^2 + (48 \times 2)) = 44{,}000$ ports in the top layer. Each of the VNT 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 $50 \times (14 \times 2) \times 10 = 14{,}000$ ports in the middle layer. In the bottom layer, there are 100 clusters, and each cluster consists of 10 nodes. The VNT candidates in a

TABLE II
NUMBER OF PORTS USED IN EACH LAYER: 1000-NODE NETWORK

Layer	Number of Ports
0th layer	44,000
1st layer	14,000
2nd layer	6000

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 to a lognormal distribution. We assume that the transmission capacity of all lightpaths is equal throughout the evaluation. Therefore, since the total traffic demand is excessive for the 1000-node network, we use half the traffic demand used in the 100-node network. The comparison method is the same as that used in the evaluation of the 100-node network.

Figure 10 shows the distribution of maximum link utilization for each traffic pattern. In Fig. 10, we can see that the VNT candidates obtained by our method make the maximum link utilization lower than those by the other method for all traffic patterns. We find that our method can design better VNT candidates for 1000-node networks, and we believe that our method can design better VNT candidates for even larger-scale networks. Comparing Figs. 9 and 10, we can see that the variance of the lowest value of maximum link utilization among the VNT candidates for each traffic pattern for the 1000-node network is much less than that of the 100-node network. This is because the plotted value in Fig. 10 is likely to be the suboptimal value for each traffic pattern with the same distribution. The number of VNT candidates for the 1000-node network is much larger than that of the 100-node network. Specifically, the number of VNT candidates is 676 for the 1000-node network, and it is 49 for the 100-node network. When the number of VNT candidates is large, there is a high possibility that any of them will show the suboptimal value of maximum link utilization. Moreover, we believe that the suboptimal values for traffic patterns with the same distribution are close to each other. Therefore, the variance of the plotted values for the 1000-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 1000-node network, although the number of layers is the same, the cluster size (i.e., the number of nodes in a cluster) and the number of clusters in each layer in the latter network are 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 I. In the 1000-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 II.

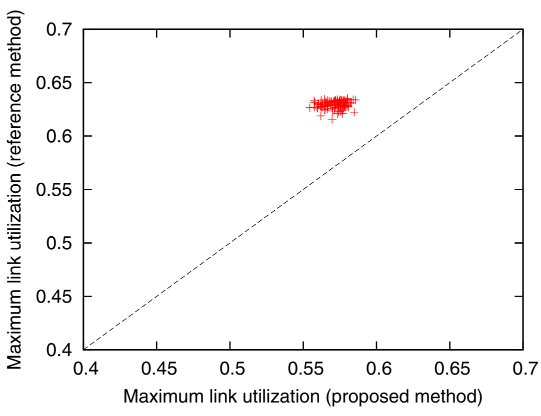

Fig. 10. Distribution of maximum link utilization: 1000 nodes, 64 ports.

B. Adaptability of VNT Control Based on Attractor Selection

In this section, we evaluate the adaptability of our VNT control method based on the attractor selection described in Section II, using VNT candidates obtained by our method as attractors. We use the term "adaptability" to represent a shorter number of steps of VNT control until convergence, i.e., a shorter time until VNT control finds a solution. When VNT control requires a shorter time to find a solution, the VNT control is more "adaptive" in response to traffic changes. We set the target maximum link utilization θ in Eq. (3) to 0.5, and consider our VNT control to have succeeded when the maximum link utilization is reduced to less than 0.5 within 10 successive steps of the VNT control. We assume that traffic fluctuations occur at time zero and evaluate the number of steps in VNT control until convergence, which is required for success of the VNT control. At each step, our VNT control method collects load information on all lightpaths, calculates the activity α, and reconfigures the VNT. We set μ of the sigmoidal function $\varsigma(z)$ in Eq. (2) to 20, and set γ to 1 and δ to 50 in Eq. (3).

Figure 11 shows the distribution of the number of steps until convergence in the 100-node network. The traffic demand and VNT candidates used as attractors are similar to those described in Subsection V.A. 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 VNT control using the VNT candidates given by our method as attractors requires fewer steps until convergence. Since the VNT candidates from our method are able to reduce the maximum link utilization, as shown in Subsection V.A, our VNT control method is guided by better attractors and finds a solution, that is, a VNT that can accommodate traffic, within a shorter time.

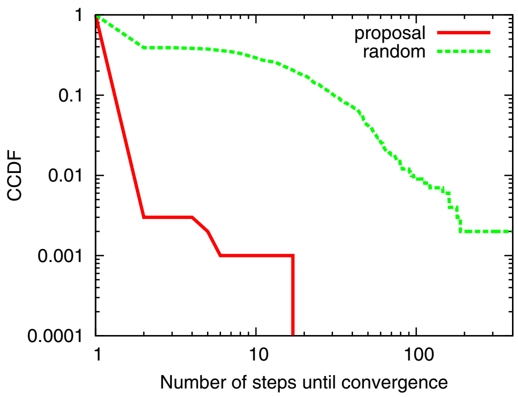

Fig. 11. Distribution of the number of steps until convergence: 100 nodes, 32 ports.

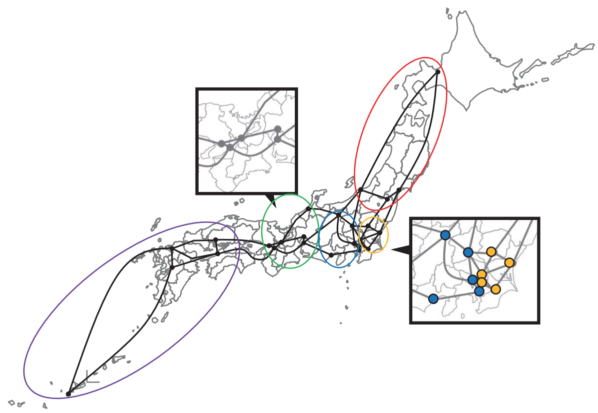

Fig. 12. Clusters in JPN25 model.

When we evaluate the VNT control method based on attractor selection in the 1000-node network by using the VNT candidates from our method as attractors, we expect to obtain similar results to those above on the basis of the results we have already obtained. The VNT candidates from our method in the 100-node network can better reduce the maximum link utilization than the candidates from the other method, as shown in Fig. 9, and the VNT control using the VNT candidates given by our method as attractors requires fewer steps until convergence, as shown in Fig. 11. Since the VNT candidates from our method in the 1000-node network can also better reduce the maximum link utilization, as shown in Fig. 10, it is expected that the VNT control using the VNT candidates given by our method as attractors in the 1000-node network requires fewer steps until convergence.

C. Effect of Physical Network Topology on Adaptability of VNT Control

In this section, we design VNT candidates by dividing a large-scale network into clusters based on the physical network topology. We then evaluate the adaptability of our VNT control method using the VNT candidates as attractors. We use the JPN25 model and USNET as physical network topologies. We use the method in Section IV to design the VNT candidates for the two networks. The Louvain method [16] 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.

1) 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. 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 a top layer in which the nodes are the clusters and a bottom layer that 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 VNT candidates by following Step 3 of our method. The VNT 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 VNT candidates in the top layer and one VNT candidate in the bottom layer. Finally, we obtain seven VNT candidates.

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

Figure 13 shows the distribution of the number of steps until convergence. The traffic demand and method of comparison are the same as in Subsection V.B. 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 VNT control using VNT candidates from our method as attractors requires fewer steps until convergence. Since our VNT control method using the VNT candidates from our method as attractors finds a solution within a shorter time, this shows that our method can design better VNT candidates than the other method, even when we decide clusters on the basis of the physical network topology.

2) 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. 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

TABLE III
NUMBER OF PORTS USED IN EACH LAYER: JPN25 MODEL

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

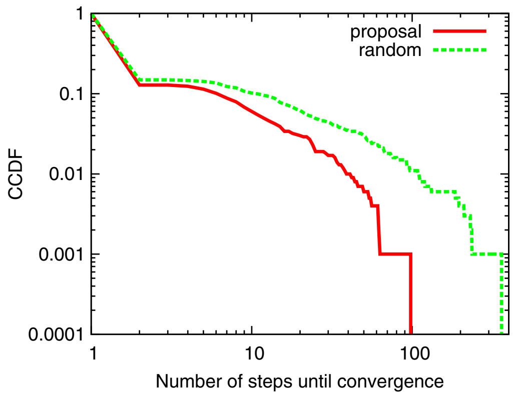

Fig. 13. Distribution of the number of steps until convergence: JPN25 model, 10 ports.

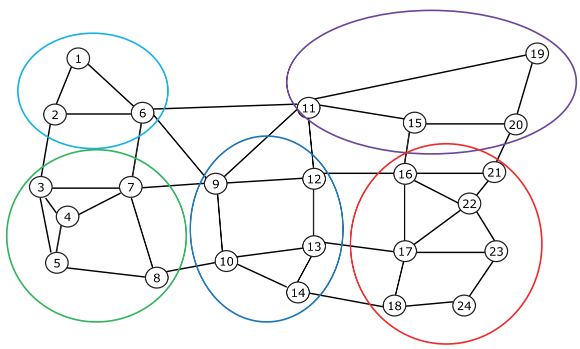

Fig. 14. Clusters in USNET.

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 router ports, and we obtain seven VNT candidates by following Step 3 of our method. The VNT 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 VNT candidates in the top layer and one VNT candidate in the bottom layer. Finally, we obtain seven VNT candidates.

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

Figure 15 shows the distribution of the number of steps until convergence. The traffic demand and method of

TABLE IV
Number of Ports Used in Each Layer: USNET

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

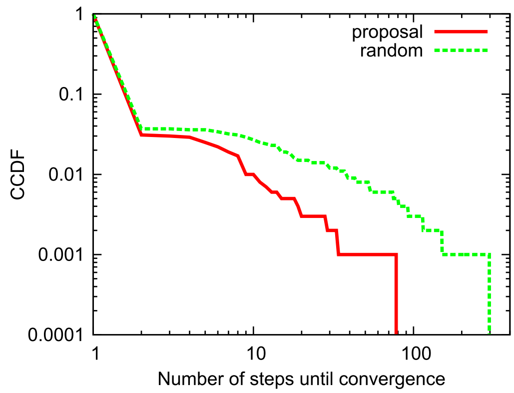

Fig. 15. Distribution of the number of steps until convergence: USNET, 10 ports.

comparison are the same as in Subsection V.B. 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 VNT control using VNT candidates from our method as attractors requires fewer steps until convergence. That is, our VNT control method using VNT candidates from our method as attractors finds a solution within a shorter time. Our method can design better VNT candidates than the other method for another physical network.

As mentioned in Subsection V.A, 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 the JPN25 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 VNT. 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.

VI. Conclusion

In this paper, we proposed a method for designing attractors for our VNT control method based on attractor selection. Our basic approach is to prepare a limited number of attractors with a diversity of characteristics by classifying VNT candidates in 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 VNT candidates obtained by our method can accommodate various traffic demand patterns, so that our VNT control method can find a solution, that is, a VNT 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 VNT control method based on attractor selection will become even more adaptive to traffic changes.

Acknowledgment

Part of this work was supported by the National Institute of Information and Communications Technology (NICT).

References

[1] J. Wu, “A survey of WDM network reconfiguration: Strategies and triggering methods,” *Comput. Netw.*, vol. 55, pp. 2622–2645, May 2011.

[2] D.-R. Din and C.-W. Chou, “Virtual-topology adaptation for mixed-line-rate optical WDM networks under dynamic traffic,” in *Proc. Int. Conf. on Computer Communications and Networks*, Aug. 2014.

[3] X. Zhang, H. Wang, and Z. Zhang, “Survivable green IP over WDM networks against double-link failures,” *Comput. Netw.*, vol. 59, pp. 62–76, Feb. 2014.

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

[5] S. Gieselman, N. Singhal, and B. Mukherjee, “Minimum-cost virtual-topology adaptation for optical WDM mesh networks,” in *Proc. Int. Conf. on Communications*, June 2005, pp. 1787–1791.

[6] G. Agrawal and D. Medhi, “Lightpath topology configuration for wavelength-routed IP/MPLS networks for time-dependent traffic,” in *Proc. IEEE GLOBECOM*, Nov. 2006.

[7] D. Banerjee and B. Mukherjee, “Wavelength-routed optical networks: Linear formulation, resource budgeting tradeoffs, and a reconfiguration study,” *IEEE/ACM Trans. Netw.*, vol. 8, pp. 598–607, Oct. 2000.

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

[9] A. Schaeffer-Filho, P. Smith, A. Mauthe, and D. Hutchison, “Network resilience with reusable management patterns,” *IEEE Commun. Mag.*, vol. 52, no. 7, pp. 105–115, July 2014.

[10] 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 Trans. Netw.*, vol. 18, pp. 177–189, Feb. 2010.

[11] Y. Koizumi, T. Miyamura, S. Arakawa, E. Oki, K. Shiomoto, and M. Murata, “Adaptive virtual network topology control based on attractor selection,” *J. Lightwave Technol.*, vol. 28, pp. 1720–1731, June 2010.

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

[13] Y. Baram, “Orthogonal patterns in binary neural networks,” NASA Technical Memorandum No. 100060, Mar. 1988.

[14] C. Furusawa and K. Kaneko, “A generic mechanism for adaptive growth rate regulation,” *PLoS Comput. Biol.*, vol. 4, pp. 35–42, Jan. 2008.

[15] R. Rojas, *Neural Networks: A Systematic Introduction*. Berlin: Springer, 1996.

[16] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, “Fast unfolding of communities in large networks,” *J. Stat. Mech.*, vol. 2008, pp. 10008–10019, Oct. 2008.

Toshihiko Ohba received an M.E. degree in information science and technology in 2015 from Osaka University, where he is currently a postgraduate studying for a Ph.D. degree. His research interests include optical networks.

Shin’ichi Arakawa received M.E. and D.E. degrees in informatics and mathematical science from Osaka University in 2000 and 2003, respectively. From August 2000 to March 2006, he was an Assistant Professor with the Graduate School of Economics, Osaka University, Japan. In April 2006, he moved to the Graduate School of Information Science and Technology, Osaka University, Japan. He has been an Associate Professor since October 2011. His research interests include optical networks and complex networks. He is a member of IEEE and IEICE.

Yuki Koizumi received M.E. and D.E. degrees in information science from Osaka University, Japan, in 2006 and 2009, respectively. He is currently an Assistant Professor at the Graduate School of Information Science and Technology, Osaka University, Japan. His research interests include traffic engineering in photonic networks and biologically inspired networking. He is a member of IEEE and IEICE.

Masayuki Murata received M.E. and D.E. degrees in information science and technology from Osaka University in 1984 and 1988, respectively. In April 1984, he joined the Tokyo Research Laboratory, IBM Japan, as a Researcher. From September 1987 to January 1989, he was an Assistant Professor with the Computation Center, Osaka University. In February 1989, he moved to the Department of Information and Computer Sciences, Faculty of Engineering Science, Osaka University. From 1992

to 1999, he was an Associate Professor with the Graduate School of Engineering Science, Osaka University, and since April 1999, he has been a Professor. He moved to the Graduate School of Information Science and Technology, Osaka University, in April 2004. He has published more than 300 papers in international and domestic journals and conferences. His research interests include computer communication networks, performance modeling, and evaluation. He is a fellow of IEICE and a member of IEEE, the Association for Computing Machinery (ACM), The Internet Society, and IPSJ.