Application-defined control of virtual networks over IP-optical networks

Akeo Masuda, Akinori Isogai, Takashi Miyamura, Kohei Shiomoto and Atsushi Hiramatsu
NTT Network Service Systems Laboratories, NTT Corporation
Musashino-shi, Tokyo, Japan
Email: masuda.akeo@lab.ntt.co.jp

***Abstract*—This paper proposes a novel networking technology by which the network can be controlled according to the demands of the applications. Virtualization and resource management method based on GMPLS enable to independently operate multiple virtual networks upon a common physical network. A server called Physical Network Manager (PN Manager) collects resource and traffic measurement information and constructs the IP topology by setting up optical paths. It provides API in order for Virtual Network Agents (VN Agents) to design and adapt the virtual network topologies in accordance to the application-specific traffic demands. In this paper, we show a design of a VN Agent that provides a basic set of functions that enables application-defined network control. Through experiment conducted upon a nation-wide testbed, we have successfully proven that by VN Agent using the PN Manager API, multiple virtual networks can be independently operated with sufficient performance.**

I. Introduction

In the past decade, we have seen many times a new application appeared and gave new way of using the internet technology. Each time the applications brought change to the traffic pattern of the IP network, network operators had to adapt their network equipments, topology settings and protocol configurations in order to optimize the performance and the cost. However, it is assumed that much more innovative applications with diverse traffic characteristics may emerge in the future. For example, video applications may transmit uncompressed high-definition video streaming with consuming several gigabits per second. On the other hand, communications of sensor nodes and RFID tags emit very small amount of traffic but they require the network to accommodate an enormous number of terminals. Constructing and operating independent physical networks for each of the applications cost too much that we might not be able to deliver the full experience of the applications to the users. Without solving this problem, we cannot benefit from the emerging applications in the future. Therefore, future networks should be installed efficiently by sharing a common physical network, but at the same time it should offer open and individual control that cope with each of the different demands of various applications.

Network virtualization[1] is a technology that slices the resources of a single physical network (PN) in order to overlay multiple virtual networks (VNs). VNs will be used individually as if there are different set of network equipments prepared for them. By introducing virtualization, network providers expect quick launch of cost-efficient network services by sharing physical network resources.

Some technologies are proposed and actually deployed mainly in R&E testbeds for experimental purposes[2], [3]. However, there are still issues left to be solved in order for the practical deployment in carrier's production networks. From the viewpoint explained above, requirements for the network virtualization architecture can be derived as follows:

(1) Isolation: VNs should provide high-performance enough to accommodate bandwidth-intensive applications such as HD class video. Simple IP convergence might cause conflicts between the flows. **(2) Open control:** VN operators desire freedom to design the topology of VNs in accordance to the changing traffic demands. However, conflicts and contentions may happen if operators of each VN directly configure the shared network simultaneously. Furthermore, not many service managers are skilled enough to configure nodes without errors. They only want to know where the nodes are placed, not the detailed network information such as interface devices and running protocols. **(3) Quickness:** although transport layer is already shared in some part of the current carrier networks, usually allocating transport resources to service managers takes a lot of time for inter-department negotiation. To foster novel services and applications, network is required to be provisioned and reconfigured in quick response to the demand. **(4) Dynamic reallocation:** amount of allocated resource to VNs should be optimized according to the change of traffic demands. For example, resources of a VN whose traffic is decreasing can be returned and reallocated to a VN whose traffic is increasing. This helps to optimize the total network cost.

Previously, we had proposed a novel architecture of network virtualization[4]. It assumes an IP-optical network as the physical network consisted of GMPLS enabled IP routers and optical cross-connects (OXCs) that can support signaling-based automatic configuration of optical paths. A server entity called the $PNManager$ is responsible for collecting the network resource information and setting up optical paths by interworking with the network nodes. On the other hand, it provides API for agents to obtain the portion of the resources and utilize them to handle the application flows. Since the PN Manager interfaces the network equipments on behalf of the agents and provide network control functions based on abstracted information that describes the network, various types of agents can be developed easily using the API.

This paper focuses mainly on the design of a system called $VNAgent$ which provides a set of basic functions making use of the PN Manager API, such as traffic matrix estimation and VN topology reconfiguration. These functions enable to design and optimize the VN topology in accordance to the application-specific traffic demands. We also show the results of the experiment that constructed and operated multiple VNs upon a real IP-optical network using a Japanese national R&E testbed. Through the experiment, we confirmed the benefits of our proposal regarding the four requirements addressed above.

Next section illustrates the concept and benefits of our network virtualization architecture, functionalities of the PN Manager, and give description of the PN Manger API. In section III, we show the design of the VN Agent, as well as the operation cycle enabled by its functions. Section IV shows the results of the experiment, and section V concludes the paper.

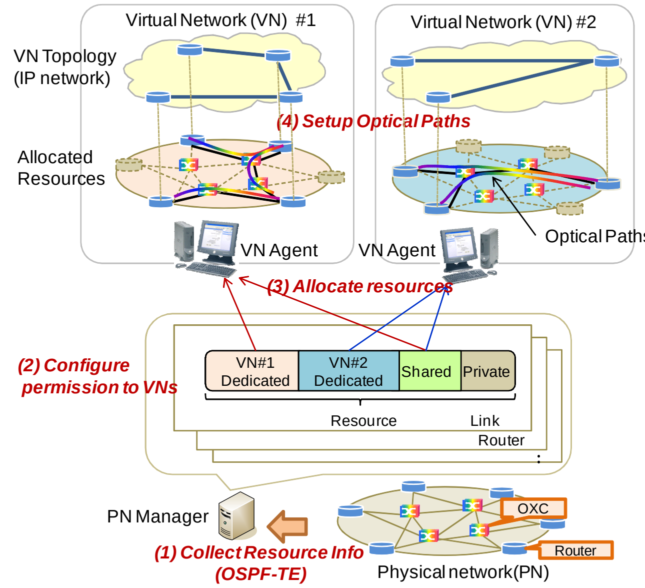

Fig. 1. Construction of VNTs using the resources allocated from the PN Manager.

II. Proposed Architecture of Network Virtualization in IP-Optical Networks

A. Virtual network topology

For the physical network infrastructure, we assume an IP-optical network which is consisted of GMPLS[5] enabled IP routers and optical cross connects (OXCs). This can be prepared with ordinary products that are already available in the market.

IP network of the VN, which we call the $VirtualNetworkTopology(VNT)$[6], is formed by set of optical paths those connect IP router pairs. In other words, an optical path between an IP router pair is set up along the OXCs, but when we look at it in the IP layer, this is a single link between an IP router pair. Therefore, set of them will form an IP network. VNTs constructed by optical paths are shown in Fig. 1.

Optical paths between IP routers are established by triggering the GMPLS signaling (RSVP-TE [7]). On the other hand, IP routers are sliced to support multiple VNs by the logical router functionality. As soon as the optical path for a VN is set up, IP routers are configured in order to associate the optical path to logical router of the VN.

Configuring multiple set of optical paths in this manner enables to provide multiple VNs upon a common PN. Since the VNTs are formed by optical paths that are capable to provide high bandwidth capacity, they can offer sufficient performance to bandwidth-intense applications. Resource isolation in the optical layer ensures to avoid traffic conflicts between VNs. Furthermore, construction and the reconfiguration of VNTs are done quickly because those are formed by optical paths that are able to be setup automatically by GMPLS signaling, instead of doing plugging and configuring each IP link manually.

B. Resource management

Resources used to setup optical paths are routers, OXCs, fibers and wavelengths. Information of the existing resources are automatically collected by listening to OSPF-TE[8] advertisement in GMPLS. For each unit of resource, the administrator of the PN will apply permission for VNs to obtain them. Using the obtained resources, VN operators are allowed to setup optical paths in order to form their own VNT.

Balance of the amount of resources allocated to each virtual network can be modified flexibly by changing the permission of each resource. This enables efficient utilization of the resources in accordance to the change in traffic demands.

C. Physical Network Manager

Total control of the network virtualization is done by a server entity called the $PhysicalNetworkManager$ $(PNManager)$. PN Manager is responsible of resource management and optical path setup control. We assume another server entity of an agent which provide network operation functions such as topology design, traffic monitoring and advanced traffic engineering of the VN. One instance of the agent is responsible for one VN. Operators of VNs run their own agent for operating their VN. Through agents, VN operators can request allocation of the permitted resources to the PN Manager. After VN operators design VNTs by editing the optical path layouts using the allocated resources, agents will request the setup of those paths to the PN Manager. PN Manager is the only system that interfaces the network equipments and executes the path setup on behalf of the agents. According to the request from agents, it triggers the GMPLS signaling (RSVP-TE) to setup optical paths.

Fig. 1 gives also an overview of the proposed architecture. This architecture that decouples the VN operation from the PN management enables the VNs to be operated with high degree of freedom. Generally, VN operators such as application users or service managers and PN operators such as the division of transport layer management are different. Control of the transport layer is not open to the application and service layer. However, in our architecture, VN Operators can control their VN not only in the IP layer but also in the optical layer, for advanced optimization regarding the specific characteristics of the application traffic flows. There are several reasons for this. First, operations of VNs can be done individually and simultaneously without any conflict because the allocated resources for setting up optical paths are isolated in advance. Each VN operator cannot be aware and make use of the resources permitted to other VNs. Furthermore, PN operators do not need to worry about the contention between the network controls of multiple VNs even if they let the VN operators setup optical paths. This is because agents do not interface the network equipments directly. The control of the network equipments is done only by the PN Manager. Finally, operation of the VNs can be done without being aware of the technological details that only the PN operators are familiar to, because PN Manager provides the network resource with simple abstracted information such as nodes and edges. Another benefit of this architecture is that it enables independent elaboration of each of the entities. For example, PN Manager can support new protocols and vender specific control interfaces. Agents can be enhanced to support novel traffic engineering algorithms. For example, there are several methods to estimate the traffic matrices from measured link loads [9]. Optimized VNT regarding the estimated traffic matrix can also be calculated by several methods, such as heuristic algorithm [10], [11] and linear programming [12]. Note that the development of agents is made much easier since they can be designed with treating the abstracted network information. PN Manager can conceal the details of physical network information such as running protocols and device configurations.

D. PN Manager API

PN Manager provides API to let agents obtain resource and request path setup for their VNTs. Description of the methods allowed for the agents to invoke is described in table I.

We expect that various kinds of agents can be developed using this API. One example is the future teleconference system. Innovations in video transmission technologies now enable high-resolution and bidirectional visual communication [13]. We can experience high-presence because of not only the HD class resolution, but also very low delay by uncompressed transmission. Conversation between remote locations can be done without stress since it avoids delay caused by encoding and decoding. However, HD class uncompressed video streaming over IP network consumes about 1.5 Gbps of the bandwidth. Therefore, cooperation between video application and the network control is seriously required. For this, the teleconference system which provides functions such as conference reservation and control of the streaming servers may be extended to become the agent and work as the client of the PN Manager API. Making use of the API, it may request optical path setup for the video transmission.

For another example we can think of is the advanced distributed computing, such as grid computing and datacenters. For large scale grid computing, computers distributed in remote sites need to exchange huge amount of traffic. Therefore, the scheduling system that allocates compter resources to a request may also allocate network resources for setting up high-capacity optical paths between computers at the same time, making use of the PN Manager API. Similarly, future datacenters are expected to handle large and fluctuating traffic between servers. It may optimize the datacenter network by deploying an agent using the PN Manager API in order to dynamically re-provision the path setup between the top-of-rack switches. Furthermore, recent datacenter servers support quick failover by virtual machine migration technologies. Optical path may be used to carry the heavy traffic for

TABLE I
PN MANAGER API

1. Session Management	
Authentication	Request connection to the PN Manager. Checks the userID/password and assigns a VN ID.
2. Collection of Network Information	
Permitted resource	Request information of nodes and TE-links with those properties such as ID, topology, switching capability and reservable bandwidth, that are permitted to the VN as dedicated or shared.
Allocated resource	Request info. of nodes and TE-links that are already allocated to the VN.
Optical paths	Request status of layer-1 LSPs setup.
IP topology	Request info. of layer-3 status such as topology and reservable bandwidth.
MPLS paths	Request status of MPLS LSPs setup upon the IP network.
Measured traffic	Request information of amount of measured traffic at each IP link.
3. Resource allocation	
Obtain resource	Request allocation of nodes and TE-links selected from the permitted resources.
Return resource	Give back nodes and TE-links from the resources allocated to the VN.
4. Path setup	
Create optical path	Request setup of layer-1 LSP along the designated layer-1 nodes and configuration of the routers to associate the LSP as an IP link.
Delete optical path	Request teardown of layer-1 LSP.
Create MPLS path	Request setup of MPLS LSP along the designated layer-3 nodes.
Delete MPLS path	Request teardown of MPLS LSP.
5. Notification	
Traffic monitoring	Notify the agent of a detection of traffic volume exceeded the threshold at a IP link in the corresponding VN.
Network failure	Notify the agent of a detection of a failure of a link or node allocated to the corresponding VN.
6. Resource permission control	
Configure permission	Request assignment of permission of each of the resources for the VNs. This method is only allowed for administratoor of the PN.

synchronization of active and standby servers. Management system of the datacenter may invoke the PN Manager API in case of failover, so that the path will be changed to be setup between the new active server, which had been the standby before failover, and the new standby server.

We have also designed $VNAgent$ as an instance of an agent using the PN manager API. This agent is designed considering a general network operation workflow for carrier's service. Main aim of the functionalities is to design the optimal IP topology according to the given traffic matrix. It provides a basic package of functions needed for traffic engineering. Details are shown in section III.

E. Implementation of the PN Manager

We have implemented the PN Manager that supports the specifications addressed above. Details of the implementation are shown as follows.

1) Interface: API for the agents is implemented to support the methods shown in table I. Methods can be invoked by CORBA, and parameters are describeed in a XML format.

As previously mentioned, information of the optical resources such as nodes, TE-links and the topology are automatically collected from the optical network by listening to OSPF-TE advertisements (LSA packets). For this purpose, PNMngr has direct layer-2 connectivity with the control plane of the optical network. Note that all of the GMPLS nodes are connected to the control plane. It is used for optical resource advertisement (OSPF-TE) and signaling (RSVP-TE) between the nodes.

In addition, PN Manager collects the layer-3 information in order to be aware and notice the agents of the status of the IP network of each VN. For this purpose, PN Manager is also connected to the layer-3 data plane, which is formed by optical paths established by the PN Manager itself. Generally, control packets such as OSPF-TE advertisements and RSVP-TE signaling messages in an IP network are "in-channel" of the data links. Through listening to OSPF-TE LSAs that are advertised in this data plane, PN Manager collects information of the IP network.

PN Manager collects measured traffic of each IP link by requesting interface MIBs via SNMP protocol. This is done periodically in order to be aware of the real-time state of the network load.

For triggering RSVP-TE signaling to setup and teardown optical and MPLS paths, PN Manager directly edits the configuration and invokes commit commands of the IP routers. This is done by telnet CLI through the management plane. PN Manager and all of the nodes have direct layer-2 connectivity to the management plane.

2) Resource management: Link resources are handled in a unit called TE-link which defined in the GMPLS technology. We can describe and utilize the resource to setup optical paths because properties of TE-link provides sufficient information of the link such as connected node address, link address, maximum and minimum reservable bandwidth, switching capability (fiber, lambda, TDM and packet), shared risk link

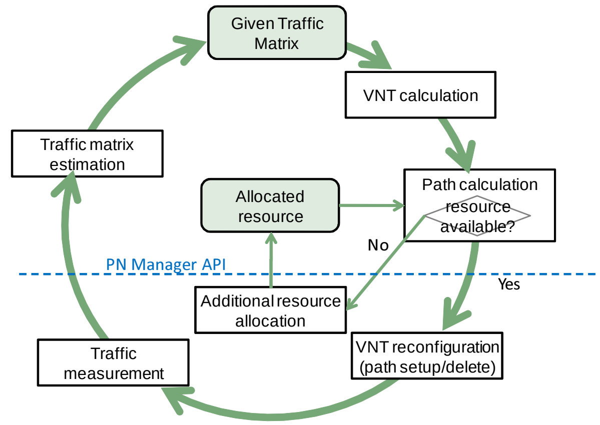

Fig. 2. Assumed cycle of the VN operation.

group, and so on. As mentioned above, each of the resources are assigned permission for the VNs. Resources are permitted as either dedicated or shared. Shared resources can be noticed by multiple VNs, but it will be allocated to only one of that VNs. Sharing the unallocated resources enables capital cost reduction of the physical network infrastructure, by sharing the redundant resource that should have been prepared for each of the network service if no virtualization is adopted.

III. Design of VN Agent

In this section, we describe a design of the $VNAgent$ as an instance of an agent that is a user of the PN Manager API. This agent is designed to enable a general network operation cycle for carrier's service. We focus on designing the optimal VNT (IP topology) according to the given traffic matrix. Optimal VNT is calculated in order to maximize the amount of traffic that the VN can accommodate, while minimizing the required amount of network resources. Resource required for a VN may be different at different time of the day, week or year. A part of resource used in a VN at a certain term may be given back to the PN and may be utilized in another VN at another term. Therefore, optimizing the VNT and dynamic adaptation to the traffic changes lead to reduce the total network resource required in the physical network.

A. VN operation cycle

We assume an operation cycle as shown in Fig.2. First, VN operator designs the initial VNT manually considering the assumed pattern of the traffic demand. After the VNT is constructed, it will be reconfigured according to the measurements of injected traffic. This cycle of measurement and reconfiguration is the fundamental part of the assumed network operation. Within the cycle, VN requests additional resource to the PN when it lacks of resource to accommodate the traffic demand.

B. Functions

VN Agent provides a basic package of functions needed for traffic engineering. The operation cycle in Fig.2 is one of the use case enabled by making use of them. Major functions are listed as follows.

1) Resource Management: This function is triggered manually by the operator. It requests information of resources allocated to the VN, and stores in the database. Operators can view the resources currently allocated to the VN. It also requests and let the operator confirm the resources that are not allocated to any of the VN, and are assigned permission to the VN. According to the request by the operator, it requests additional allocation of resources to the PN Mangaer. It gives back of them to the PN Manager as well.

2) VNT Editorial: This functions allows the operator to design and configure the VNT manually. Using the editorial window of the GUI tool, operators can draw routes of the layer-1 or MPLS LSPs upon a network map, clicking the nodes along the desired route. Note that if the operator selects only the endpoints of the LSP, VN Agent will automatically compute the route by the path computation function described in the following. Route of LSPs drawn on the window or those already established can be easily modified by drag-and-drop of the lines. Operator selects the commit command when finished the VNT editorial. VN Agent will analyze the difference between the edited and current path setup, and creates a scenario of invoking the PN Managaer API. After the operator confirms the scenario, VN Agent will send messages such as path setup, teardown, modify requests to the PN Manager.

3) Path Computation: This function provides path computation of the layer-1 and MPLS LSP according to the request from the operator. Properties of the requested path that the operator can specify is shown in table II. Using the GUI tool for VN Agent, source/destination router IDs are automatically set by clicking the node icons on the network map. LSP can be added a redundant path for failover by path protection options. Protection paths can be found considering not sharing the nodes and/or the links with the primary path. For considering SRLG (Shared Risk Link Groups), we use the WSRLG (Weighted Shared Risk Link Group) algorithm. For advanced control, operators can specify the switching capability, encoding type and the administrative group (resource color), if needed.

TABLE II
LSP PROPERTIES FOR PATH COMPUTATION.

Property	Layer-1 LSP	MPLS LSP
Source/destination router ID	m	m
Bandwidth	m	o
Path protection	o	o
Switching Capability	o	
Encoding Type	o	
Administrative Group		o

m: mandatory o: optional

4) Network State Monitoring: VN Agent obtains measured traffic information by periodical and/or manual polling to the PN Manager, or receiving notification of exceeding the threshold, from the PN Manger. Operators can be aware of the amount of measured traffic at each of the IP links. It can be seen not only as a list, but also on the network map. Color of the link will change according to the measured traffic. For example, it will turn to red when traffic exceeds the upper threshold, and turn to blue when traffic falls short of the lower threshold. VN Agent is notified of the failure of links from the PN Manger as well. By these functions, operators can always be aware of the network state and utilization, and take quick action for failures or dynamic fluctuation of the traffic demand.

5) Traffic Matrix Estimation: Traffic matrix is the amount of data traffic between every pair of network ingress/egress points. Understanding traffic matrices is crucial for traffic engineering. For example, we cannot determine which flows to be rerouted in order to balance the link loads if we only know the traffic amount at each link. VN Agent provides an traffic matrix estimation function based on the tomogravity method [14] that gives an calculation for a given set of link traffic measurements and routing information. Details of the algorithm and performance evaluation regarding the computing time and accuracy is described in [15]. Result of the estimation can be viewed by the operator graphically on the GUI. It is also used in the VNT computation explained in the next part.

6) VNT Reconfiguration: This function computes and proposes to the operator of the best way to change the VNT configuration in response to the dynamic change of traffic demands. It is done by considering the estimated traffic matrix, but does not compute the optimal VNT for that. Redesigning the optimal VNT without taking the current VNT into account may cause drastic change of the optical paths. Furthermore, traffic matrix estimation may sometimes make errors. Operators cannot take risk of making a drastic reconfiguration of the VNT fully depending on the estimation results. Hence, we employ an heuristic algorithm that gradually changes the VNT. Essentially, it is a step-by-step operation coping for each of the highly loaded link. It tries to balance the link load by selecting a flow that can be rerouted, from flows traversing the loaded link. A new route may be created by setting up a new optical path. Result of traffic matrix estimation is used here in order to detect the flows coming into this link. By dividing the whole VNT transition sequence into multiple transitions, estimation errors are calibrated at each stage by using network state information of prior stages. Our approach tries to increase the constraint conditions for traffic matrix estimation by introducing partial reconfiguration, and to relax the impact of estimation errors by limiting the number of optical paths reconfigured at each stage. Details of the algorithm are described in [16], and Fig. 3 shows an example of the reconfiguration.

IV. WIDE-AREA EXPERIMENT

In January and February 2011, we have conducted an experiment to show the proof of our concept and confirm the benefits of our network virtualization architecture. The IP-optical network consisted of nine IP routers and seven OXCs was implemented upon JGN2plus[17], the Japanese national R&E testbed.

Three VNs were constructed upon the IP-optical network. Operation to assign permission of each of the resources to the VNs was done through the GUI tool of the PN Manager, and

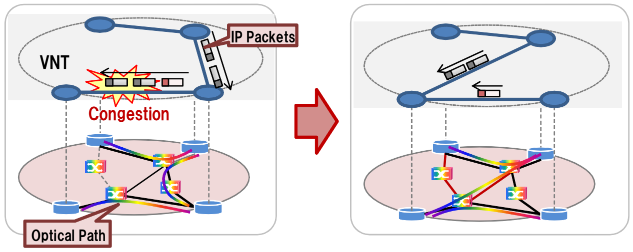

Fig. 3. Dynamic reconfiguration of VNT adaptive to the change in traffic patterns.

it took only about 30 minutes. Operations to form each of the VNTs by VN operators were also done within about 30 minutes. Processing time of a single optical path setup was 15.8 [sec]. From these results we confirmed the quickness of the deployment of VNs.

Three different parties had done their experiment using the VNs. Each of them had carried out testing, experiments, measurement and demonstrations during the experiment event which lasted about a month. They transmitted many kinds of traffic, and frequently changed the VNT by setting up and tearing down the optical paths. We confirmed that these activities were done simultaneously without any problem regarding the shared control of the IP-optical network.

In VN#1, an experiment of dynamic reconfiguration of the VNT was done. In accordance to the measured traffic, optical paths were set up and tore down in order to change the VNT to an optimal topology which was automatically calculated by the optional function of VN Agent. VN#2 transmitted compressed HDTV video with about 300 Mbps for live broadcasting in actual TV program. Furthermore, VN#3 transmitted uncompressed 4K HDTV video stream consuming about 6.4 Gbps. Exclusive use of the optical paths provided sufficient performance for them. Main objective of the experiment in VN#3 was verifying the high transmission capacity of an advanced wireless communication system. The network resource of 10 Gbps connection and the wireless communication system were desired to be used also in the experiment upon VN#2. Therefore, we have dynamically changed the resource allocation so that these resources were moved from VN#3 to VN#2. Through this operation, we confirmed that our network virtualization concept enables dynamic reallocation of the network resources. Users of VN#3 were engineers of wireless devices who are not skilled of optical network technologies. They were not aware of the configuration of the network equipments. However, only with a brief lecture of the usage of GUI tool of the VN Agent, they were able to continue switching over the transmission path between the primary path using the wireless connection and the backup path by setting up and tearing down the optical paths by their own. This proves the ability of open and user-friendly VNT control enabled by our network virtualization architecture and the functions of VN Agent.

V. Conclusion

Based on our proposed architecture of network virtualization, we have implemented PN Manager and VN Agent that construct and operate multiple virtual networks. Experiment had successfully confirmed that the proposed architecture and implementation were able to provide functionalities to meet the four requirements; isolation, open control, quickness and dynamic reallocation. For further study, we plan to enhance our architecture to support multi-layer, multi-domain virtual networks. We also pursue scalability in terms of number of network nodes and number of VNs that the PN Manager can support.

References

[1] N. M. M. K. Chowdhury and R. Boutaba, "A survey of network virtualization," *Computer Networks*, vol. 54, no. 5, pp. 862–876, 2010.
[2] "geni: Exploring Networks of the Future," http://www.geni.net/.
[3] T. Anderson, L. Peterson, S. Shenker, and J. Turner, "Overcoming the internet impasse through virtualization," *Computer*, vol. 38, pp. 34–41, April 2005.
[4] T. Miyamura, A. Masuda, and K. Shiomoto, "High-performance network virtualisation for multilayer GMPLS networks," *International Journal of Communication Networks and Distributed Systems*, vol. 6, no. 3, pp. 283–301, 2011.
[5] E. Mannie, "Generalized Multi-Protocol Label Switching (GMPLS) Architecture," RFC 3945 (Proposed Standard), Internet Engineering Task Force, Oct. 2004.
[6] K. Shiomoto, D. Papadimitriou, J. L. Roux, M. Vigoureux, and D. Brungard, "Requirements for GMPLS-Based Multi-Region and Multi-Layer Networks (MRN/MLN)," RFC 5212 (Informational), Internet Engineering Task Force, Jul. 2008.
[7] L. Berger, "Generalized Multi-Protocol Label Switching (GMPLS) Signaling Resource ReserVation Protocol-Traffic Engineering (RSVP-TE) Extensions," RFC 3473 (Proposed Standard), Internet Engineering Task Force, Jan. 2003.
[8] K. Kompella and Y. Rekhter, "OSPF Extensions in Support of Generalized Multi-Protocol Label Switching (GMPLS)," RFC 4203 (Proposed Standard), Internet Engineering Task Force, Oct. 2005.
[9] A. Gunnar, M. Johansson, and T. Telkamp, "Traffic matrix estimation on a large ip backbone: a comparison on real data," in *Proceedings of the 4th ACM SIGCOMM conference on Internet measurement*, ser. IMC '04. New York, NY, USA: ACM, 2004, pp. 149–160.
[10] K. Shiomoto, E. Oki, W. Imajuku, S. Okamoto, and N.Yamanaka, "Distributed virtual network topology control mechanism in GMPLS-based multiregion networks," *IEEE Journal on Selected Areas in Communications*, vol. 21, pp. 1254–1262, 2003.
[11] S. Gieselman, "Minimum-cost virtual-topology adaptation for optical wdm mesh networks," *Proceedings of IEEE ICC, June 2005*, vol. 3, pp. 1787–1791, 2005.
[12] S. Salsano, F. Ricciato, M. Listanti, and A. Belmonte, "Off-line configuration of a mpls over wdm network under time-varying offered traffic," in *INFOCOM*, 2002.
[13] "Internet video studio system for HDTV production," http://www.i-visto.com/.
[14] Y. Zhang, M. Roughan, N. Duffield, and A. Greenberg, "Fast accurate computation of large-scale ip traffic matrices from link loads," in *Proceedings of the 2003 ACM SIGMETRICS international conference on Measurement and modeling of computer systems*, ser. SIGMETRICS '03. New York, NY, USA: ACM, 2003, pp. 206–217.
[15] T. Takeda and K. Shiomoto, "Traffic matrix estimation in large-scale ip networks," in *The 17th IEEE Workshop on Local and Metropolitan Area Networks (LANMAN 2010)*, 2010.
[16] Y. Ohshita, 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, February 2010.
[17] "JGN2plus: an advanced testbed network for R&D," http://www.jgn.nict.go.jp/.