Optimal Standby Virtual Routers Selection for Node Failures in a Virtual Network Environment

Xuan Liu[1] and Deep Medhi[1]
Networking Research Lab, University of Missouri–Kansas City, USA
{xuan.liu, dmedhi}@umkc.edu

***Abstract*—Network virtualization allows flexibility to configure virtual networks in a dynamic manner. In such a setting, to provide resilient services to virtual networks, we consider the situation where the substrate network provider wants to have standby virtual routers ready to serve the virtual networks in the event of a failure. Such a failure can affect one or more virtual routers in multiple virtual networks. The goal of our work is to make the optimal selection of standby virtual routers so that virtual networks can be dynamically reconfigured back to their original topologies after a failure. We present an optimization formulation and a heuristic for this problem. By considering a number of factors, we present numerical studies to show how the optimal selection is affected. The results show that the proposed heuristic's performance was close to the optimization model when there were sufficient standby virtual routers for each virtual network and the substrate nodes have the capability to support multiple standby virtual routers to be in service, concurrently.**

I. Introduction

Network virtualization allows the capability for dynamic virtual network (VN) reconfiguration, where virtual routers (VRs) for VNs are easy to create, configure, and connect to the virtual networks. In traditional (physical) computer networks, adding or removing network components, such as links or nodes, is time consuming. In a virtual network environment, configuration occurs at the logical level using a software-based approach. For example, VN providers can create multiple VRs from different substrate nodes or the same substrate node; for each VR, virtual interfaces are configured through a software-defined approach, compared to physical routers.

As a result of such capabilities of network virtualization, multiple VNs are easy to be created over a substrate network. In such an environment, we consider the problem of restoring VNs from a node failure. A substrate network node failure can potentially impact multiple VN nodes belonging to different VNs. For any failure, the service provider of the VNs may be required to restore VNs as specified in the service level agreements (SLAs). We refer to a VN service provider as a VN customer, while its VN being provisioned by the substrate network provider.

In this work, we focus on optimally restoring VNs back to their original topological structures and bandwidth requirements from a node failure where the failed node could be either at the substrate network or in one or more VNs. Such requirements may be imposed by the VN customers who want virtual topologies leased from the substrate network provider to be the same at any time as part of the service level agreements (SLAs).

[1] *Supported by NSF Grant Nos. 0916505 and 1217736.*

978-3-901882-66-1 © 2014 IFIP

An advantage of VN-over-substrate network architecture is that the substrate network provider can provision standby virtual routers (S-VRs) spread across the VNs so that they can be quickly activated through a software-defined mechanism in case of any node failure by restoring back to the original VN topologies. In this case, the VN customer do not need to make changes to the routing tables and can minimize transient issues. For example, consider a VN that has deployed the OSPF (open shortest path first) routing protocol. OSPF can recognize a single link failure well; however, OSPF does not support a node failure function well as it has to learn from correlating the non-functioning of links associated with the failed node to eventually recognize the node failure—this may result in an extended transient window when traffic flow may suffer for the VN customers. Thus, having S-VRs preconfigured can allow quick restoration from a VR failure for each VN. Secondly, if the substrate network provider is using software-defined networks (SDN), such functions can be orchestrated through a centralized controller. We are also partly motivated to address this standby router deployment from our own experience with GpENI (www.gpeni.net), an inter-continental research network testbed spanning the US and Europe. In this testbed, software-based functionalities were successfully developed to deploy S-VRs to restore a node failure in supporting a VN with encouraging results [1].

Consider Fig. 1, when a core virtual network is created, a set of S-VRs are reserved. These S-VRs are functionally identical to active virtual routers (VRs) but not connected to any active VRs. When there is a VR failure in the virtual network, the dynamic reconfiguration process is triggered, and

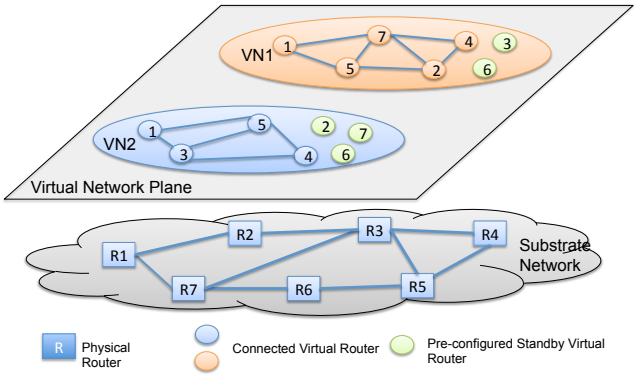


Fig. 1: Dynamic Reconfiguration Scheme for Virtual Networks

it selects one of the S-VRs and replaces the failed VR. Because both traffic load and hardware capacity are changing at the substrate nodes where these S-VRs reside, selecting an S-VR that has minimum impact to the current virtual network while minimizing operations cost is important during the dynamic reconfiguration process.

In this paper, we address the problem of selecting optimal S-VRs when nodes in the core virtual networks are affected due to a failure. End hosts may be connected to multiple edge nodes, and the loss to end users is not the focus of this work. The scope is to recover from a failure that occurs at core VRs, *not* edge VRs of the VNs. We present an optimization model and a heuristic that can be used by the virtual network management system to dynamically reconfigure and restore VNs. We then introduce numerical studies to present the dependency of optimal selection using a number of parameters, and the results show that the proposed heuristic performs closely to the optimization model when the reserved S-VR set for each VN was relatively large, and the number of failures was within a particular range.

The rest of the paper is organized as follows: In Section II, we present the optimization formulation to the optimal standby virtual router selection problem along with a heuristic. In Section III, we present studies on three substrate network topologies by discussing two types of virtual router failures; our optimization model is efficient in selecting optimal S-VRs and we present a comparison between the heuristic and optimal selection for a number of scenarios. Section IV presents related works on dynamic reallocation of resources in virtual networks. Section V summarizes the paper and presents plans for future work.

II. Problem Formulation

The scope of this work is to optimally determine standby virtual routers (S-VRs) that are to be activated to recover from VR failures in the core VNs. A VR failure in a VN may be caused due to the VR's software failure or a failure of its hosting node in the substrate network. Consider now a number of virtual networks (denoted by $\mathcal{G}$) operating over the common substrate network. We are given the requirement that each core VN is to be reconfigured back to the same topology as it was before the failure. A set of S-VRs is available to support the VNs for dynamic reconfiguration in the event of a VR failure. We assume the following in this work:

- Within each VN, any two VRs, including the S-VRs, are not hosted on the same substrate node.
- One substrate node fails at a time.
- For each VN, at most one core VR fails at a time.

The first assumption reflects that when creating a virtual network (especially a wide-area virtual network), it is not common to create a virtual link connecting two VRs for a VN that is hosted by the same substrate node. Secondly, if two VRs for the same VN were hosted on a substrate node, where one is activated while the other is reserved as an S-VR, then in the case of a failure of the working VR, it can be automatic switched to the S-VR by the hosting substrate node, while preserving the VN topology or bandwidth. That is, we focus on the non-trivial problem where the S-VR resides at a different substrate node from the failed VR. In regard to the second assumption, it does not rule out the possibility that a substrate node failure may cause the single-VR failures in multiple virtual networks, where these failed VRs are hosted by the failed substrate node.

TABLE I: Model Entities

Symbols	Description
$\mathcal{R}$	Set of substrate nodes
$\mathcal{L}$	Set of substrate links
$\mathcal{Q}_{ik}$	Set of paths from the substrate node i to the substrate node k
$\mathcal{P}_i$	Set of used physical interfaces on substrate node i
$\mathcal{G}$	Set of virtual networks
$\mathcal{V}^j$	Set of connected VRs in VN j
$\mathcal{S}^j$	Set of S-VRs available to VN j
$\mathcal{F}^j$	Set of failed VRs in virtual network j,
n^j	Number of failed VRs in virtual network j, $n^j = \lvert\mathcal{F}^j\rvert$
$b_{i,p}(b_l)$	Residual capacity of used interface p on a substrate node i, or the substrate link $l \in \mathcal{L}$, where the source of l is node i's interface p
T	Threshold for maximum residual bandwidth utilization
r_i^j	A VR on a substrate node i in a virtual network j
$\mathcal{M}_i^j$	The set of virtual interfaces on a VR r_i^j
δ_i^j	1 if r_i^j is reserved as an S-VR, 0 otherwise
α_i^j	1 if r_i^j is connected in VN j, 0 otherwise
$e_{i,k}^j$	1 if a virtual link between r_i^j and r_k^j is up, 0 otherwise
$\gamma_{i,m,k}^{j,f}$	1 if a virtual interface m on S-VR r_i^j is configured to connect r_k^j (a neighbor to the failed VR r_i^j)
β_i^j	1 if a VR r_i^j fails, 0 otherwise
h_i	The maximum number of S-VRs can be selected from a substrate node i
$c_{i,m,k}^{j,f}$	The bandwidth required on interface m of a candidate S-VR r_i^j that is going to replace the failed VR r_f^j to connect the VR r_k^j
$\eta_{i,m,k}^{j,f}$	The operation cost on configuring a virtual interface m of a candidate S-VR r_i^j to connect to a existing VR r_k^j, by replacing the failed VR r_f^j
$\sigma_{i,m,k}^{j,f}$	The cost of connecting a candidate S-VR to a existing VR r_k^j (a neighbor to the failed VR r_i^j) in a virtual network j via the S-VR's virtual interface m
$\Delta_{q,l}^{(jfik)}$	The link-path indicator. For a capacity demand from r_i^j to r_k^j given the failure of the virtual router r_f^j, a substrate path $q \in \mathcal{Q}_{ik}$ from node i to node k uses the substrate link l

TABLE II: Variables

Variable	Description
$u_i^{j,f}$	Binary variable; it equals to 1 if an S-VR r_i^j is selected to replace a failed VR r_f^j
$v_{i,m,k}^{j,f}$	Binary variable; it equals to 1 if an S-VR r_i^j's is connected to the failed VR r_f^j's neighbor r_k^j via virtual interface m
t	Maximum residual bandwidth utilization on a substrate node when determine the S-VR r_i^j for each affected VN j, $t \in (0, T)$, T is the threshold for t
$x_q^{(jfik)}$	Flow variable on a substrate path $q \in \mathcal{Q}_{ik}$. Under a failure of r_f^j ($f \in \mathcal{F}^j$), the path is between a candidate S-VR v_i^j and the failed r_f^j's neighbor r_k^j on the substrate network

A. Notations

We first briefly highlight a few key notations used in the optimization model; all are listed in Tables I, II, and III.

TABLE III: Weight Parameters

Symbols	Description (all non-negative)
λ, π	Indicates the proportion for sub-objectives
w_θ	Primary weight parameters used in (11): $w_\theta = \langle w_{\theta 1}, w_{\theta 2} \rangle$
w_b	Secondary weights in (13): $w_b = \langle w_{b1}, w_{b2} \rangle$
w_a	Third level weights in (13): $w_a = \langle w_{a1}, w_{a2} \rangle$

There are two main sets of components in the virtualized network environment: substrate nodes ($\mathcal{R}$) and virtual networks ($\mathcal{G}$). A VR that belongs to virtual network j and is hosted by substrate node i is denoted by r_i^j. For any substrate node i ($i \in \mathcal{R}$), its physical interfaces are denoted by the set $\mathcal{P}_i$, where each physical interface p ($p \in \mathcal{P}_i$) has residual capacity $b_{i,p}$. For any VR r_i^j, we define its virtual interfaces as set $\mathcal{M}_i^j$. For each virtual network j ($j \in \mathcal{G}$), we denote $\mathcal{F}^j$ to be the set of failed VRs in the virtual network j.

For any virtual network j ($j \in \mathcal{G}$) that has already been created and is in service, the substrate nodes hosting VRs that have been connected are pre-established, which is denoted by set $\mathcal{V}^j (\subset \mathcal{R})$; thus, the following parameters are set: $\alpha_i^j = 1$, $e_{i,k}^j = 1$, $\beta_i^j = 0$ for $i, k \in \mathcal{V}^j$. We also denote the substrate nodes hosting the S-VRs pre-reserved for a virtual network j as $\mathcal{S}^j (\subset \mathcal{R})$, so $\delta_i^j = 1$ for S-VRs, $i \in \mathcal{S}^j$. In other words, $\delta_i^j = 0$ for $i \in \mathcal{R} \backslash \mathcal{S}^j$. When a VR r_i^j fails, we set $\beta_i^j = 1$, where $i \in \mathcal{V}^j \cup \mathcal{S}^j (\subseteq \mathcal{R})$. Parameter h_i indicates the maximum number of S-VRs associated with a substrate node that are candidates for selection during the reconfiguration process, and the value of h_i can be decided based on the real-time resource utilization on the substrate nodes.

The solution to the reconfiguration problem is determined by two sets of binary variables (shown in Table II) that determine the optimal selection of an S-VR. Variable t represents the maximum residual bandwidth utilization of the substrate nodes after the S-VRs have been decided, while x is for the flow from the S-VRs to the failed VR's neighbors.

B. Constraints

There are ten sets of constraints.

- A VR r_i^j can be selected to replace a failed VR r_f^j only if it is identified as an S-VR and it does not fail.

$$u_i^{j,f} \leq \delta_i^j (1 - \beta_i^j), \\ \forall i \in \mathcal{R}, \forall f \in \mathcal{F}^j, \forall j \in \mathcal{G}, i \neq f \quad (1)$$

- For any VR failure in a VN, only one identified S-VR can be selected.

$$\sum_{i \in \mathcal{R}} \delta_i^j \cdot u_i^{j,f} = 1, \forall f \in \mathcal{F}^j, \forall j \in \mathcal{G} \quad (2)$$

- One reserved S-VR can replace at most one failed VR at a time:

$$\sum_{f \in \mathcal{F}^j} \delta_i^j \cdot u_i^{j,f} \leq 1, \forall i \in \mathcal{R}, \forall j \in \mathcal{G} \quad (3)$$

- The allocation must satisfy the upper bound on the maximum number of S-VRs that can be selected from the same substrate node at a time, based on the substrate node resource utilization at the moment (e.g., CPU or bandwidth utilization).

$$\sum_{f \in \mathcal{F}^j} \sum_{j \in \mathcal{G}} \delta_i^j u_i^{j,f} \leq h_i, \forall i \in \mathcal{R} \quad (4)$$

- A virtual interface m on an S-VR r_i^j can be enabled to connect r_k^j (a neighbor VR of the failed VR r_f^j), only when r_i^j is selected to replace r_f^j.

$$\gamma_{i,m,k}^{j,f} \cdot v_{i,m,k}^{j,f} \leq u_i^{j,f}, \\ \forall f \in \mathcal{F}^j, \forall i, k \in \mathcal{R}, \forall j \in \mathcal{G}, \forall m \in \mathcal{M}_i^j, i \neq k \neq f \quad (5)$$

- For topology integrity from before failure to after failure, conservation of links during the reconfiguration process must be satisfied.

$$\sum_{m \in \mathcal{M}_i^j} \sum_{k \in \mathcal{R}} e_{f,k}^j \cdot \delta_i^j \cdot \gamma_{i,m,k}^{j,f} \cdot v_{i,m,k}^{j,f} = \sum_{k \in \mathcal{R}} \delta_i^j \cdot u_i^{j,f} \cdot \beta_f^j \cdot e_{f,k}^j \\ \forall j \in \mathcal{G}, \forall f \in \mathcal{F}^j, \forall i \in \mathcal{R}, i \neq k \neq f \quad (6)$$

  Here, the left side of (6) indicates that a candidate S-VR (r_i^j) for a failed VR (r_f^j) should be able to establish connectivities to every failed VR's neighbor (r_k^j, where $i \neq k$), via specific virtual interface m ($m \in \mathcal{M}_i^j$)—this must be equal to the right side of (6) that represents the total number of virtual links that should be created from r_i^j in order to replace r_f^j.
- Regarding the residual capacity of the interfaces on a candidate S-VR r_i^j that is to be selected to replace the failed VR r_f^j, the accumulated bandwidth demanded from all connected virtual interfaces should not exceed the residual total interface residual capacity on the corresponding substrate node:

$$\sum_{m \in \mathcal{M}_i^j} \sum_{k \in \mathcal{R}} \delta_i^j \cdot \gamma_{i,m,k}^{j,f} \cdot c_{i,m,k}^{j,f} \cdot v_{i,m,k}^{j,f} \leq \sum_{p \in \mathcal{P}_i} \delta_i^j \cdot b_{i,p} \cdot u_i^{j,f}, \\ \forall j \in \mathcal{G}, \forall f \in \mathcal{F}^j, \forall i \in \mathcal{R}, i \neq k \neq f, \quad (7)$$

- The residual capacity utilization on a substrate node does not exceed the maximum residual bandwidth utilization t, due to the extra bandwidth demand by the one or more selected S-VRs hosted by this substrate node.

$$\sum_{j \in \mathcal{G}} \sum_{f \in \mathcal{F}} \delta_i^j \cdot u_i^{j,f} \cdot (\sum_{k \in \mathcal{R}} \sum_{m \in \mathcal{M}_i^j} \gamma_{i,m,k}^{j,f} \cdot c_{i,m,k}^{j,f}) \leq \sum_{p \in \mathcal{P}_i} b_{i,p} \cdot t, \\ \forall i \in \mathcal{R}, i \neq k \neq f \quad (8)$$

- When selecting an S-VR r_i^j, we need to guarantee that the links on the path from the substrate node i hosting r_i^j to the substrate node k hosting the failed VR r_f^j's neighbor r_k^j has sufficient residual bandwidth on the substrate network, and that the bandwidth demand on the new virtual link are the same as before. This can be represented through the following demand flow and capacity constraints:

$$\sum_{q\in\mathcal{Q}_{(ik)}} x_q^{(jfik)} = \sum_{m\in\mathcal{M}_i^j} \gamma_{i,m,k}^{j,f}\cdot c_{i,m,k}^{j,f}\cdot \delta_i^j \cdot u_i^{j,f},$$
$$\forall j\in\mathcal{G}, \forall f\in\mathcal{F}^j, \forall i,k\in\mathcal{R}, i\neq k\neq f \quad (9)$$

$$\sum_{(j,f,i,k)}\sum_{q\in\mathcal{Q}_{ik}} \Delta_{q,l}^{(jfik)}\cdot x_q^{(jfik)} \leq b_l,$$
$$\forall l\in\mathcal{L}, i\neq k\neq f \quad (10)$$

C. Objective Function

The goal (11) is to minimize a composite objective that represents the cost of selecting S-VRs for each VN under an event of VR failure (so-called minimum cost selecting problem) and the maximum resource utilization (t) of substrate nodes (so-called resource balancing problem).

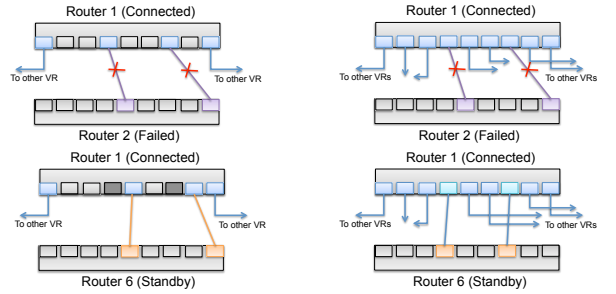
There are two types of cost functions for S-VR selection and connection. The first type is the cost of operations on virtual interfaces (Type-I: $\eta_{i,m,k}^{j,f}$), for instance, configuring IP addresses, and activating or deactivating virtual interfaces. The second type is the cost of connecting two VRs (Type-II: $\sigma_{i,m,k}^{j,f}$), for instance, the geographical distance or the RTT between the two VRs.

For the resource balancing problem, we define the total capacity of all interfaces on i (i.e., $\sum_{p\in\mathcal{P}_i} b_{i,p}$) as the residual resources on substrate node $i\in\mathcal{R}$, and we want the total interface capacity utilization on each substrate node not to be over a particular threshold t, after this substrate node actively hosts the S-VRs in the VNs.

To select a proper S-VR to replace a failed VR for a VN, it is necessary to balance the cost of activation and connectivity. For more than one VN under single-VR failures, we consider the resource balancing problem on the substrate nodes for which we assign a weight parameter ($w_\theta = \langle w_{\theta 1},\ w_{\theta 2}\rangle$) to indicate the importance of the relevant cost term, where $w_{\theta 1}$ is for $\eta_{i,m,k}^{i,j}$, and $w_{\theta 2}$ is for $\sigma_{i,m,k}^{j,f}$. To weigh the minimum cost selection and resource balancing, we use weight parameters λ and π, respectively, where $\lambda + \pi = 1$.

$$\min\{\lambda\cdot[\sum_{j\in\mathcal{G}}\sum_{f\in\mathcal{F}^j}\sum_{i\in\mathcal{R}}\sum_{m\in\mathcal{M}_i^j}\sum_{k\in\mathcal{R}}(w_{\theta 1}\cdot\eta_{i,m,k}^{j,f} + w_{\theta 2}\cdot\sigma_{i,m,k}^{j,f})\cdot$$
$$\delta_{i,j}\cdot\gamma_{i,m,k}^{j,f}\cdot v_{i,m,k}^{j,f}] + \pi\cdot t\}, i\neq k\neq f \quad (11)$$

1) Type-I: Virtual Interface Operations Cost $\eta_{i,m,k}^{j,f}$: In the substrate network, when adding or removing a substrate router from the network, the basic operations are to enable or disable the interfaces if the router has already been connected. If the router is not connected yet, then plugging cables is an extra operation. Furthermore, if the interface is not created or is not assigned any IP address, then configuring IP addresses is also an extra operations cost. Analogously, the operations cost on the virtual interfaces also needs to consider these three parts: virtual IP address configuration, creation of virtual link on a virtual interface (similar to a plugin cable to a substrate router), and enabling/disabling the virtual interfaces. Here, we combine the first two into one operations cost. This is because creating virtual links and assigning IP addresses can be done by a programmable operation. For example, in [2], when users specified a topology, the virtual link between two VRs was created automatically, and the corresponding virtual interfaces were assigned virtual IP addresses at the same time. The other operations cost in a virtualized networking environment is due to enabling necessary virtual interfaces on the virtual interfaces. It is intuitive that not all virtual interfaces need to be enabled, and the number of active virtual interfaces on an S-VR depends on the number of VRs that it needs to be connected to. Meanwhile, the interfaces of other VRs that connect to the failed VR have to be disabled as well.



(a) With Extra Virtual Interfaces (b) No Free Virtual Interfaces

Fig. 2: Two Scenarios on Virtual Interface Operations

To construct the Type-I cost for the virtual interface operation, there are two scenarios. First, as presented in Fig. 2(a), each VR has enough virtual interfaces with preconfigured virtual IP addresses, and all of the S-VRs' virtual interfaces are disabled. When an S-VR is selected, we just need to enable and disable its corresponding virtual interfaces. In other words, the first scenario considers only one part of the Type-I cost. Secondly, as shown in Fig. 2(b), a connected VR has a limited number of virtual interfaces and all of them are active, so if it needs to connect to the selected S-VR, the specific virtual interface must be re-initialized with a new IP address to connect the S-VR. Thus, this scenario needs to consider both parts of the Type-I cost. Note that s^- and s^+ represent the cost of disabling a virtual interface on a remote VR r_i^j and enabling the virtual interface m on the standby r_i^j, respectively. $\tau_i^{j,f}$ is a binary value that indicates whether re-initialization of the virtual IP addresses for virtual interfaces is needed, and l is the cost of configuring virtual IP addresses. To summarize, the Type-I cost can be presented as follows:

$$\eta_{i,m,k}^{j,f} = 2\cdot(s^- + s^+) + \tau_i^{j,f}\cdot l$$
$$\forall m\in\mathcal{M}_i^j, \forall k\in\mathcal{R}, i\neq k\neq f \quad (12)$$

2) Type-II: Virtual Router Connectivity Cost $\sigma_{i,m,k}^{j,f}$: The Type-II cost is to connect two VRs. We consider two factors: geographical distance and round trip time (RTT). The geographical distance is the the distance between two substrate nodes that host the VRs. Here the geographical distance is either between an S-VR and a failed VR, or the distance between an S-VR and the remote VRs to be connected. Note that the RTT cannot be measured directly before the S-VR

is connected. The virtual network control central can create a separate VN as a reference that is a full mesh topology and each VR in the reference virtual network is created from a different substrate router. Thus, we can collect RTT in real time and the packet loss rate information for the cost function. Hence, the virtual router connectivity cost can be constructed as (13), where a and b are weight parameters.

$$\sigma^{j,f}_{i,m,k} = w_{b1} \cdot (w_{a1} \cdot d_{i,f} + w_{a2} \cdot d_{i,k}) + w_{b2} \cdot rtt_{i,k}, \quad \forall j \in \mathcal{G}, \forall m \in \mathcal{M}^j_i, i \neq k \neq f \tag{13}$$

D. Heuristic Algorithm

Taking a cue from the optimization model, we now propose a heuristic algorithm that selects S-VRs to restore the VR failures, which is shown in Algorithm 1. In (11), two types of costs ($\eta^{j,f}_{i,m,k}$ and $\sigma^{j,f}_{i,m,k}$) are considered in the minimum cost selection objective, so the proposed Algorithm 1 implements a multi-criteria S-VR selection for all virtual networks with VR failures.

Algorithm 1 takes the following as the input.

- A hash map $failed\_map$ that stores each VN's failed VR, along with its bandwidth usage, and the keys are the identities of the VNs;
- Geographical information (geo) on substrate nodes;
- The most recent network dynamics monitored in real-time for both substrate and VNs before the failures, such as bandwidth (bw) on substrate nodes and the round trip time (rtt) between the VRs;
- Parameter $limit(= h_i)$ is to provide a boundary for the number of S-VRs that can be selected from a common substrate node.
- Weights, w_θ, w_a, w_b, on each factor (refer to (13) and (11)) to understand the influence on the selection and the balancing weights λ and π.
- A threshold T for the maximum residual bandwidth utilization of substrate nodes to guarantee that the substrate nodes are not congested.

The heuristic algorithm first sorts the VNs based on the bandwidth used by the failed VR at the failure, and starts to select the S-VR from the VN whose failed VR has the highest bandwidth utilization. It then, sequentially, selects S-VRs for the rest of the VNs. For each VN with the failure, the algorithm first calculates the selecting cost for each S-VR candidate. The calculation is very similar to the objective function, so it also considers two parts. The first part is a summation of Type-I and Type-II costs; the second part accounts for the residual bandwidth utilization on the corresponding substrate node if this S-VR candidate is to be selected. Weights λ and π are associated with each part, respectively.

Once the algorithm completes the computation for each candidate S-VR, these candidate S-VRs are sorted by the computed costs. Then the algorithm starts from the S-VR with the least selection cost, and checks two conditions: $limit$ and T. Since the objective (11) is a composite of two sub-optimization problems, the heuristic needs to consider the residual bandwidth balancing during the selection, based on the ratio of $\frac{\pi}{\lambda}$ ($\frac{\pi}{\lambda} = \infty$ for $\lambda = 0$). If $\frac{\pi}{\lambda}$ is greater than a particular level, i.e., 10, for each affected VN, the heuristic tends to select the S-VR hosted by different substrate node to balance the residual bandwidth utilization, otherwise the heuristic considers the purely minimum cost selection objective.

Algorithm 1: Heuristic Multi-Criteria S-VR Selection

Input: $failed\_map$, geo, bw, rtt, $limit$, T, w_a, w_b, w_θ, λ, π
Output: $selected$

```
Initialize selected as empty
/* Sort the virtual networks by the bandwidth used by the
   fvr_id at the failure, and store it as sorted_vnets */
sorted_vnets ← Sort_VN (failed_map)
for vn in sorted_vnets do
    vn_id, bw_req ← Get_Info(vn)
    /* The hash table Map stores the cost of selecting a
       S-VR in a vn, in form of ⟨S-VR, cost⟩ */
    Initialize Map as empty
    fvr_id ← Get_FailedVR(failed_dict[vn_id])
    if vn has failure fvr_id then
        Get the list of S-VRs in vn: svr_id
        for svr_id in standby_list do
            Compute the total cost of selecting svr_id:
            total, using (11), (12) and (13)
            Map.Add (svr_id, total)
        //Sort the Map by the cost
        sorted_map ← Sort_SVR (Map)
        for svr_id in sorted_map do
            Compute utilization t if svr_id is selected
            Allocate substrate path with sufficient
            capacity for the virtual link between svr_id
            and nbr_id
            if Count (selected, svr_id) < limit and
            t < T then
                if π/λ ≥ 10 then
                    //Do residual bandwidth balancing
                    If a substrate node hosting svr_id is
                    least used, then select this svr_id
                else
                    //Purely minimum cost selection
                    Select the svr_id
                Update the residual capacity on the
                substrate links
return selected
```

III. Results

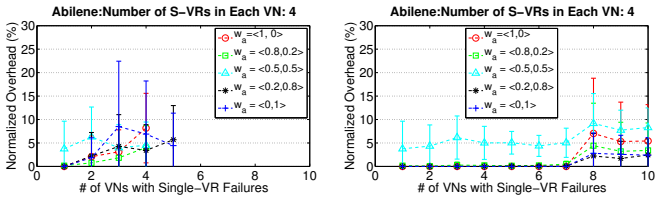
In this section, we present numerical studies to show how the optimal selection is affected under different scenarios, and compare the performance of the heuristic and optimal selection. Before discussing the results, we first present an overview on the experimental scenarios.

TABLE IV: Scenarios with 10 VNs

Substrate Network	*Abilene*	*Nobel-EU*	*Germany50*
Nodes / Links	12 / 15	28 / 41	50 / 88
Residual BW	$0 \sim 1$	$0 \sim 1$	$0 \sim 1$
# of reserved S-VRs	$2 \sim 6$	$6 \sim 10$	$8 \sim 24$
# of Independent VR Failures	$1 \sim 10$	$1 \sim 10$	$1 \sim 10$
# of Dependent VR Failures	N/A	N/A	10
h_i	$1 \sim 10$	$1 \sim 10$	$1 \sim 10$

TABLE V: Weight Parameters

Dominant Criteria	Primary		Secondary		Third	
	$w_{\theta 1}$	$w_{\theta 2}$	w_{b1}	w_{b2}	w_{a1}	w_{a2}
Type-II (Geolocation)	0	1	1	0	0	1
	0	1	1	0	0.2	0.8
	0	1	1	0	0.5	0.5
	0	1	1	0	0.8	0.2
	0	1	1	0	1	0



(a) Abilene: $h_i = 1$ (b) Abilene: $h_i = 3$

Fig. 3: Impact of w_a (Abilene Network): 4 S-VRs per VN

A. Scenarios

We studied three topologies obtained from the `SNDlib` (Survivable Network Design Library) [3]; they are *Abilene* topology, *Nobel-EU* topology, and *Germany50* topology. The geographical distance between nodes was normalized for each topology, and the residual capacity on each active physical interface of a substrate node was normalized as well.

Consider the origin of a VR failure, which could be either a software failure at the VR itself, or a failure at its substrate host. For a software failure at the VR, it will not affect other VNs, so we denote this type of VR failure as an *independent* VR failure. For a VR failure caused by a substrate node failure, other VNs may also be affected if these VNs contain the VRs hosted by the failed substrate node—this is referred to as a *dependent* VR failure.

There are several factors that may affect the total cost of selecting S-VRs for all VNs with single VR failures: the number of S-VRs provided for each VN, the number of VR failures for different VNs, and the restriction on the amount of S-VRs (h_i) can be selected from a common substrate node. Thus, we created a set of scenarios (see Table IV) to study the impact of these factors on the optimal selection for each topology. Note that the number of VR failures can be equal up to the total number of VNs. This is possible when there is a substrate node failure that affects every virtual network. We initially provisioned 10 VNs randomly (before any failure) on the substrate network; each VN was independent from the others in terms of failures and S-VR availability.

Recall that the objective (11) consists of two cost types weighted by two primary weight parameters. In this study, we assumed each VR had enough virtual interfaces, and the virtual interfaces were homogeneous, so $\tau_i^{j,f} = 0$, and $\eta_{j,m,k}^{j,f}$ equaled to a constant η. Hence, Type-II was the major cost, and we considered the geolocation as the main factor; in other words, $w_{b1} = 1$ and $w_{b2} = 0$.

Table V presents groups of values that we used for each set of weight parameters to study the impact of the optimal S-VR selection under various weight combinations. Details on the impact of each weight combination will be discussed in the next subsection. In our study, $T = 0.8$ for all scenarios.

In the rest of this section, we first present a study on the impact of w_a in (13), and then discuss the S-VR selection with respect to the independent VR failures and dependent VR failures, successively. Since the VNs were randomly created initially, and the failed VR and S-VRs within each VN were randomly determined, we repeated 10 independent runs for each case, and computed the 95% confidence interval (95% CI) for the normalized overhead of the heuristic over the optimal solution from the optimization model. The optimization model was solved to optimality using CPLEX (version 12.5.1.0)—this solution will be referred to as the optimal solution. The heuristic algorithm was implemented in Python 2.7.

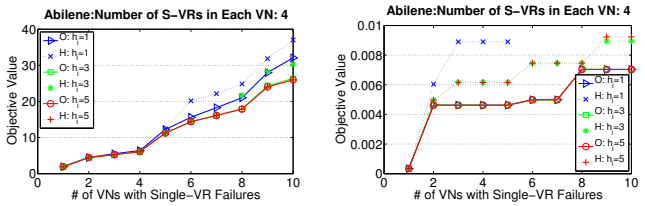
B. Impact of w_a

To study the impact of w_a, we set $\pi = 0$. Since we already assumed $\eta_{i,m,k}^{j,f} = \eta$, and $w_b = \langle 1, 0 \rangle$, the geolocation was the main factor to Type-II cost. Intuitively, when a VR failure occurs, we may select an S-VR that is closest to the failed VR (w_a = $\langle$1,0$\rangle$); or select an S-VR that has a minimum average distance to the failed VR's neighbors in the VN (w_a = $\langle$0,1$\rangle$). The rest of the sets of w_a considered a joint impact of the two intuitive cases, as shown in Table V.

We used the 12-node, 15-link Abilene substrate network where 10 VNs were provisioned to illustrate the impact of w_a, and assigned $h_i = 1$ or 3 (as shown in Fig. 3).

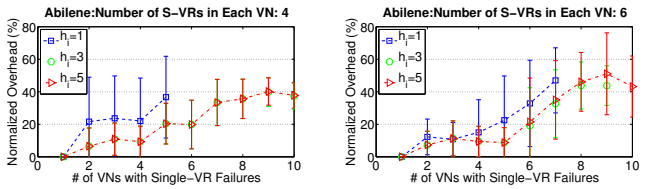
For $h_i = 1$, the heuristic may not have a feasible selection except for a small number of failures (e.g., five failures as shown in Fig. 3(a)). In general, the normalized overhead of the heuristic over the optimal selection increased as the number of failures increased. By looking at the 95% CI of the normalized overhead for each w_a, under a feasible solution, the heuristic performed as well as the optimal selection for four failures or less. Thus, when the substrate node has a restriction for actively provisioning multiple S-VRs for VR failure recovery, the heuristic may have a high overhead. On the other hand, in this scenario, as we increased the portion of w_{a2} (e.g., $w_{a2} > 0.5$), the heuristic had more feasible solutions than the cases with $w_{a2} \leq 0.5$.

When a substrate node was able to support multiple S-VRs (i.e., $h_i = 3$), the heuristic has a feasible selection for up to 10 failures (Fig. 3(b)). In general, the normalized overhead is less than about 15% for all five combinations of w_a that we have tested. Although the normalized overhead increased as the number of failures raised, for $w_a = \langle 0, 1 \rangle$ and $w_a = \langle 0.2, 0.8 \rangle$, the normalized overhead was less than 3%. We also



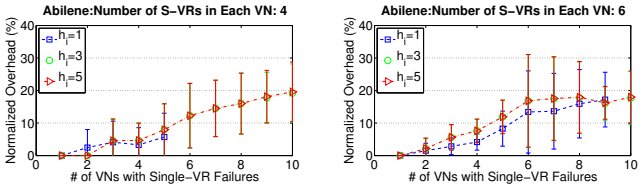
(a) $\lambda = 1, \pi = 0$ (b) $\lambda = 0, \pi = 1$

Fig. 4: *Abilene*: Independent VR Failures (Objective Value)



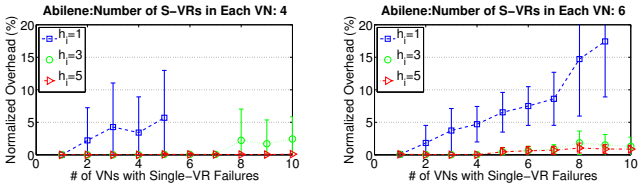
(a) Four S-VRs per VN (b) Six S-VRs per VN

Fig. 5: *Abilene*: Independent VR Failures ($\lambda = 0$, $\pi = 1$)



(a) Four S-VRs per VN (b) Six S-VRs per VN

Fig. 6: *Abilene*: Independent VR Failures ($\lambda = 0.02$, $\pi = 0.98$)



(a) Four S-VRs per VN (b) Six S-VRs per VN

Fig. 7: *Abilene*: Independent VR Failures ($\lambda = 1$, $\pi = 0$)

observed that for $w_a = \langle 0.5, 0.5 \rangle$, the normalized overhead was higher than the rest of the four combinations of w_a, so it is better to consider one type of connectivity as the primary cost, which is the distance to either the failed VR or to its neighbors.

From the discussion above, $w_a = \langle 0.2, 0.8 \rangle$ and $w_a = \langle 0, 1 \rangle$ were found to be two proper weights for use in (13), as the normalized overhead is more consistent when the number of failures increased, and the swing of the 95% CI was small. In the rest of our discussion, we used $w_a = \langle 0.2, 0.8 \rangle$, along with the variations of λ and π.

In the above and in the rest of the paper, we present our results as the normalized cost overhead comparing the heuristic and the optimal solution. In Fig. 4(a) and Fig. 4(b) corresponding to $\lambda = 1$ and $\lambda = 0$, the actual objective function values are plotted. As anticipated, the optimal cost increases significantly as the number of failures increases, while it is easy to see that the cost difference is more pronounced between the heuristic and the optimal solutions when $\lambda = 0$, i.e., when the goal is purely load balancing on the residual capacity at substrate nodes.

C. Independent Virtual Router Failures

An independent VR failure defines a VR failure such as the one caused by a software failure within the VR; it is assumed that this failure in a VN does not affect other VNs. In this section, we present normalized overhead of a heuristic over optimal S-VR selection for *Abilene*, *Nobel-EU*, and *Germany50* topologies.

1) Abilene Topology: If $\lambda = 0$ and $\pi = 1$, the goal is to purely balance residual bandwidth. When $h_i = 1$, the heuristic did not result in a feasible solution when there were many failures (six in Fig. 5(a) and eight in Fig. 5(b)). Thus, under a limited resource provisioning capability on the substrate network, reserving more S-VRs for each VN is good for the heuristic to find a feasible solution. However, only increasing the number of reserved S-VRs for each VN is not sufficient. For $h_i = 1$, the normalized overhead of a heuristic over optimal was still high with a large number of failures, and the width of a 95% confidence interval was high as well. Thus, increasing h_i can help to reduce the normalized overhead. In Fig. 5(a), if we increased h_i from 1 to 3 at five failures, the 95% CI was reduced from $(36.67 \pm 25.15)\%$ to $(20.40 \pm 12.54)\%$.

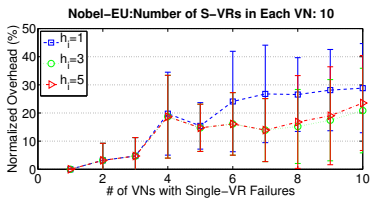
For $\lambda = 0.02$ and $\pi = 0.98$, the objective is a composite of minimum cost selection and residual bandwidth balancing, but the latter is still the dominant objective. Compared with Fig. 5(a), at 10 failures, Fig. 6(a) shows that when we slightly increased λ from 0, given $h_i \geq 3$, the normalized overhead's CI was reduced from $(37.75 \pm 7.83)\%$ to $(19.58 \pm 9.16)\%$. Fig. 6(b) presents a similar behavior.

For $\lambda = 1$ and $\pi = 0$ (see Fig. 7), the goal is purely a minimum cost selection. The normalized overhead was high when $h_i = 1$, and it was close to 0% when $h_i \geq 3$.

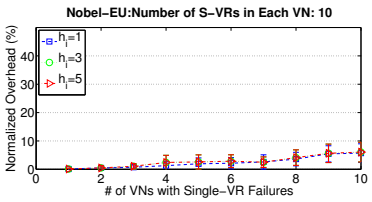
2) Nobel-EU Topology: For *Nobel-EU* topology, we varied the number of S-VRs for each scenario from 6 to 14, and we present the result for 10 S-VRs per VN, which not only allowed customers to request VNs with a maximum 18 VRs, but also provided a proper amount of candidate S-VRs for each VN. This way, based on our discussion on the feasibility of a heuristic algorithm, we can always find both a feasible heuristic and an optimal S-VR selection.

Fig. 8 presents the 95% CI of normalized overhead when $\langle \lambda, \pi \rangle$ was $\langle 0, 1 \rangle$, $\langle 0.2, 0.98 \rangle$, and $\langle 1, 0 \rangle$, respectively. We observed that when the proper size of the candidate S-VR set was provided, h_i was no longer a critical constraint reflected by the overlapping of the 95% CIs for a different h_i. For $\pi = 1$, the optimal selection aimed to balance the residual bandwidth utilization over the substrate nodes, and both heuristic and optimal selection tended to spread out the selected S-VR to different geographical locations across the whole substrate network.

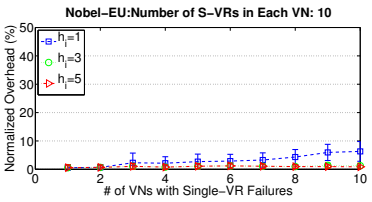
When we increased λ slightly (see Fig. 8(b)), the minimum



(a) $\lambda = 0$, $\pi = 1$

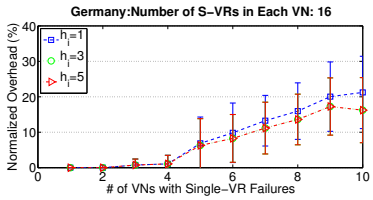


(b) $\lambda = 0.02$, $\pi = 0.98$, w_a = $\langle 0.2{,}0.8 \rangle$

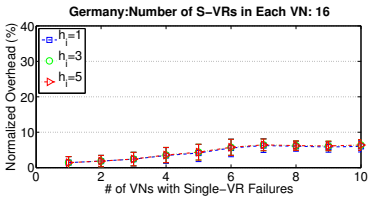


(c) $\lambda = 1$, $\pi = 0$,w_a = $\langle 0.2{,}0.8 \rangle$

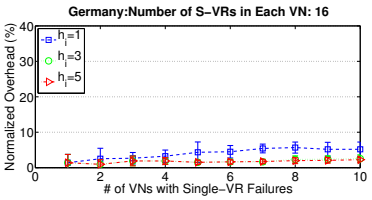
Fig. 8: Independent Failures in *Nobel-EU* Topology



(a) $\lambda = 0$, $\pi = 1$



(b) $\lambda = 0.02$, $\pi = 0.98$, w_a = $\langle 0.2{,}0.8 \rangle$



(c) $\lambda = 1$, $\pi = 0$, w_a = $\langle 0.2{,}0.8 \rangle$

Fig. 9: Independent Failures in *Germany50* Topology

cost selection objective helped reduce the normalized overhead at a constant level for more than three failures. For $\lambda = 1$, higher value of h_i still showed advantages (see Fig. 8(c)) as the selected S-VRs from multiple VNs with a VR failure can be provisioned from a common substrate node that minimized Type-II cost.

3) Germany50 Topology: For *Germany50* topology, we varied the number of S-VRs per VN from 12 to 24. We report the 95% CI of the normalized overhead in Fig. 9, where each VN had 16 S-VRs provided.

When we varied h_i and the number of failures, we found from Fig. 9(a) $\sim$ Fig. 9(c), a similar behavior on the normalized overhead as the *Nobel-EU* topology. When we increased h_i for the case of $\pi = 1$, the normalized overhead plots were entirely overlapped, except for the case with the large amount of failures, i.e., 10 failures for $h_i = 1$ in Fig. 9(a).

D. Dependent Virtual Router Failure

Substrate node failures have critical impacts on the virtual networks. For instance, when a substrate node fails, which could host VRs that belong to n VNs, this single substrate node failure affects services in n VNs at the same time. Hence, it is important to restore the failures in the VNs by quickly selecting proper S-VRs. Table VI presents the 95% CI of the normalized overhead for a different size of the S-VR set, given that $\langle \lambda, \pi \rangle$ was $\langle 1, 0 \rangle$, $\langle 0, 1 \rangle$ and $\langle 0.02, 0.98 \rangle$, respectively.

For $\lambda = 1$ and $\pi = 0$, the S-VR selection does not consider the residual bandwidth balancing on the substrate at all. In other words, it allocates as many S-VRs as possible to a substrate node as long as the bandwidth utilization on that node does not exceed threshold $T(= 0.8)$. For the optimal selection, $t = 0.8$., whereas it was 0.017 for $\lambda = 0$ and $\pi = 1$. If we increased λ to be slightly greater than 0 ($\lambda = 0.02$), the normalized overhead was reduced, and the t was about 0.021, close to 0.015.

TABLE VI: Normalized Overhead for *Germany50* ($h_i = 3$)

$\langle \lambda, \pi \rangle$	Size of S-VR set	95% CI of Normalized Overhead
$\langle 0, 1 \rangle$	12	$(11.87 \pm 6.39)\%$
	18	$(16.61 \pm 7.20)\%$
	24	$(18.80 \pm 4.96)\%$
$\langle 0.02, 0.98 \rangle$	12	$(6.11 \pm 4.34)\%$
	18	$(6.60 \pm 1.78)\%$
	24	$(5.30 \pm 1.43)\%$
$\langle 1, 0 \rangle$	12	$(2.54 \pm 2.46)\%$
	18	$(1.89 \pm 0.81)\%$
	24	$(1.64 \pm 0.68)\%$

To compare the computation time of the heuristic and the optimal selection, we created 30 VNs provisioned over the *Germany50* topology, and allocated 25 S-VRs for each VN for restricted standby provisioning. It took the heuristic 6.1% to 81.0% less computing time to find a solution compared to solving the optimization model exactly (Table VII), as w_{a1} was increased from 0 to 1. This is an important consideration in that reducing the selection time also helps to restore the network faster, thus minimizing the impact on the transient behavior. Thus, the time advantage of the heuristic in many cases is a desirable trait. The cost overhead, on the other hand, varies noticeably depending on the weight parameters. We can see that the cost overhead can be reduced by adjusting the value of λ and π. For instance, when λ was increased from 0 to 0.02, the normalized overhead was reduced from 14.5% to 14.0%. When $\lambda = 0.02$, the maximum residual bandwidth utilization on the substate nodes was 0.0145 for the optimal selection, and 0.0181 for the heuristic.

TABLE VII: *Germany50*: Computation Time ($h_i = 8$) (time in sec.)

$\langle \lambda, \pi \rangle$	Computation Time		Objective	
	Optimal	heuristic	Optimal	Heuristic
$\langle 1, 0 \rangle$	0.200	0.038	23.723	24.230
$\langle 0.02, 0.98 \rangle$	1.110	0.562	0.491	0.562
$\langle 0, 1 \rangle$	0.870	0.817	0.010	0.014

E. Observations

The key observations are summarized below.

- A small substrate network has limitations on provisioning enough S-VRs for each VN; this restriction causes high overhead for the heuristic selection, especially in cases of a large amount of concurrent VR failures. If the provider allocates the proper amount of S-VRs for each VN, the overhead of applying the heuristic can be reduced to less than 5% (on average) compared to the optimal solution. In other words, the heuristic is well suited for large substrate networks to dynamically restore VR failures in VNs.
- For independent failures, the failed VRs are usually from different locations. Thus, increasing λ can help improve the performance of the heuristic selection.
- For dependent failures, setting $\lambda = 1$ increases the residual bandwidth utilization on one or more particular substrate nodes if they host multiple selected S-VRs for different VNs. When π is dominant, slightly increasing λ can still achieve residual bandwidth balancing among the substrate nodes and reduce the normalized overhead of using the heuristic significantly.

IV. Related Work

Most of the current work on network virtualization focus on virtual network embedding or survivability [4]–[8]. Restoration from a link failure in a substrate network has been studied [9]. However, there is little work on optimal reconfiguration for virtual network restoration *after* a VR failure(s).

DaVinci [10] proposed a dynamic network reconfiguration framework that can periodically reallocate the bandwidth for multiple virtual networks, but it does not address the node reallocation or restoration problem. The paper [11] studied dynamically assigning resources to virtual networks. It uses the term stress to represent the number of virtual nodes or links that share the same substrate nodes or links. For example, a substrate node A's stress is the number of virtual nodes it hosts. The authors in [11] use node stress ratio to evaluate the node assignment. Ideally the node stress ratio is expected to be 1 at the optimal solution. In their work, dynamic reconfiguration has been briefly discussed. The reconfiguration cost was defined as a weighted summation of the reconfiguration rate, node and path switching rates.

There are several research network infrastructures [2], [12]–[15] that support building network virtualization testbeds. In [1], an experimental study on dynamic reconfiguration in a virtualized networking environment using autonomic management on the GpENI [2] inter-continental testbed was explored. It was found that reconfiguration with autonomic management is practically possible in a wide-area network. A geographical distance as a metric was used for selecting an S-VR, and the work tested single VR failure in a single virtual network on the GpENI experimental testbed. The scope of this work is to develop a robust optimal selection scheme that can be used in a virtual network environment such as GpENI for dynamic reconfiguration.

V. Conclusion and Future Work

Dynamic network virtualization becomes flexible and efficient in a virtualized network environment, especially with a centralized control system that has the capability to allocate substrate resources during the virtual network creation or reconfiguration. In this paper, we presented an optimization model and a heuristic to dynamically select optimal S-VRs to restore single VR failures occurring in multiple VNs.

The proposed optimization model considers a composite objective of minimum cost selection and balancing of residual bandwidth. The minimum cost selection objective consists of two cost functions from the operation and connectivity aspects, respectively. In this work, we considered the virtual interfaces to be homogenous, and studied the impact to the S-VR selection under a number of factors on three topologies, such as the number of concurrent virtual router failures, the capability of supporting S-VRs on a substrate node (h_i), and the number of S-VRs provided for each VN. Based on the origin of the VR failure, we categorized the VR failures to be either independent or dependent failures. Compared to the optimal S-VR selection, we found that the heuristic has low overhead when the substrate nodes have the capability to support more than one S-VR in service, and each VN was provided a proper number of S-VRs.

References

[1] X. Liu, P. Juluri, and D. Medhi, "An experimental study on dynamic network reconfiguration in a virtualized network environment using autonomic management." in *Proc. of IFIP/IEEE Symposium on Integrated Network Management (IM'2013)*, 2013, pp. 616–622.

[2] D. Medhi, B. Ramamurthy, C. Scoglio, J. P. Rohrer, E. K. Çetinkaya, R. Cherukuri, X. Liu, P. Angu, A. Bavier, C. Buffington *et al.*, "The GpENI testbed: Network infrastructure, implementation experience, and experimentation," *Computer Networks*, vol. 61, pp. 51–74, 2014.

[3] S. Orlowski, R. WessŁly, M. Piro, and A. Tomaszewski, "Sndlib 1.0 - survivable network design library." *Networks*, vol. 55, no. 3, pp. 276–286, 2010.

[4] M. Yu, Y. Yi, J. Rexford, and M. Chiang, "Rethinking virtual network embedding: substrate support for path splitting and migration," *SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 2, pp. 17–29, Mar. 2008.

[5] N. M. K. Chowdhury, M. R. Rahman, and R. Boutaba, "Virtual network embedding with coordinated node and link mapping," in *IEEE INFOCOM 2009*, 2009, pp. 783–791.

[6] G. P. Alkmim, D. M. Batista, and N. L. S. da Fonseca, "Optimal mapping of virtual networks," in *IEEE Globecom'2011*, 2011.

[7] J. Nogueira, M. Melo, J. Carapinha, and S. Sargento, "Virtual network mapping into heterogeneous substrate networks," in *IEEE Symposium on Computers and Communications (ISCC'2011)*, 2011, pp. 438–444.

[8] M. Rahman and R. Boutaba, "SVNE: Survivable virtual network embedding algorithms for network virtualization," *IEEE Transactions on Network and Service Management*, vol. 10, pp. 105–118, June 2013.

[9] D. Medhi and R. Khurana, "Optimization and performance of restoration schemes for wide-area teletraffic networks," *Journal of Network and Systems Management*, vol. 3, pp. 265–294, 1995.

[10] J. He, R. Zhang-shen, Y. Li, C. yen Lee, J. Rexford, and M. Chiang, "Davinci: Dynamically adaptive virtual networks for a customized internet," in *in Proc. CoNEXT*, 2008.

[11] Y. Zhu and M. Ammar, "Algorithms for assigning substrate network resources to virtual network components," in *IEEE INFOCOM*, 2006.

[12] www.geni.net.

[13] M. Hibler, R. Ricci, L. Stoller, J. Duerig, S. Guruprasad, T. Stack, K. Webb, and J. Lepreau, "Large-scale virtualization in the emulab network testbed," in *USENIX 2008 Annual Technical Conference on Annual Technical Conference*, 2008, pp. 113–128.

[14] "OneLab," http://www.onelab.eu/.

[15] "CoreLab," http://www.corelab.jp/.