A Managed Self-Organization of Virtual Network Topology Controls in WDM-based Optical Networks

Shin'ichi Arakawa*, Yuki Minami*, Yuki Koizumi*, Takashi Miyamura†, Kohei Shiomoto† and Masayuki Murata*
**Graduate School of Information Science and Technology*
Osaka University
Osaka 565-0871, Japan
{*arakawa, y-minami, ykoizumi, murata*}*@ist.osaka-u.ac.jp*
†*NTT Network Service Systems Laboratories*
NTT Corporation
Tokyo 180-8585, Japan
{*miyamura.takashi, shiomoto.kohei*}*@lab.ntt.co.jp*

***Abstract*—In this paper, we propose a framework to construct multiple VNTs in WDM-based optical network and then propose a control scheme, which we call a managed self-organization, to achieve adaptive and efficient VNT controls. Our basic idea is that, we introduce a system manager for managing and controlling the overall performance in multiple VNTs environments. The system manager collects the activities that represent the conditions of VNTs, calculate the overall condition of the network, and loosely controls each VNT through the feedback of the information about the overall condition. Simulation results show that the VNT control with system manager can adapt to the large traffic fluctuation within 47 minutes, while the VNT control without system manager takes more than 200 minutes.**

***Keywords*-Wavelength Division Multiplexing); Virtual Network Topology Control; Self-organized control; Managed self-organization; IP (Internet Protocol)**

I. Introduction

Wavelength-routed optical network is one of promising architectures to accomodate IP traffic in a cost-effective way. In the IP over WDM network, virtual network topology (VNT), which consists from lightpaths, is constructed on top of the wavelength-routed optical network.

Recently, overlay services on existing IP-based networks have expected to be a viable solution to deploy new services without violating underlying protocol standards. However, rapid increase of new overlay services may lead to the increase of traffic growth without control, which will result in a severe degradation of quality of service perceived by existing services. In Ref. [1], authors point out that the various services, such as peer-to-peer networks, voice over IP, and video on demand causes large fluctuations on traffic demand in networks. Koizumi et. al [2] points out that, when there are overlay networks on top of the network controlled by the VNT control mechanism, traffic demand fluctuates greatly and changes in traffic demand are unpredictable. In more recent years, various new services such as SaaS and cloud services have emerged. However, because we do not know the traffic requirements of the new emerging services and we do not know its effect on the quality of existing services, we urgently need a framework for control and manage the current and future services in a flexible manner.

One of approach to overcome the performance degradation induced by multiple services is to prepare multiple VNTs over WDM-based optical networks, and then assign one or more VNTs to each service. We can take several approaches for managing multiple VNTs. For example, we can deploy centralized VNT control methods that control VNTs by collecting detailed information about the all of VNTs and by calculating VNT based on the information. However, it is easily imagined that the centralized VNT control methods have long control duration, and cannot adapt to rapid traffic changes induced by the overlay services.

In this paper, we investigate a framework for a self-organized VNT control in WDM-based optical networks. Basis ideas of our framework is that each service deploys a VNT controller, and then configure the VNT by interactions between own condition and local information such that required quality of each service is satisfied. To realize this, we apply VNT control method based on attractor selection [3] for each VNT control. In attractor selection, the system works by deterministic behavior and stochastic behavior. These two behaviors are controlled by simple feedback of the current condition in the system. The system can adapt to unknown changes by searching preferable condition randomly with noise. Self-organized VNT control methods cannot configure highly optimized VNT. However, self-organized VNT control methods have high adaptability against changes of the environments [4], [5]. On one hand, self-organized VNT control methods can adapt to the change of environments. On the other hand, it is insufficient for multiple VNTs environments to use only the self-organized VNT control methods because self-organized VNT control method is not careful about other VNTs.

We therefore develop a managed self-organization method of VNT controls to achieve adaptive and efficient VNT controls for multiple VNTs environments. In the method, we introduce a system manager to control general situation of the all of VNTs. The system manager does not observe the details of the network and does not control all VNTs, but observes a condition of each VNT. The VNT controller prepared for each VNT collects conditions of its own VNT, and grasp whether each VNT control is acceptable or not. The conditions are collected by the system manager. Using the collected information, the system manager send a feedback which in turn used by the VNT controller of each VNT. Note that it is important that what kind information system manager collects and what kind of information system manager sends as the feedback. We describe details of managed self-organization in Section IV.

This paper is organized as follows. Section III summarizes the related work. Section III shows our network model. Section IV-A explains details of VNT control based on attractor selection. Section IV describes details of managed self-organization of VNT controls. Section V shows the computer simulation and the results of our method. Finally, we conclude this thesis in Section VI.

II. Related Work

Many approaches to accommodate traffic demand by configuring VNTs have been investigated [3], [6]–[17]. One of approaches is that VNTs are statically constructed to efficiently accommodate one or multiple traffic demand matrices. For example, in Ref. [13], authors propose VNT control to minimize link utilization of VNT. The method redesigns the virtual topology according to an obtained traffic demand matrix through traffic measurements for a long-period of time.

Recently, the dynamic VNT control that dynamically reconfigures VNTs based on their detection of degraded performance or periodic measurements of the network status without a priori knowledge of future traffic demand has been proposed. In Ref. [15], authors try to keep the link utilizations between an upper limit and a lower limit against the traffic that grows during peak hours and falls during off-peak hours. Ohsita et al. developed a VNT control method with an estimation of traffic demand matrices, and show that the method works with an hour-order change of traffic demand [17].

More recently, Koizumi et al. demonstrated a self-organized VNT control based on the attractor selection model [3]. The self-organized VNT controls have high adaptability against changes of environment. However, they work randomly by noise when the system conditions are poor, and they are not careful about other VNTs. Therefore when there are many VNTs at the same time on a physical network, VNTs might be satisfied by the current VNT configurations and continue to use the resources allocated to the VNT, which leads to a monopoly of physical resources.

There are many researches about self-organizing systems and the management of them. In Ref. [18], [19], authors describe “autonomic computing” or “organic computing”. These are technical systems that adapt dynamically to the current conditions of its environment. They have features like self-organization, self-optimization, self-healing, self-protection, and so on. In Ref. [20], authors show generic observer/controller architecture for organic computing. The system architecture have degree of freedom allowing for self-organized behavior, but at the same time the system architecture can control the emerging global behavior of these self-organizations. In Ref. [20], authors propose observer/controller architecture to realize this. The observer observes the self-organized systems that react to inputs and output the results. The main tasks of observer are to grasp the current status of the self-organized system and to predict the future status of the self-organized system. The controller controls the self-organized system by using information reported by observer. The main tasks of controller is to influence the self-organized system such that a desired emergent behavior appears, to disrupt an undesired emergent behavior, and to construct the system in a way such that no undesired emergent behavior can appear.

III. Network Model

In this section, we describe the network model that we will use for the multiple VNTs configuration.

Each node in the physical network has IP routers and OXCs (Figure 1) and nodes are connected with optical fibers (Figure 2). The OXCs consist of three main blocks: input section, non-blocking optical switches, and output section. In the input section, optical signals are demultiplexed into W fixed wavelengths. Then, each wavelength is transferred to an appropriate output port by the non-blocking optical switch. Finally, at the output section, each wavelength is multiplexed again and sent to the next node. By configuring the switches along a path, a lightpath is configured and a particular wavelength is carried from a transmitter at a node to a receiver at the other node without any electronic processing (Figure 3).

A VNT is then constituted by setting up lightpaths on top of the physical network (Figure 4). The actual traffic of the upper layer protocol, such as the IP, is carried on the constructed VNT. When a lightpath terminates at this node, the IP packets on the lightpath are converted to electrical signals and forwarded to the electronic router. When a lightpath begins at this node, IP packets from the electronic router are transmitted over the lightpath after being converted to optical signals. Each IP router refers transport layer header of the packet and judges which VNT should be used. By the information, IP router decides next output interface. If lightpath is configured between all node-pairs, we do not need IP’s packet processing in the VNTs. However, since

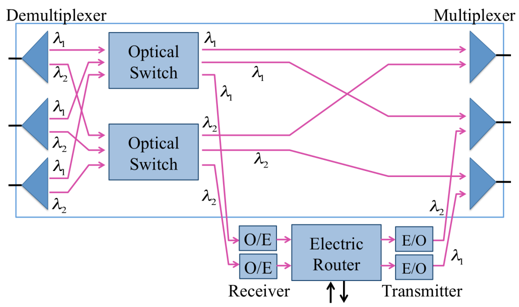

Figure 1. Node architecture

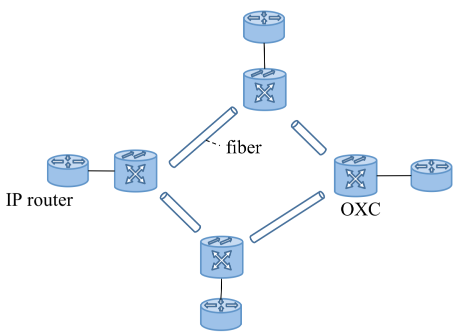

Figure 2. IP-over-WDM network

the number of transmitters/receivers is limited, we should properly configure and reconfigure multiple VNTs.

Figure 5 shows an example of use of physical resources by multiple VNTs. Each IP router is connected with each OXC as shown in Figure 4. Service0 uses IP router A', B', and C'. Service1 uses IP router A', B', C', and D'. VNT0 for service0 uses transmitters and receivers of OXC A, B, and C for setting up the lightpaths. VNT1 for service1 uses transmitters and receivers of OXC A, B, C, and D for setting up the lightpaths. Therefore VNT0 and VNT1 share the physical resources at OXC A, B, and C in the WDM network.

VNT controller configures VNT for the service. VNT controller observes own VNT performance and tries to reconfigure VNT properly according to the VNT control method. A VNT controller sends a request to physical resource manager to assign physical resources for the VNT configuring. Physical resource manager manages physical resources of the network. Physical resource manager receives the requests from each VNT controller and assigns physical resources to each VNT following the requests (Figure 6).

In our network model, when enough physical resources for the request remain, physical resource manager assigns physical resources and VNT controller configures the VNT. When enough physical resources do not remain, VNT controller keeps current VNT and sends a request to physical resource manager again at the next control period.

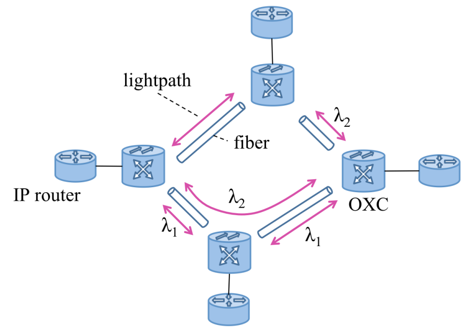

Figure 3. Wavelength routing

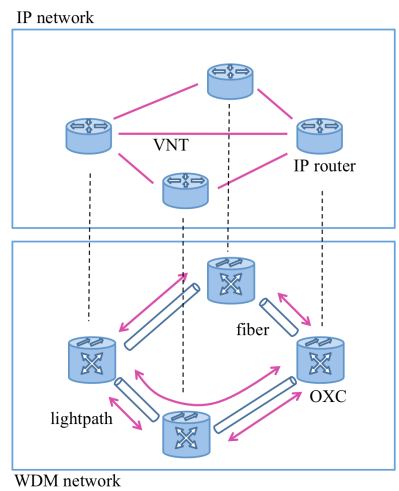

Figure 4. VNT configuration in IP-over-WDM network

Therefore, if VNT0 use all transmitters and receivers of OXC A, then VNT1 cannot set up a new lightpath from or to IP router A'. In such case, we can avoid the competition by reconfiguring VNT0 such that VNT0 does not use all transmitters and receivers of OXC A. In another method, we can change allocation of physical resources by setting the physical resources as exclusive resources or shared resources. Exclusive resources are used only for one service. On the other hand, shared resources can be used for all services. In this way, services share physical resources of WDM network and configure VNTs.

We next explain the sequence of VNT control. Figure 7 shows the sequence when only one VNT exists. Figure 8 shows the sequence when multiple VNTs and system manager exist.

First, we explain the sequence of one VNT reconfiguration.

Step. 1 VNT controller observes performance of the VNT over which the service transfers IP traffic.

Step. 2 VNT controller calculates new VNT to improve the performance according to the VNT control method by using the observed information.

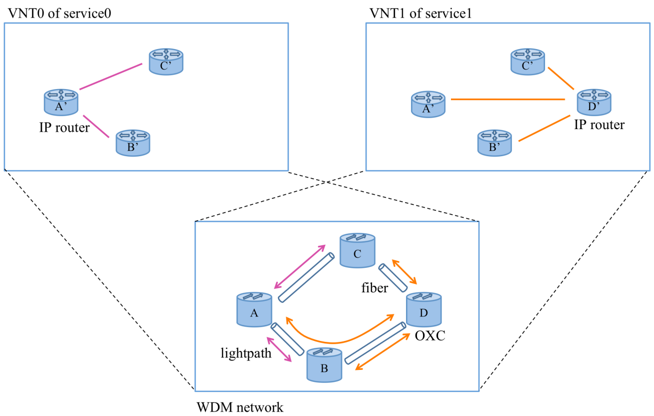

Figure 5. Multiple VNTs on top of a physical network

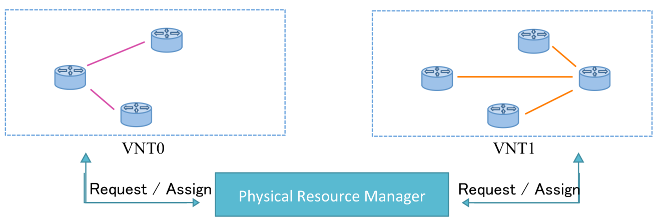

Figure 6. Physical resource manager

Step. 3 VNT controller sends a request to physical resource manager to assign physical resources for the new VNT. Physical resource manager assign physical resources if enough resources remain. If not, physical resource manager reports it to VNT controller and does not assign physical resources.

Step. 4 VNT controller reconfigures VNT if physical resources were assigned. If not, VNT controller keeps current VNT.

Step. 5 Service transfers the IP traffic over the new VNT. Consequently the performance of the VNT changes again, so we repeat these steps again.

Next, we explain the sequence of multiple VNTs reconfiguration with system manager.

Step. 1 VNT controller observes performance of the VNT over which the service transfers IP traffic.

Step. 2 VNT controller calculates activity of the VNT. System manager collects activity of each VNT.

Step. 3 System manager calculates feedback from collected activities. The feedback indicates overall performance of all VNTs. System manager sends feedback to each VNT controller.

Step. 4 VNT controller calculates new VNT with the local information and feedback about the overall VNTs.

Step. 5 VNT controller sends a request to physical resource manager to assign physical resources for the new VNT. Physical resource manager assign physical resources if enough resources remain. If not, physical resource manager reports it to VNT controller and does not assign physical resources.

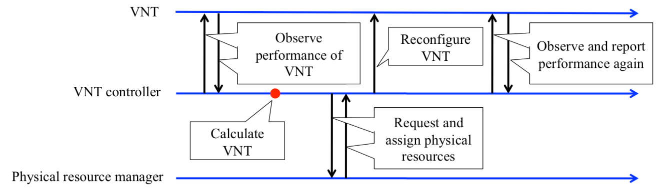

Figure 7. Information exchange for single VNT control

Step. 6 VNT controller reconfigures VNT if physical resources were assigned. If not, VNT controller keeps current VNT.

Step. 7 Transfer the IP traffic over the new VNT. Consequently the performance of the VNT changes again, so we repeat these steps again.

In this way, we configure one or multiple VNTs.

In multiple VNT controls, we should reduce control overheads more strictly than one VNT control. The larger the number of VNTs becomes, the larger the information to control the all of VNTs become. Moreover, to adapt to various changes in the network quickly before influences of the change become big, control duration of each VNT control should be short. Therefore, the information to manage the all of VNTs should be simple and small.

In our method, we use self-organized VNT control method to control each VNT and use managed self-organization of VNT controls to manage the all of VNTs. Our method can control multiple VNTs by the adaptability of self-organization. Moreover, our method can manage the all of VNTs by using the simple feedback from system manager.

We briefly explain the self-organized VNT control method in Section IV-A and show the details of managed self-organization of VNT controls in Section IV.

IV. Managed Self-Organization of VNT Controls

In this section, we show the framework for the managed self-organization of VNT controls. We first explain self-organized VNT control method that can be used to control each VNT. From Section IV-B, we show the framework for the managed self-organization of VNT controls. We first explain the functions of VNT controller, physical resource manager, and system manager. Next, we describe management of the interaction between the multiple VNTs. In Section IV-C, we explain details of managed self-organization of VNT controls by showing the mathematical model.

A. Self-organized VNT control

Koizumi et al. developed self-organized VNT control method based on attractor selection [3] for single VNT control. The attractor selection model originally represents metabolic reactions controlled by gene regulatory networks

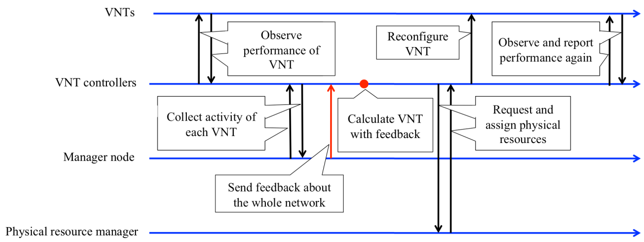

Figure 8. Information exchange for multiple VNTs control

in a cell. Each gene in the gene regulatory network has an expression level of proteins and deterministic and stochastic behaviors in each gene control the expression level. An attractor selection model is consists of regulatory behaviors having attractor which is determined by activation and inhibition between each genes, growth rate as feedback of the current condition of the network, and noise, which is stochastic behavior.

Koizumi et al. consider the dynamical system that is driven by the attractor selection. The authors place genes on every source-destination pair (denote p_{ij} for nodes i and j) in the WDM network, and the expression level of the genes $x_{p_{ij}}$ determines the number of lightpaths on between nodes i and j. The dynamics of $x_{p_{ij}}$ is defined by the following differential equation,

$$\frac{dx_{p_{ij}}}{dt} = v_g \cdot f\left(\sum_{p_{sd}} W(p_{ij}, p_{sd}) \cdot x_{p_{sd}} - \theta_{p_{ij}}\right) - v_g \cdot x_{p_{ij}} + \eta \quad (1)$$

where η represents Gaussian white noise, f is the sigmoidal regulation function, and v_g is the growth rate. v_g indicates the condition of the IP network.

Attractors are a part of the equilibrium points in the solution space in which the current condition is preferable. In the current case, attractor represents a VNT. The basic mechanism of an attractor selection consists of two behaviors: deterministic and stochastic behaviors. When the current condition is suitable for the current environment, i.e., the system state is close to one of the attractors, deterministic behavior drives the system to the attractor.

When the current condition is poor, stochastic behavior dominates over deterministic behavior. While stochastic behavior is dominant in controlling the system, the system state fluctuates randomly due to noise and the system searches for a new attractor. When the current condition has recovered and the system state comes close to an attractor, deterministic behavior again controls the system. These two behaviors are controlled by simple feedback of the current condition in the system. In this way, attractor selection adapts to environmental changes by selecting attractors using stochastic behavior, deterministic behavior, and simple feedback.

B. Outline of Managed Self-Organization

1) Functions of each control unit: Figure 9 shows the model of one VNT control based on self-organization, and Figure 10 shows the model of multiple VNT controls with system manager.

- VNT controller
  VNT controller configures VNT for the service. The VNT control method is based on attractor selection (Section IV-A). VNT controller observes own activity, and tries to reconfigure VNT properly to improve the activity. When system manager sends feedback to VNT controller, VNT controller reconfigures VNT by regarding the feedback as its own activity. Therefore, if the feedback indicates that the condition of the network is poor, VNT controller reconfigures its own VNT randomly by noise. After calculation of new VNT, the VNT controller sends a request to physical resource manager to assign physical resources for the new VNT configuring. VNT controller reconfigures VNT when physical resources are assigned. If physical resources are not assigned because of physical resource competition between other VNTs, VNT controller keeps current VNT.
- Physical resource manager
  Physical resource manager manages physical resources of the network. Physical resource manager receives requests from each VNT controller, and assigns physical resources to each VNT following the situation. When enough physical resources for a request remain, physical resource manager assigns physical resources for the VNT. If enough physical resources for a request do not remain because of physical resource competition between other VNTs, physical resource manager sends message about it to the VNT controller.
- System manager
  System manager consists of network observer and network controller. System manager collects activities of all VNTs. This is different from centralized network control method that collects much information from network. Activity is just a simple information; we do not need information of traffic demand matrix and/or link utilizations of all lightpaths that may be necessary for centralized VNT control. System manager calculates the activity of the whole network. We will refer to the activity as *network activity*. The network activity is based on the collected activities. System manager sends feedback to each VNT controller, and the network activity is used as the feedback. Again, the feedback is simple information. Each VNT controller reconfigures own VNT using the feedback. By this way, system manager controlls the degree of self-organization and manages the all of VNTs by sending feedback.

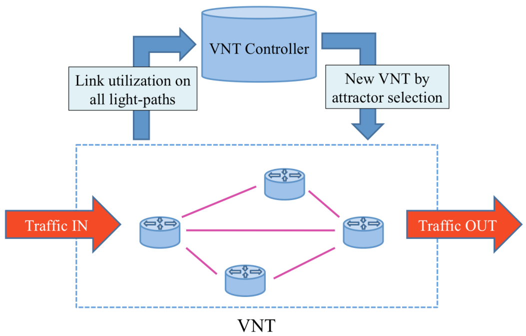

Figure 9. Self-organized VNT control method

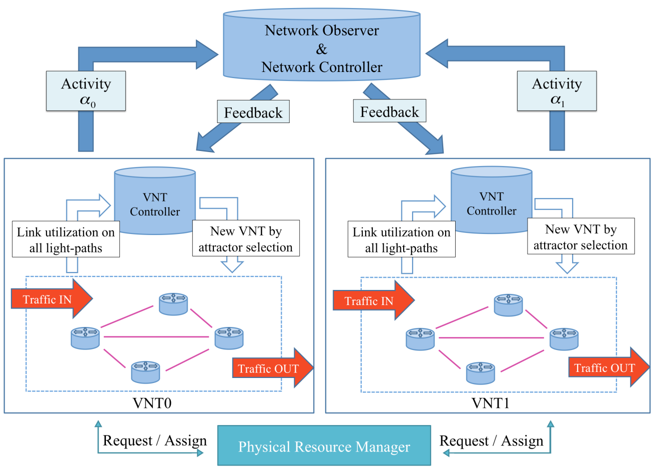

Figure 10. Concept of multiple VNTs control based on a managed self-organization

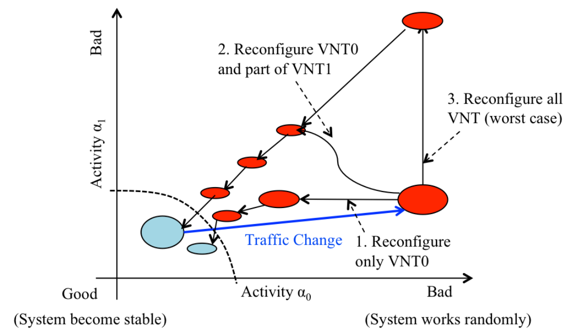

Figure 11. Interaction between the multiple VNTs

2) Management of interaction between the multiple VNTs: Figure 11 shows image of management of interaction between the multiple VNTs. In Figure 11, we assume that there are two VNT, called VNT0 and VNT1, in a physical network. Horizontal axis represents the value of α_0, which is the activity of the VNT0. Vertical axis represents the value of α_1, which is the activity of the VNT1. α_0 and α become high value when the condition of the system is good, and become low value when the condition of the system is bad. In the right of the figure, α_0 is low value. In the top of the figure, α_1 is low value. At first, both α_0 and α_1 are high value. When the change of traffic demand of VNT0 occurs, α_0 become low value (blue arrow in the figure). Here, when the change of traffic demand is low, it would be solved by reconfiguring of only VNT0. Therefore we reconfigure only VNT0 until a certain time pass (arrow #1). After a certain time passed, if α_0 do not become high value, we think that the change of traffic demand is high or physical resource competition between other VNTs occurs, and it cannot be solved by reconfiguring of only VNT0. Therefore we reconfigure VNT0 and part of VNT1 by sending feedback from system manager to VNT0 and VNT1 (arrow #2). After further time passed, we full reconfigure VNT0 and VNT1 (arrow #3). When the change of traffic demand is small, it would be more quick and efficient to reconfigure only the VNTs in poor condition rather than to reconfigure all of the VNTs in the network. Then we almost fix the VNTs in preferable condition and reconfigure the VNTs in the poor condition. When the change of traffic demand is large and VNTs in poor condition cannot recover the condition after a certain period of time, we consider the VNTs in poor conditions cannot adapt to the change by reconfiguring only its own VNTs. We then reconfigure all of VNTs in the network by the feedback from system manager. Then the physical resources in the network are reallocated to each VNT, and we can avoid physical resource competitions. In this way, our method can adapt to changes, and avoid physical resource competition.

C. Controlling activities for Managed Self-organization

In this section, we explain details of managed self-organization of VNT controls. When there are N VNTs in the network, each VNT has an activity, $\alpha_0, \alpha_1, ..., \alpha_N$. Each activity indicates the condition of the corresponding VNT. These activities are calculated by each VNT controller as explained in Section IV-A. We define α_{master} as the network activity. α_{master} is calculated from $\alpha_0, \alpha_1, ..., \alpha_N$.

Here, when the all activities in the network are high value, system manager does not update α_{master}. System manager send α_{own} as the feedback to each VNT. α_{own} is the same value as the value of α of each VNT.

Because then all VNTs are good condition, we do not need to change the VNTs. When some activities are low value, we update α_{master} and send α_{master} as the feedback to each VNT.

We update α_{master} by the following equation.

$$\alpha_{master} = \begin{cases} \alpha_{\min} * D(t) & \text{if } \alpha_{\min} < Th_\alpha \text{ and } t > Th_{time} \\ \alpha_{own} & \text{if } \alpha_{\min} \geq Th_\alpha \text{ or } t \leq Th_{time} \end{cases} \quad (2)$$

$\alpha_{\min}$ is the minimum value between $\alpha_0, \alpha_1, ..., \alpha_N$. Th_α is a threshold value about α. Th_{time} is a threshold value about time. t is the elapsed time from when $\alpha_{\min}$ become worse than Th_α. $D(t)$ is the rate decided by t, and it ranges from 0 to 1.

V. Computer Simulation

We conduct computer simulation to evaluate our management model. Section V-A shows simulation conditions, and Section V-B shows simulation results.

A. Simulation Conditions

We use the European Optical Network (EON) topology as the physical topology (Figure 12). The EON topology has 19 nodes and 39 bidirectional fibers. Each node has transmitters and receivers that are used as exclusive physical resources of each service and that are used as shared physical resources between all services. In the simulation, we deploy two services in the network, and the two services configure VNT0 and VNT1. The number of transmitters/receivers of each node is $D_{node}*2+8$. Here, D_{node} is the degree number of the node. Both services can use exclusive transmitters and receivers for base VNT that corresponds with physical topology. The lightpaths used for configuring the base VNT is set at all time irrespective of VNT control. Moreover, each service shares eight transmitters and receivers with other services per a node.

Each VNT controller configurse the VNT by self-organized VNT control (Section IV-A). The parameter settings used in Equation 1 are shown in Table I, and we set N_{path} to 20. Each VNT controller collects information about the link utilization by SNMP and reconfigures VNT every one minute. System manager collects activities from all VNTs and sends a feedback based on the activities to each VNT every one minute. The value of the feedback is used as the activity in each VNT, and the VNT controller reconfigure VNT based on the updated activity.

We prepare the traffic demand matrices where traffic demand from node i to j, d_{ij}, follows a lognormal distribution. We set the variance of logarithm of d_{ij} to be σ^2 and with the mean to be 1. Each traffic demand matrix is normalized such that the total amount of traffic, $\sum_{p_{ij}} d_{p_{ij}}$, is the same and is set to 6.5 in a unit of bandwidth of lightpaths. We change traffic demand of only VNT1 at 10 minutes by setting the different value of random seed for d_{ij}. We use the maximum link utilization on VNT as a metric that indicates the current condition of VNT. In the simulation, we set the target maximum link utilization to 0.5. Then the activity of the VNT is 50 in the parameter settings. Therefore, we set Th_α to 50. When the maximum link utilization of one VNT becomes less than 0.5, and the maximum link utilization of another VNT becomes higher than 0.5, system manager begins to count the time t, that is the time from when $\alpha_{\min}$ becomes worse than Th_α. When t becomes bigger than Th_{time}, that is threshold value about time, we consider that the situation cannot be solved by reconfiguration of VNT in a poor condition because of large change of the environment of the VNT and/or physical resource competition. Therefore, the system manager sends low value feedback to each VNT, and the VNTs in the network are reconfigured by noise. To adapt the all of VNTs to change of the environment quickly, we set Th_{time} to 4 minute, and set $D(t)$ as Equation 3.

Table I
PARAMETER SETTINGS OF EACH SELF-ORGANIZED VNT CONTROL METHOD

Name	Values
γ	100
δ	13
δ'	3
ζ	0.5
μ	1

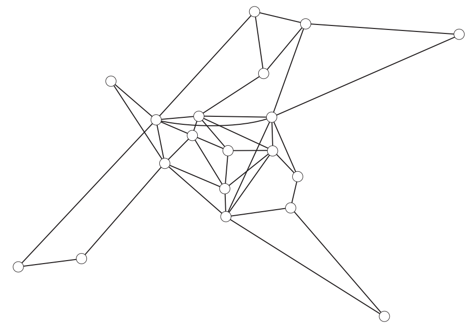

Figure 12. EON topology

$$D(t) = \begin{cases} 1 & \text{if } 4\text{minutes} < t \leq 6\text{minutes} \\ 0.5 & \text{if } 6\text{minutes} < t \leq 8\text{minutes} \\ 0.25 & \text{if } 8\text{minutes} < t \leq 10\text{minutes} \\ 0.125 & \text{if } 10\text{minutes} < t \leq 12\text{minutes} \\ 0.0625 & \text{if } 12\text{minutes} < t \leq 14\text{minutes} \\ 0 & \text{if } 15\text{minutes} < t \end{cases} \quad (3)$$

In this way, when poor condition of VNT cannot be solved by reconfiguration of the VNT because large change of environment or physical resource competition occurs, we reconfigure the all of VNTs by sending feedback.

B. Simulation Results

Figure 13, 14, 15, 16, 17, and 18 show maximum link utilization dependent on time, activity dependent on time, and transition of activities. In Figure 13, 14, and 15, system manager does not work. In Figure 16, 17, and 18, system manager works and sends feedback to VNT0 and VNT1.

In Figure 13 and 14, the change of traffic demand occurs in VNT1 at 10 minutes, and maximum link utilization of VNT1 becomes higher than 0.5 and the activity of VNT1 becomes low value. We can see that the VNT controller of VNT1 tries to reconfigure VNT1, but physical resource competition occurs and the maximum link utilization keeps high value until at 90 minutes. Moreover, we continue the simulation until 200 minutes, and we confirm that the maximum link

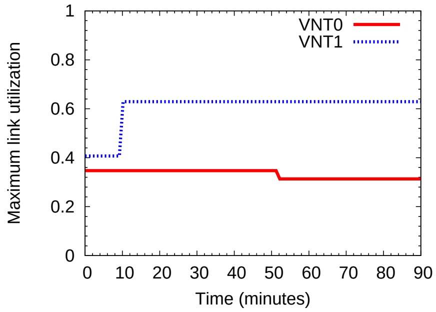

Figure 13. Changes of maximum link utilization without managed self-organization

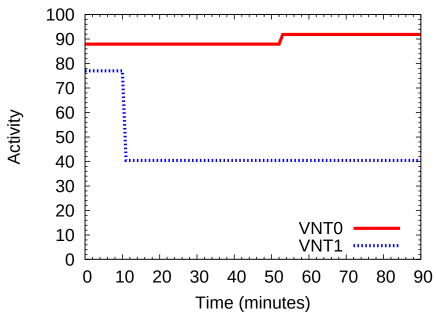

Figure 14. Changes of activity without managed self-organization

utilization is higher than 0.5 until 200 minutes. On the other hand, in Figure 16 and 17, the change of traffic demand occurs in VNT1 at 10 minutes, and the maximum link utilization of VNT1 becomes higher than 0.5 and the activity of VNT1 becomes lower than 50. Then the maximum link utilization of VNT0 is under 0.5. Therefore system manager begins to count the time. VNT controller of VNT1 tries to reconfigure VNT1, but the maximum link utilization do not become lower than 0.5. At time 15, we regard that physical resource competition occurs and VNT1 cannot adapt to the change of traffic demand. Therefore system manager sends feedback to VNT0 and VNT1. We can see activities of VNT0 and VNT1 drop down gradually by $D(t)$. Activities of VNT0 and VNT1 become 0 at time 15. VNT0 and VNT1 work randomly by noise and the whole network is reconfigured. At time 47, VNT0 and VNT1 avoid physical resource competition and configure proper VNTs that can drop maximum link utilization to lower than 0.5. The maximum link utilizations of both VNTs become less than 0.5 by the reconfigurations. We can see the activities of both VNTs return to ordinary value.

Figure 15 and 18 show transition of activities of VNT0 and VNT1. Red point represents the value of α of each VNT at 1 minute, and pink point represents the value of α of each VNT at 90 minute. Blue points represent the value α of each VNT from 2 minutes to 89 minutes. Black arrow means the flow of transition. In Figure 15, we can see activities become

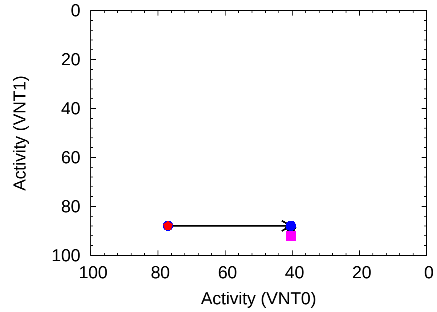

Figure 15. State transition of two VNTs without managed self-organization

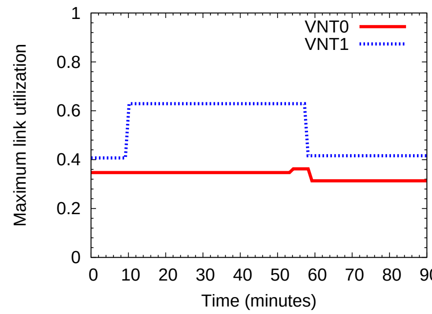

Figure 16. Changes of maximum link utilization with managed self-organization

high value gradually. However the change of traffic demand of VNT1 occurs, and activity of VNT1 become low value. The value does not become high until 90 minutes.

On the other hand, in Figure 18, we can see activities become high value gradually and the activity of VNT1 become low value by the change of traffic demand of VNT1 as the same in Figure 15. However, system manager send feedback to VNT0 and VNT1, and the activities of VNT0 and VNT1 become low value gradually. Therefore, VNT0 and VNT1 are reconfigured by noise of the self-organized VNT control method, and both VNTs configure proper VNT that can drop maximum link utilization to lower than 0.5. At 90 minutes, the activities of VNT0 and VNT1 are high value.

In this way, the managed self-organization of VNT control can avoid physical resource competition and adapt to change of traffic demand quickly by using feedback based on the network activity. However, these simulation settings are tentative. We should consider methods to send feedback based on other information or to set parameters dynamically based on feedback, and we should conduct more simulations. Moreover, the frequency of physical resource competition depends on N_{path}. We should consider setting N_{path} dynamically by feedback to avoid physical resource competition more effectively.

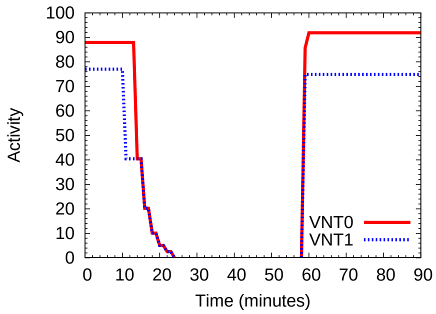

Figure 17. Changes of activity with managed self-organization

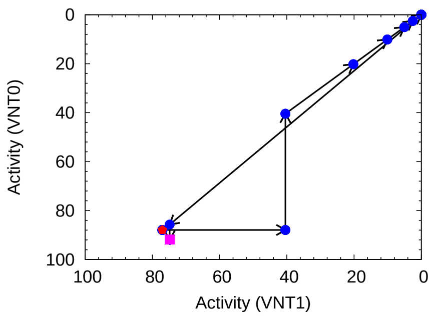

Figure 18. State transition of two VNTs with managed self-organization

VI. Conclusion

Overlay services on existing IP-based networks have expected to be a viable solution to deploy new services without violating underlying protocol standards. However, rapid increase of new overlay services may lead to the increase of traffic growth without control, which will result in a severe degradation of quality of service perceived by existing services. One of approach to overcome this is to prepare multiple VNTs over WDM-based optical networks, and then assign one or more VNTs to each service. In this paper, we proposed managed self-organization of VNT controls, that is a framework to construct multiple VNTs in WDM-based optical network and proposed a control scheme. In our control scheme, the system manager collects the activities that represent the conditions of VNTs, calculate the overall condition of the network, and loosely controls each VNT through the feedback of the information about the overall condition. We conducted computer simulations to evaluate our control scheme. Simulation results showed that managed self-organization of VNT control was adapted to local and small traffic fluctuation through the configuration of each VNT, and was also adapted to global and large traffic fluctuation through the feedback from the system manager.

References

[1] Y. Liu, H. Zhang, W. Gong, and D. Towsley, "On the interaction between overlay routing and underlay routing," in *Proceedings IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2005)*, 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," *OSA Journal of Optical Networking*, vol. 7, pp. 704–719, July 2008.

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

[4] Y. Minami, Y. Koizumi, S. Arakawa, and M. Murata, "Adaptability of virtual network topology control based on attractor selection," in *Proceedings of NOLTA*, Oct. 2009.

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

[6] 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.

[7] 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," *IEEE Journal on Selected Areas in Communications*, vol. 22, pp. 1681–1691, Nov. 2004.

[8] S. Arakawa, M. Murata, and H. Miyahara, "Functional partitioning for multi-layer survivability in IP over WDM networks," *IEICE Transactions on Communications*, vol. 83, pp. 2224–2233, Oct. 2000.

[9] N. Ghani and S. Wang, "On IP-over-WDM integration," *IEEE Communications Magazine*, vol. 38, pp. 72–84, Mar. 2000.

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

[11] J. Comellas, R. Martinez, J. Prat, V. Sales, and G. Junyent, "Integrated IP/WDM routing in GMPLS-based optical networks," *IEEE Network*, vol. 17, pp. 22–27, Mar. 2003.

[12] Y. Koizumi, S. Arakawa, and M. Murata, "An integrated routing mechanism for cross-layer traffic engineering in IP over WDM networks," *IEICE Transactions on Communications*, vol. E90-B, pp. 1142–1151, May 2007.

[13] R. Ramaswami, K. Sivarajan, I. Center, and Y. Heights, "Design of logical topologies for wavelength-routed optical networks," *IEEE Journal on Selected Areas in Communications*, vol. 14, pp. 840–851, June 1996.

[14] Z. Zhensheng and A. Acampora, "A heuristic wavelength assignment algorithm for multihop WDM networks with wavelength routing and wavelength re-use," *IEEE/ACM Transactions on Networking*, vol. 3, pp. 281–288, June 1995.

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

[16] A. Soule, A. Nucci, R. Cruz, E. Leonard, and N. Taft, "Estimating dynamic traffic matrices by using viable routing changes," *IEEE/ACM Transactions on Networking*, vol. 13, pp. 485–498, June 2007.

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

[18] P. Horn, "Autonomic computing: IBM's perspective on the state of information technology," *IBM Corporation*, vol. 15, pp. 1–39, Oct. 2001.

[19] C. Muller-Schloer, "Organic computing: On the feasibility of controlled emergence," in *Proceedings of the 2nd IEEE/ACM/IFIP Hardware/Software Codesign and System Synthesis*, pp. 2–5, Sept. 2004.

[20] U. Richter, M. Mnif, J. Branke, C. Muller-Schloer, and H. Schmeck, "Towards a generic observer/controller architecture for organic computing," *INFORMATIK*, vol. P–93, pp. 112–119, June 2006.