Proceedings of the International MultiConference of Engineers and Computer Scientists 2014 Vol I,
IMECS 2014, March 12 - 14, 2014, Hong Kong

Virtual Network Construction with K-Shortest Path Algorithm and Prim's MST Algorithm for Robust Physical Networks

Yasuhiro Urayama and Takuji Tachibana

***Abstract*—Network virtualization has attracted considerable attention currently, and it has been expected to be utilized as new-generation network technology. For the network virtualization, improved network robustness is indispensable for virtual networks and a physical network. Moreover, it is important to utilize network resources in the physical network and construct a lot of virtual networks. In this paper, we propose a new virtual network construction so as to construct a lot of virtual networks. Our proposed method utilizes K-shortest path algorithm and Prim's MST algorithm. At first, in the proposed method, K topologies of virtual networks are designed by using the K-shortest path algorithm and the Prim's MST algorithm instead of the KMB algorithm. Because many topologies are designed, it is expected that a request of constructing a virtual network can be accepted with a high probability. We evaluate the performance of our proposed method for three different topologies with simulation. We investigate the effectiveness of our proposed method by comparing this method with the conventional method. In addition, we evaluate the computation time of the K-shortest path algorithm. In numerical examples, we show that our proposed method is effective in a large-scale backbone network.**

***Index Terms*—Network virtualization, Topology design, Robustness, K-shortest path algorithm, Admission control**

I. Introduction

NETWORK virtualization has attracted considerable attention currently, and this technology has been expected to be utilized as a new-generation network technology [1], [2]. For the network virtualization, multiple virtual networks can be constructed over a physical network by using network resources. The constructed virtual networks are provided with users, and each user can utilize the provided virtual network independently of other virtual networks.

It is important to consider how multiple virtual networks should be constructed over a physical network. This is because the amount of available resources is limited and virtual networks have to share the limited resources. Therefore, some methods for virtual network construction have been studied in [3], [4], [5], [6]. These methods consider several kinds of issues; the minimization of traffic load, the simplicity of resource allocation, the independence of network topology, the revenue from users, and so on.

Here, in recent network management, one of the important challenges is to satisfy service level agreement (SLA) requirements of users [7], [8]. For the network virtualization, improved network robustness is also indispensable for virtual networks and physical network. In [9], in order to construct virtual networks while keeping the robustness of a physical network, authors have considered a virtual network construction. In this method, a performance metric of the network robustness, which is called *network criticality*, is calculated for the physical network by using an extended network graph. Then, a topology of each virtual network is designed according to KMB algorithm so as not to reduce the robustness of the physical network.

This topology design of virtual network is effective to keep the robustness of the physical network. On the other hand, in this method, only one topology is designed for each virtual network. If this topology cannot satisfy construction conditions about the amount of resources and the robustness of physical network, this construction is rejected immediately. Therefore, some virtual networks may not be constructed on the physical network even if other topologies can satisfy the construction condition.

In this paper, we propose a new virtual network construction by extending the method in [9] so as to construct a lot of virtual networks. Our proposed method utilizes K-shortest path algorithm and Prim's MST algorithm instead of the KMB algorithm. At first, in the proposed method, K topologies of virtual networks are designed by using the K-shortest path algorithm and the Prim's MST algorithm. Because many topologies are designed, it is expected that a request of constructing a virtual network can be accepted with a high probability. We evaluate the performance of our proposed method in three different topologies with simulation. We investigate the effectiveness of our proposed method by comparing this method with the conventional method. In addition, we evaluate a computation time of the K-shortest path algorithm.

The rest of this paper is organized as follows. Section II introduces two performance metrics in order to evaluate network robustness. Section III explains the detail of our proposed virtual network construction. Section IV shows some numerical examples and Sect. V describes conclusions and future work.

II. Related Work

A. Robustness Evaluation of Physical Network for Network Virtualization

In this subsection, we explain how to evaluate robustness of a physical network to construct virtual networks [9].

Now, we denote the number of nodes as M and denote the number of links as N. Here, $V = \{v_1, \cdots, v_M\}$ and $E = \{e_1, \cdots, e_N\}$ represent the set of nodes and the set of links, respectively. In addition, let $w_i^v > 0$ be a weight that

Y. Urayama is with Graduate School of Engineering, University of Fukui, Fukui 910–8507, Japan. Email: yasuhiro-u@network.fuis.u-fukui.ac.jp
T. Tachibana is with Graduate School of Engineering, University of Fukui, Fukui 910–8507, Japan. Email: takuji-t@u-fukui.ac.jp

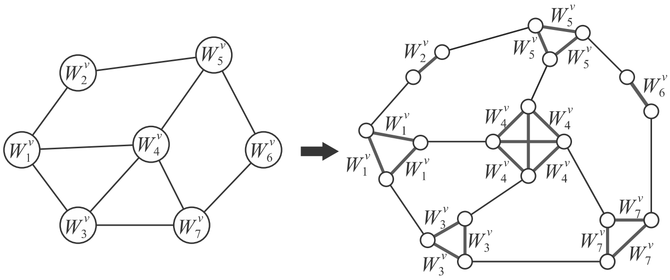


Fig. 1. Generated new network graph.

represents the amount of resources of node v_i and let $w_i^e > 0$ be a weight that represents the amount of resources of link e_i. From these weights, the two sets of weights are denoted as $W^V = \{w_1^v, \cdots, w_M^v\}$ for nodes and $W^E = \{w_1^e, \cdots, w_N^e\}$ for links. Finally, we denote a graph of the physical network as $G = (V, E, W^V, W^E)$.

Here, we focus on node v_i $(i = 1, \cdots, M)$ whose weight is w_i^v. Let the number of links that are connected with v_i be $h_i \geq 1$. In this case, the node v_i is divided into h_i nodes $\{v_i^1, \cdots, v_i^{h_i}\}$. Then, a new generated node for the original node, v_i^k $(k = 1, \cdots, h_i)$, is connected with a new generated node for another original node v_j, v_j^l $(j \neq i)$, by using an original link weight of this link is equal to the weight of the original link. Moreover, the h_i nodes for the original node v_i are connected with each other by using $h_i(h_i - 1)/2$ new links as a full-mesh network. The weights of these links are equal to the weight of the original node v_i, i.e., w_i^v. Thus, the weight of each node is denoted as the link weight in the new network graph (see Fig. 1).

For the obtained network graph, weight matrix is denoted as W^V and diagonal matrix is denoted as D^V. In this case, a new Laplacian matrix L^V is given by

$$L^V = D^V - W^V. \tag{1}$$

Finally, network criticality τ that is a performance metric about network robustness is derived from

$$\tau = 2n^V \mathrm{Tr}(L^{V+}), \tag{2}$$

where n^V is the number of nodes in the new network graph.

B. Virtual Network Construction with Network Criticality

In [9], a virtual network construction with network criticality has been proposed. This method can be utilized in a simple system of the network virtualization. In the system, an user who would like to utilize a virtual network sends a request for a desired virtual network to a service provider. Figure 2 shows how the service provider provides a virtual network with the user.

Here, the request includes any nodes that should be included in the desired virtual network and the amount of network resources needed for the virtual network. The service provider has to provide a virtual network with the user over the physical network so as to satisfy both robustness of the physical network and the user's request. In the conventional method, in order to find the optimal topology of a virtual network, KMB algorithm is utilized because this problem is a Steiner minimal tree (SMT) problem [10].

Then, for the obtained topology, the service provider checks whether the virtual network can be constructed or not with the remaining network resources. If the amount of remaining network resources is insufficient, the user's request is rejected. Otherwise, assuming that the virtual network was provided and the network resource in the physical network was reduced, the service provider evaluates the network criticality of the remaining physical network from (2). By using an admission control, the virtual network is constructed and provided if the network criticality of the remaining physical network does not exceed the pre-defined threshold. This admission control can keep the robustness of the remaining physical network.

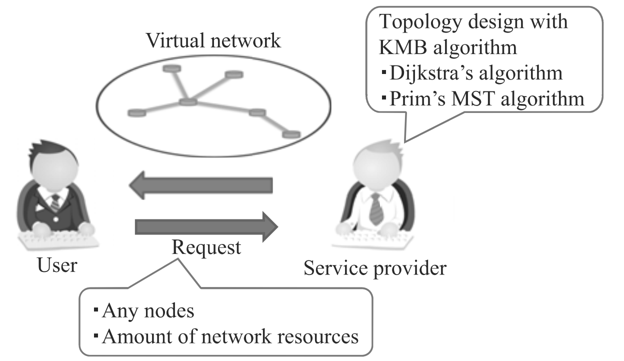


Fig. 2. Conventional virtual network construction with network criticality.

C. Performance Issue of Virtual Network Construction with Network Criticality

As explained in the previous subsection, in the conventional virtual network construction, the topology of virtual network is determined with the KMB algorithm. Here, the KMB algorithm consists of Dijkstra's algorithm and Prim's MST algorithm, and hence only one topology of the virtual network is designed. If the amount of resources is insufficient on the designed topology, the virtual network cannot be constructed. In addition, if the network robustness of the remaining physical network exceeds the threshold, the virtual network cannot be constructed. Therefore, some virtual networks may not be constructed on the physical network even if other topologies can satisfy the construction condition.

III. Proposed Method

In this paper, in order to resolve the above-mentioned performance issue for the conventional method, we propose a new virtual network construction with K-shortest path algorithm and Prim's MST algorithm.

A. Overview

Our proposed method is utilized in a simple network virtualization system, as is the case with [9]. In the following, due to simple explanation, we assume that network robustness of a physical network has already been evaluated from (2) and network criticality τ has already been obtained. In addition, we assume that threshold τ_{th} for the network criticality has been determined.

In this system, a service provider constructs a virtual network on a physical network according to a user's request and provides the virtual network with users. In order to construct and provide a virtual network, the service provider utilizes our proposed method as follows (see Fig. 3).

ISBN: 978-988-19252-5-1
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

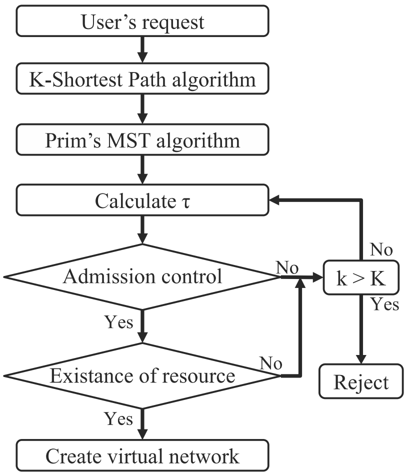


Fig. 3. A flowchart of our proposed algorithm.

1) The service provider receives a request of virtual network construction from a user. This request includes any nodes that should be included in the desired network and the amount of network resources needed for the virtual network. Go to step 2.
2) The service provider designs K topologies of the virtual network by using K-shortest path algorithm and Prim's MST algorithm instead of KMB algorithm. In the following, a virtual network that is constructed with the ith shortest path is called the ith virtual network. Then, i is set to one, and go to step 3.
3) The service provider assumes that the ith virtual network was provided. From (2), the service provider computes the network criticality τ_{now} of the remaining physical network for evaluating its network robustness. Go to step 4.
4) If the network criticality of the remaining physical network is equal to or smaller than the threshold, i.e., $\tau_{now} \leq \tau_{th}$, the service provider constructs and provides the ith virtual network. Otherwise, go to step 5.
5) In this step, at first, i increases by one. If i is equal to or smaller than K, return to step 3. If i is larger than K, the user's request is rejected.

B. A user's request for a virtual network

In our proposed method, a user sends a request to the service provider when the user would like to utilize a virtual network. This request includes the information about nodes that have to be included in the provided virtual network. Moreover, it includes the information about the amount of resources that is used in the virtual network. Here, let H $(0 < H \leq M)$ be the number of nodes in the user's request, $\{v_1^*, v_2^*, \cdots, v_H^*\}$ be a set of the nodes, and $l > 0$ be the desired amount of resources. Finally, the user's request is represented as $(v_1^*, v_2^*, \cdots, v_H^*, l)$.

C. Topology Design of Virtual Network with K-shortest path Algorithm and Prim's MST Algorithm

A topology of virtual network has to be designed so as to avoid the significant increase of network criticality for the remaining physical network. Here, let w_i^e and w_i^v be the link cost of a new network graph as explained in subsection II-A. These costs are changed into the cost that denotes the increase rate of the network criticality per the unit of resources as follows:

$$w_i^e \leftarrow \tau + w_i^e \frac{\partial \tau}{\partial w_i^e}, \quad w_i^v \leftarrow \tau + w_i^v \frac{\partial \tau}{\partial w_i^v}. \qquad (3)$$

Because the new costs represent the impact on the network robustness, the topology whose cost is small should be designed for each virtual network. Here, by considering the performance issue of the conventional method, multiple topologies should be designed in order to accept a lot of user's requests. Therefore, the K-shortest path algorithm and the Prim's MST algorithm are used for the topology design.

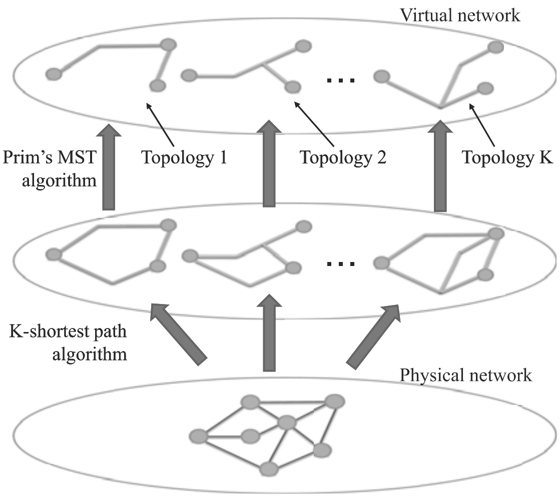


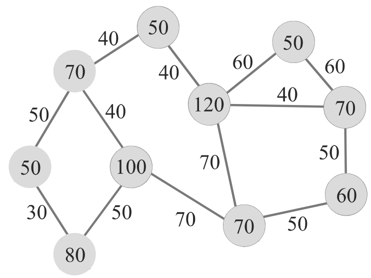
Fig. 4. Topology design of virtual network with K-shortest path algorithm and Prim's MST algorithm.

Figure 4 shows how K topologies can be designed by using our proposed method. As shown in this figure, at first, K different paths are found between any two requested nodes with the K-shortest path algorithm. From these shortest routes, each topology is constructed. Note that these topologies include the requested nodes. Then, for the topologies, Prim's MST algorithm is used to construct a minimum spanning tree [10]. Finally, K steiner trees including the steiner minimum tree are designed. For these topologies, an admission control is performed as explained in subsection III-A.

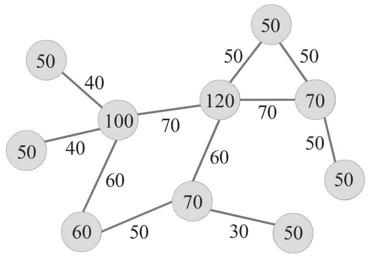
IV. Numerical Examples

In this section, we evaluate the performance of our proposed method with simulation. In the following, we consider three different physical networks shown in Figs. 5(a), (b), and (c). In order to utilize a virtual network, user's requests arrive at a service provider for each physical network according to a Poisson process with rate λ. For each user's request, the service provider performs our proposed method with K-shortest path algorithm and Prim's MST algorithm. For the user's request, the number H of nodes that are requested by each user is selected from three to $the\ number\ of\ nodes{-}1$ at random. We also assume that the H nodes are selected at random in each network. For each topology, the amount l of resources for a virtual network is selected from one to the minimum amount of resources at random.

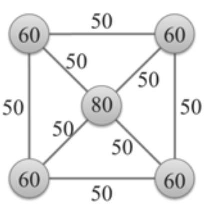
ISBN: 978-988-19252-5-1
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)



(a) Topology 1.



(b) Topology 2.



(c) Topology 3.

Fig. 5. Three different topologies of physical networks.

If the request of a user is accepted, a virtual network is provided with the user. We assume that the utilization time of each virtual network follows an exponential distribution with mean 1.0. When the utilization of a virtual network is completed, its network resources are returned immediately to the remaining physical network. For the performance comparison, we evaluate the performance of the conventional virtual network construction in [9].

A. Impact of arrival rate λ

First, we evaluate the performance of the proposed method in topology 1 shown in Fig. 5(a).

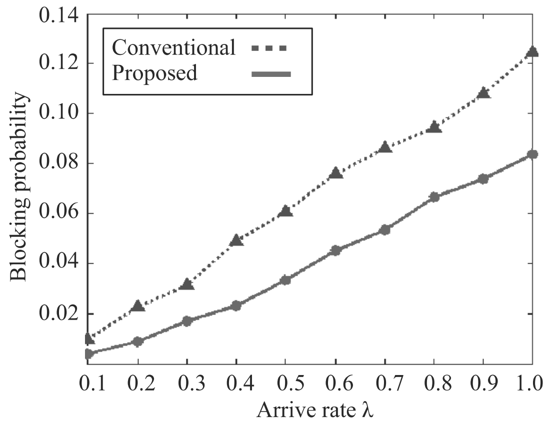
Figures 6(a) and (b) show the blocking probability and the network robustness of a virtual network against the arrival rate λ, respectively. From these figures, we find that the blocking probability of the conventional method is the highest and the blocking probability can be reduced by using our proposed method.

On the other hand, for the network robustness of a virtual network, the network robustness of the conventional method is somewhat small. Therefore, the network robustness is somewhat degraded by using our proposed method. However, this degradation is not so large, and hence the proposed method is effective to decrease the blocking probability while keeping the robustness of the physical and virtual networks.

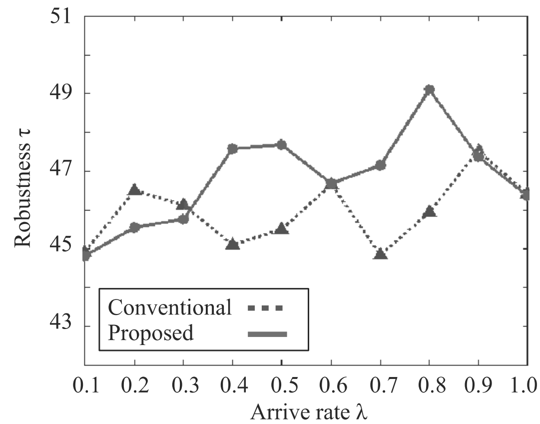
B. Impact of network topology

Next, we evaluate the performance of our proposed method in other topologies. Figure 7 shows the performance of our proposed method in topology 2 shown in Fig. 5(b) and Fig. 8 shows the performance of our proposed method in topology 3 shown in Fig. 5(c).

From Fig. 7, we find that regardless of arrival rate λ, the blocking probability for our proposed method is almost the same as that for the conventional method. Moreover, the network criticality of the virtual network does not change so much. Therefore, for the topology 2, it is shown that our proposed method is not effective. This is because the number



(a) Blocking probability vs. arrival rate.



(b) Network robustness of virtual networks vs. arrival rate.

Fig. 6. Performance of each method in network topology 1.

of routes between any two nodes is small and the K-shortest path algorithm is not available frequently.

On the other hand, by comparing Fig. 8 with Fig. 7, the effectiveness of our proposed method in the topology 3 is larger than that in the topology 2. This is because the number of routes between any two nodes in the topology 3 is larger than that in the topology 2. However, the effectiveness in the topology 3 is smaller than that in the topology 1 because the size of the topology (the numbers of nodes and links) is small.

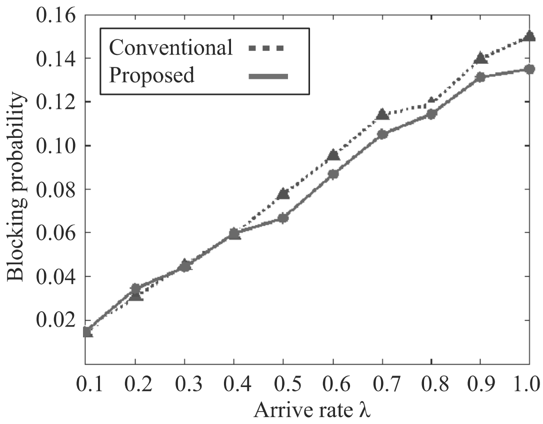
From the above, it is clear that the performance of our proposed method depends on the topology of physical network. However, the utilization of our proposed method never degrades the performance of virtual network construction. Moreover, our proposed method is effective in a large-scale backbone network that has a large number of nodes and links and has multiple routes between any nodes.

C. Computation time of each method

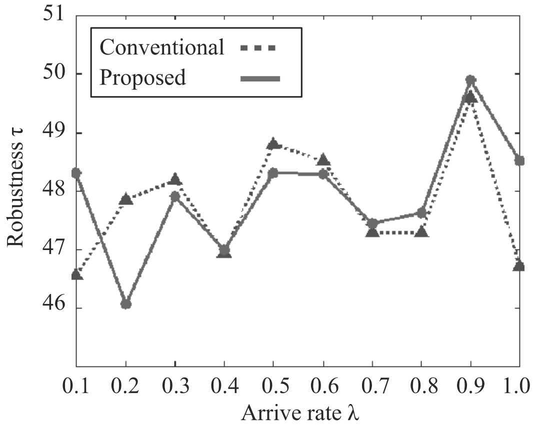
Finally, for our proposed method, we measure the processing time of our proposed method. The measurement is performed with Matlab on a PC with Intel Core 2 Quad Q8400 2.66 GHz and 4.0 GB RAM.

Table I shows computation time [sec] of the K-shortest path algorithm in each topology. From this table, we find that the computation time for each process depends on the topology of physical network. Especially, the computation time is the largest for the topology 1. This is because there are many routes between any two nodes and it takes a long

ISBN: 978-988-19252-5-1
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

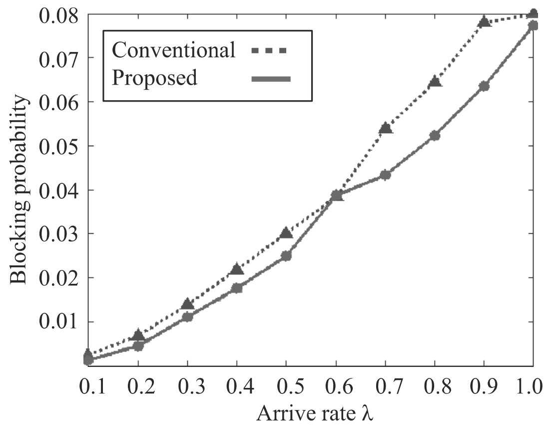


(a) Blocking probability vs. arrival rate.

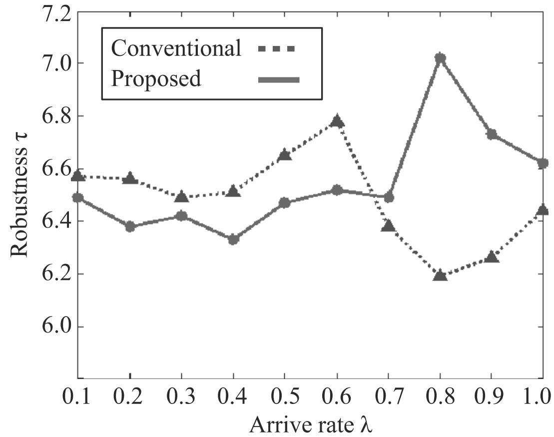


(b) Network robustness of virtual networks vs. arrival rate.

Fig. 7. Performance of each method in network topology 2.



(a) Blocking probability vs. arrival rate.



(b) Network robustness of virtual networks vs. arrival rate.

Fig. 8. Performance of each method in network topology 3.

TABLE I
PROCESSING TIME OF OUR PROPOSED METHOD.

	Topology 1	Topology 2	Topology 3
K-shortest path [sec]	19.35	2.64	0.0012

time to derive the K shortest path. However, this computation time is smaller than 20.0 seconds and this is not so large for the virtual network construction.

V. Conclusions and Future Work

In this paper, we proposed a new virtual network construction method to utilize the network resource effectively and construct a lot of virtual networks. This proposed method utilizes the K-shortest path algorithm and the Prim's MST algorithm. We evaluated the performance of our proposed method with simulation and compared its performance with the performance of the conventional method. From numerical examples, we found that a larger number of virtual networks can be provided with users by using our proposed method while keeping the network robustness of the physical networks in some topologies. Moreover, we found that the performance of our proposed method depends on the topology of a physical network significantly. Our proposed method is effective in a large-scale backbone network. In our future work, we will extend the K-shortest path algorithm to decrease the computation time.

Acknowledgement

This work was partly supported by the Strategic Information and Communications R&D Promotion Programme of the Ministry of Internal Affairs and Communications, JAPAN.

References

[1] GENI, http://www.geni.net/.
[2] PlanetLab, http://www.planet-lab.org/.
[3] Y. Zhu and Mostafa Ammar, "Algorithms for Assigning Substrate Network Resources to Virtual Network Components," in *Proc. IEEE INFOCOM 2006*, Apr. 2006, pp. 1-12.
[4] M. Yu, Y. Yi, J. Rexford, and M. Chiang, "Rethinking Virtual Network Embedding: Substrate Support for Path Splitting and Migration," *ACM SIGCOMM Computer Communication Review*, vol. 38, no. 2, pp. 17-29, Apr. 2008.
[5] A.H. Dekker and B.D. Colbert, "Network Robustness and Graph Topology," in *Proc. Australasian Computer Science Conference 2004 (ACSC 2004)*, Jan. 2004, pp. 359-368.
[6] J. He, Z.S. Rui, Y. Li, C.Y. Lee, J. Rexford, and M. Chiang, "DaVinci: Dynamically Adaptive Virtual Networks for a Customized Internet," in *Proc. ACM CoNext 2008*, Dec. 2008.
[7] R.S. Graciȧ, Y. Labit, J. D. Pascual, and P. Owezarski, "Towards an Efficient Service Level Agreement Assessment," in *Proc. IEEE INFOCOM 2009*, Apr. 2009, pp. 2581-2585.
[8] A. Tizghadam and A.Leon-Garcia, "Autonomic Traffic Engineering for Network Robustness," *IEEE Journal on Selected Areas in Communication*, vol. 28, no. 1, pp. 1-12, Jan. 2010.
[9] M. Mori, T. Tachibana, K. Hirata, and K. Sugimoto, "A Proposed Topology Design and Admission Control Approach for Improved Network Robustness in Network Virtualization," in *Proc. IEEE Globecom 2011*, Dec. 2011.
[10] L. Kou, G. Markowsky, and L. Berman, "A Fast Algorithm for Steiner Trees," *Acta Informatica*, vol. 15, no. 2, pp. 141-145, June 1981.