Implementation of a Network Provisioning System with User-driven and Trusty Protection Management

H. Lim
Supercomputing Division, Korea Institute of Science and Technology Information (KISTI)
335 Daehangno, Yuseong-gu, Daejeon, Republic of Korea
[e-mail: hklim@kisti.re.kr]

Received October 14, 2014; revised June 16, 2015; accepted October 25, 2015; published November 30, 2015

Abstract

Proper management on user-driven virtual circuits (VCs) is essential for seamless operation of virtual networks. The Network Provisioning System (NPS) is useful software for creating user-driven VCs automatically and must take fault management into account for physical layer impairments on user-driven VCs. This paper addresses a user-driven and trusty protection management in an NPS with an open standard Network Service Interface (NSI), as a contribution to show how to implement the user-driven and trusty protection management required for user-driven VCs. In particular, it provides a RESTful web service Interface for Configuration and Event management (RICE) that enable management of a distinguished data and control plane VC status between Network Service Agents (NSAs) in the event of a node or link fault and repair in a domain. This capability represents a contribution to show how network and protection events in a domain can be monitored between NSAs (NPSs with the NSI) in multiple domains. The implemented NPS controls and manages both the primary and backup VC with disjoint path in a user-driven manner. A demonstration to verify RICE API's capability is addressed for the trusty protection in the dynamic VC network.

Keywords: Network provisioning system (NPS), User-driven and trusty protection management, Virtual circuit (VC), Network Service Interface (NSI), Network service agent (NSA)

A preliminary version of this paper appeared in ICSNC 2013, Oct. 27-31, Venice, Italy. This research was supported by the "User based Research Network Platform Service and Development Project" of KISTI funded by the Ministry of Science, ICT & Future Planning in 2014.

http://dx.doi.org/10.3837/tiis.2015.11.026 **ISSN : 1976-7277**

1. Introduction

The Network Service Interface (NSI) developed by the Open Grid Forum (OGF) is an open standard network interface for intra- and inter-domain virtual circuit (VC) services [1]. NSI-based network provisioning systems enable end users to reserve and allocate VCs. They represent recent and global trend for user-driven VC services in the multi-domain environment [2-6].

Recently, the management of network virtualization environments has started receiving special attention from the research community. Among the management issues physical layer fault management of network virtualization environments is crucial for guaranteeing seamless virtual network services irrespective of physical infrastructure impairments [2, 7-9]. A protection or rerouting method can be considered for fault restoration in user-driven VC networks. Each method has advantages and disadvantages in terms of various aspects, such as robustness, restoration time, and bandwidth efficiency. This article is focused on the protection method in VC networks. However the user-driven and trusty protection management technique in user-driven VC networks is still immature. Most of works on protection techniques in IP/MPLS and optical networks, which can be operated by an administrator rather than an application user, have depended on control plane of network devices and exposed the limitations to support the user-driven and trusty protection management [10-16]. To date, the majority of approaches in Network Provisioning Systems (NPSs), to provide dynamic VC services in a user-driven manner, also do not address the user-driven and trusty protection management issue [2-6].

In this article, a user-driven and trusty protection management required for guaranteed user-driven VC networks are implemented in an NPS with NSI. It consists of a Requester Agent (RA) and a Network Service Agent (NSA) with NSI. It provides a RESTful web service Interface for Configuration and Event management (RICE) for network and protection event notification. RICE API messages are used to provide NSAs in multi-domain with network fault/repair and VC protection/retrieval event information, in the event of a network fault/repair in a domain. This capability represents a contribution to show how network and protection events in a domain can be monitored between NSAs in multiple domains for the trusty protection. In a user-driven manner, a Network Resource Manager (NRM) in the NPS is considered to reserve, allocate, release and terminate both the primary and backup VC with disjoint path. For the user-driven and trusty protection management, the NPS has a feature that facilitates the management of both primary/backup VC data plane status, provided by the RICE API message.

The remainder of this paper is organized as follows. Section 2 presents the motivations for the user-driven and trusty protection management on user-driven VC services. Section 3 outlines the NPS with a user-driven and trusty protection management. Section 4 discusses the network and protection event notification service that is a trusty protection management function in the NPS. Section 5 gives a demonstration of the network and protection event notification service. Section 6 presents conclusions and future works.

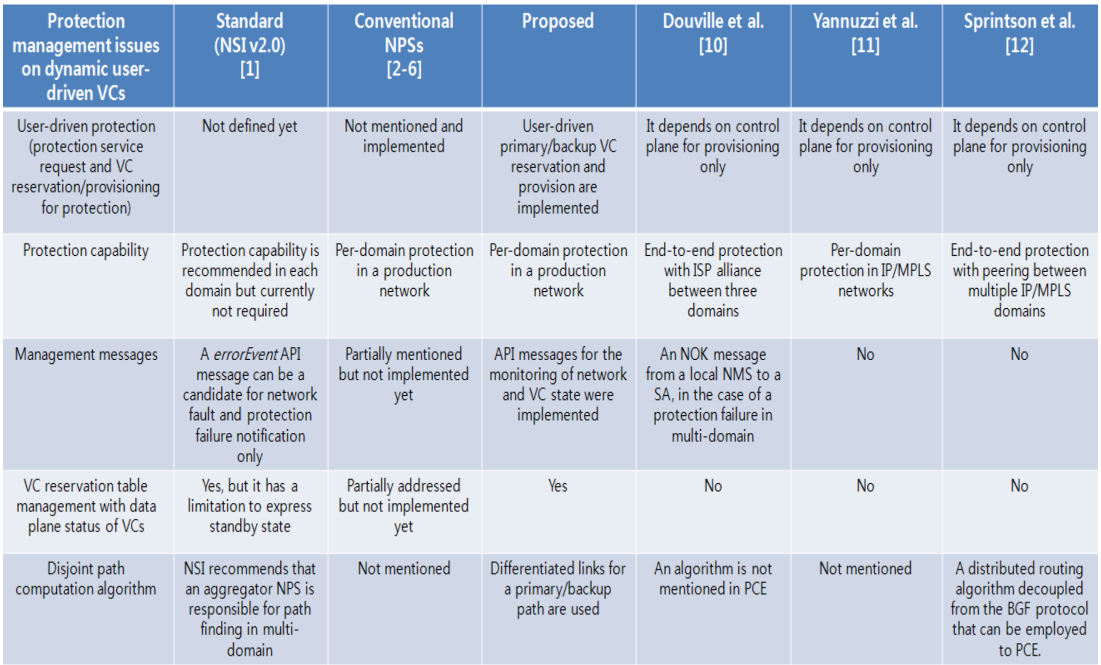
2. Why User-driven and Trusty Protection Management for User-driven VCs

2.1 Needs of User-driven and Trusty Protection Management

Two kinds of VC failures are defined in this paper: control plane failure due to errors in VC connection service lifecycle and data plane failure due to a link and node fault. The former is soft failure, whereas the latter is crucial failure because a link or node fault leads to permanent VC failures in networks [9]. For the user-driven VCs, a link or node fault in one domain results in VC data plane failures in multiple domains.

Protection management for user-driven VCs means protection of VCs and its specific management. Network devices in a domain need to support a protection capability. Protection by an NPS can be implicitly enabled by default on user VC service request. Path-disjoint backup VCs also need to be reserved, provisioned, released, and terminated by an NPS in a user-driven manner. Explicit control over both primary and backup VCs is necessary in order to maintain consistent control plane status and manage risk mitigation in the user-driven VC networks. Specifically, when primary VCs have failures due to a network fault in one domain, an NPS should be able to control and manage backup VCs implicitly via user requests. To search for a path-disjoint pair of VCs with the coincident bandwidth, an NPS has to be able to manage a reservation table for each primary/backup VC and to support a path computation engine. An NPS should be able to monitor network and VC state change in a domain for the trusty protection. Specific management API messages are required for the exchanging of a link or node fault and VC protection/retrieval event information between NPSs. With these messages, an administrator in one domain can monitor when user-driven VCs have data plane failures due to a link or node fault and whether VC protection is successful or not.

Table 1. A comparison between representative VC protection works and the proposed NPS in terms of protection management issues for guaranteed user-driven VC service in the dynamic multi-domain.



Protection management issues on dynamic user-driven VCs	Standard (NSI v2.0) [1]	Conventional NPSs [2-6]	Proposed	Douville et al. [10]	Yannuzzi et al. [11]	Sprintson et al. [12]
User-driven protection (protection service request and VC reservation/provisioning for protection)	Not defined yet	Not mentioned and implemented	User-driven primary/backup VC reservation and provision are implemented	It depends on control plane for provisioning only	It depends on control plane for provisioning only	It depends on control plane for provisioning only
Protection capability	Protection capability is recommended in each domain but currently not required	Per-domain protection in a production network	Per-domain protection in a production network	End-to-end protection with ISP alliance between three domains	Per-domain protection in IP/MPLS networks	End-to-end protection with peering between multiple IP/MPLS domains
Management messages	A *errorEvent* API message can be a candidate for network fault and protection failure notification only	Partially mentioned but not implemented yet	API messages for the monitoring of network and VC state were implemented	An NOK message from a local NMS to a SA, in the case of a protection failure in multi-domain	No	No
VC reservation table management with data plane status of VCs	Yes, but it has a limitation to express standby state	Partially addressed but not implemented yet	Yes	No	No	No
Disjoint path computation algorithm	NSI recommends that an aggregator NPS is responsible for path finding in multi-domain	Not mentioned	Differentiated links for a primary/backup path are used	An algorithm is not mentioned in PCE	Not mentioned	A distributed routing algorithm decoupled from the BGF protocol that can be employed to PCE.

2.2 Problem Statement on Related Works

The user-driven and trusty protection required for the user-driven VC networks cannot be supported by the control plane only. Issues required for user-driven and trusty protection should be addressed on both management and control plane in dynamic VC networks, which are to explicit control over both primary and backup VCs by an NPS in a user-driven manner, to manage a reservation table for each primary/backup VC, and to monitor network and VC state change in a domain using specific management API messages between NPSs. **Table 1** provides a comparison between representative VC protection works and the proposed NPS in terms of protection management issues for guaranteed user-driven VC services in the dynamic multi-domain.

The proposed NPS represents a working example to show how a user-driven protection service can be implemented based on the NSI framework, whereas most of works have mainly addressed administrator-driven inter-domain VC protection provided by control plane only in IP/MPLS and optical networks [10-12]. It should be noted that protection management for user-driven VCs for mission-critical applications should be a user-driven manner [2]. That is, a user rather than an administrator should be able to control and monitor backup VCs, as well as primary VCs.

The works of Douville et al. [10] and Spritson et al. [12] addressed end-to-end protection with ISP alliance (peering) in static multiple domains by an administrator, whereas NSI recommends, but does not currently require domain protection capabilities [1]. However those are not realistic scenarios to provide mission-critical applications with dynamic inter-domain VC protection in a user-driven manner. The conventional NPSs addressed per-domain protection capability in production networks by an administrator [2-6], whereas the proposed NPS, however, do address per-domain protection capability in production networks by a user rather than an administrator. On the other hand, Yannuzzi et al. [11] addressed per-domain protection that is responsible for its corresponding segment of an end-to-end VC path by an administrator.

For management messages to provide trusty protection service between domains of interest, an *errorEvent* API message defined in NSI can be a candidate for network fault and protection failure notification only. Attribute information for network and protection failure events should be defined in an *errorEvent* message. However, it cannot fully cover information required for protection management between NPSs in the dynamic multi-domain VC scenario. The proposed NPS addresses an API message strategy for providing network fault/repair and protection/retrieval events notification between NPSs. On the other hand, the work of Douville et al. [10] addresses the use of an *NOK* message from an Network Management System (NMS) in a source domain to a central Service Agent (SA) for an administrator, in the event of a protection failure in a source domain. On the other hand, the work of Monga et al. [2] among conventional NPSs mentioned the need of a fault trigger and transport path coordination exchange message between component domains, to provide end-to-end protection at application level for dynamic user-driven VCs.

In case primary VCs have failures due to a network fault and a network operator requires long time for its repair, an NPS could be able to control backup VCs implicitly via user requests, based on primary/backup VC data plane status management. Although NSI also provides end-to-end VC data plane status management, it lacks the ability to fully express additional data plane status, such as the standby state [1]. Both control plane and data plane status management of primary/backup VCs are essential for providing mission-critical applications with protection management in a user-driven manner. The works that have addressed VC protection in the static IP/MPLS and optical networks by an administrator exposes the limitation to control backup VCs, due to the dependency on control planes only for protection [10-12].

For disjoint-path computation, the work of Sprintson et al. [12] addressed a distributed routing algorithm decoupled from the BGF protocol that can be employed to PCE in the static multi-domain, while other works did not mention a specific algorithm in PCE. In summary, most of works regarding VC protection techniques in IP/MPLS and optical networks have exposed the limitations to reserve and control path-disjoint backup VCs in a user-driven manner, due to the dependency on control plane only to provide a protection [10-16]. On the other hand, few of works regarding administrator-driven VC protection in conventional networks have addressed the need of the trusty protection [10]. In case of the user-driven VC networks, the monitoring of network and primary/backup VCs is more required to trust the user-driven protection that is dynamically managed and controlled by both management and control plane. To date, the majority of approaches in Network Provisioning Systems (NPSs) to provide dynamic VC services in a user-driven manner, also do not address the user-driven and trusty protection management [2-6]. The implemented NPS represents a contribution to show how a user-driven and trusty protection management can be reflected in an NPS, based on both management and control plane.

3. NPS with User-driven and Trusty Protection Management

3.1 System Block

The implemented NPS consists of two agents (a Requester Agent (RA) and a Network Service Agent (NSA) with an RA and a Provider Agent (PA)), as shown in **Fig. 1**. In general an NSA can take on the role of a requester agent, a provider agent, or both. The RA provides a web-based GUI for users to make advance reservations. It has a VC reservation DB, a network topology DB, and a user account DB, and provides an AAA module for the basic user authentication and authorization process. The reservation DB manages control and data plane status information for each primary and backup VC. The NSA system consists of an NSI Handler (NSIH) with a Provider Agent (PA) and a Requester Agent (RA) to support intra and inter-domain network resource services. It also has a Path Computation and Resource Admission (PCRA), and a Grid User network interface Message Handler (GUMH) to manage and control intra-domain network resources, and a Configuration and Event Management Handler (CEMH), as shown in **Fig. 1**. The RA interfaces with the NSIH via the NSI and the CEMH via the RICE, respectively.

The NSIH executes advance reservation based VC connection management with the NSI interface. In addition, the PA in the NSIH interfaces with the PCRA via the Grid Network Service Interface (GNSI) for VC connection management of intra-domain network resources. The PA delivers requested reservation information to the PCRA through the GNSI and the PCRA performs path computation and admission control for local VC reservation. The PCRA reflects network fault information received from the CEMH to network topology information and has a VC reservation table with control/data plane status information for each primary and backup VC managed with ResvID. Using this reservation table, the PCRA controls admission for new VC reservation requests in intra domain. The PCRA interfaces with the GUMH to create and release virtual circuits on source/destination nodes. The GUMH exchanges control messages for the creation, release, and inquiry of primary and backup virtual circuits with network devices using the Grid User Network Interface (GUNI). The GUMH receives network fault/repair events via SNMP trap messages from network devices. To detect a router (node) fault event, periodic polling messages from the GUMH are received at network devices. VC protection/retrieval events can be detected using a *Query_VC* GUNI message from the GUMH. The CEMH provides an RA with network fault/repair and VC protection/retrieval event information received from the GUMH via RICE API messages. The CEMH internally interfaces with the PCRA to initialize and apply network topology information received from the RA and to request network topology renewal and VC data plane status information. In addition, the CEMH internally interfaces with the GUMH to accept network event information coming from network devices and to request VC protection/retrieval information inquiry.

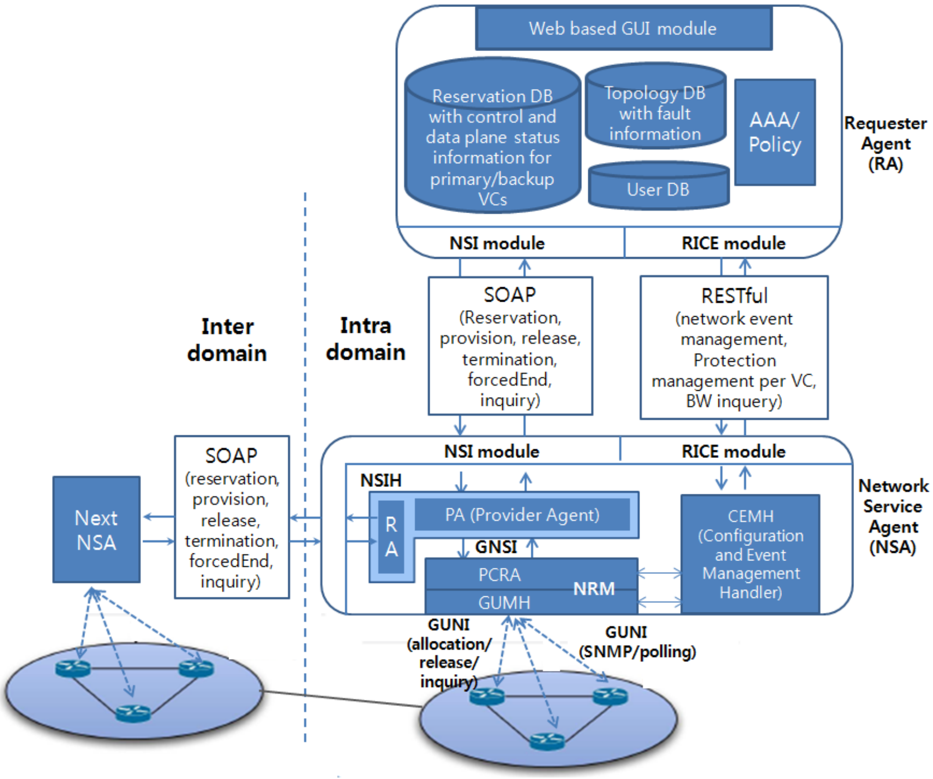

Web based GUI module
Reservation DB with control and data plane status information for primary/backup VCs
Topology DB with fault information
User DB
AAA/ Policy
Requester Agent (RA)
NSI module
RICE module
SOAP (Reservation, provision, release, termination, forcedEnd, inquiry)
RESTful (network event management, Protection management per VC, BW inquery)
Inter domain
Intra domain
NSI module
RICE module
NSIH
R A
PA (Provider Agent)
GNSI
PCRA
GUMH
NRM
CEMH (Configuration and Event Management Handler)
Network Service Agent (NSA)
Next NSA
SOAP (reservation, provision, release, termination, forcedEnd, inquiry)
GUNI (allocation/ release/ inquiry)
GUNI (SNMP/polling)

Fig. 1. NPS with user-driven and trusty protection management.

3.2 Interfaces in the NPS

1) NSI interface

A request message from the RA is delivered to the PA. The PA sends a confirmation or failure response message to the RA that indicates success or failure for a request message from the RA. An NSI message is delivered via the Simple Object Access Protocol (SOAP). All the NSI messages have a correlation ID as an identifier for a request and response message.

A *reserveRequest* message is used to request a VC reservation. A reservation request message has the following factors: ConnectionId (CID), service parameter, and path information. A start time, an end time, and a bandwidth are included in the service parameter. Direction (bi-direction by default) and the addresses of source/destination nodes are included in the path information. A *provisionRequest* message is used to allocate a reserved VC identified as a CID. To release a provisioned VC, a *releaseRequest* message is used. Further, a *terminateRequest* message is used to terminate a reserved or a provisioned VC. Finally, a *queryRequest* message is used to inquire about a reserved or provisioned VC. A *forcedEndRequest* message is used to notify an RA that the PA has administratively terminated a reservation [1]. If any reservation, provisioning, release, termination, or inquiry service has failed, the PA sends a failure message to the RA.

2) RICE interface

The RICE API message was implemented for network and VC protection event notification, application of network topology information to the NSA, and BW inquiry for a specific path with reservation duration. The RICE API messages that provide network and protection event notification services between NSAs are described in detail in Section 4.

3) GNSI interface

GNSI is an interface for a network resource reservation service defined by the Global Lambda Integrated Facility (GLIF) organization [17]. It is an internal interface between the NSIH and the PCRA that is used for reservation and connection management of the local domain. The GNSI messages implemented for resource reservation service are as follows. *GreateResourceResv* was used for resource reservation and *ProvisionResourceResv* was used to allocate a reserved network resource. *ReleaseResourceResv* and *ReleaseResourceProv* were used to release a reserved resource and a provisioned resource, respectively. *GetResourceProperty* was used to return attribute information corresponding to a reserved resource. A *GetAllReservedResources* message was used to inquire about the available BW between a requested reservation's start time and end time in a designated network path. To provide interoperability between the NSI and GNSI interfaces, a CID2ResvID mapping table is used in NSIH.

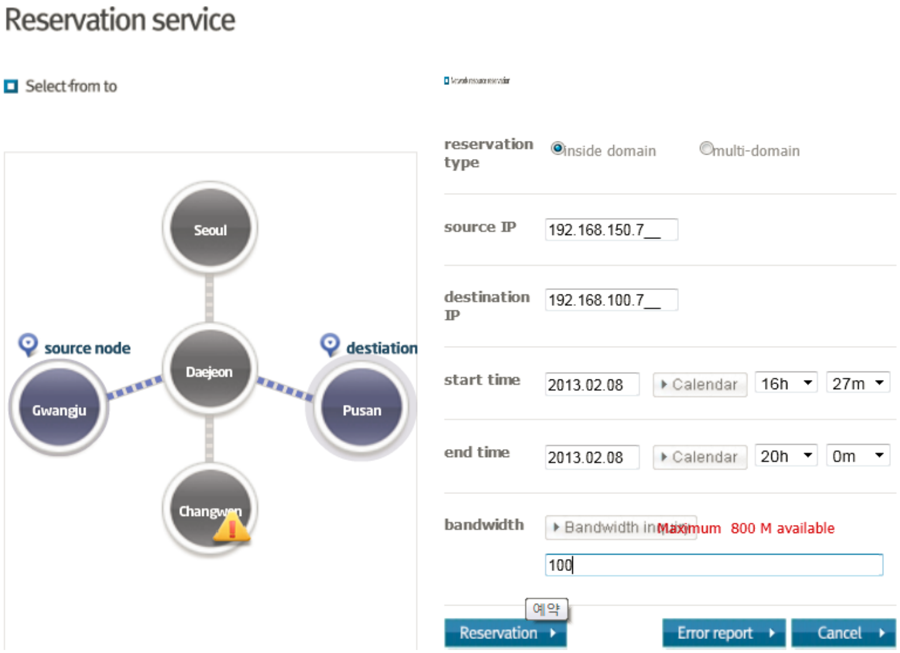
4) GUNI interface

The GUNI is an interface with network devices for VC creation, release, and inquiry. *Activate_VC* and *Deactivate_VC* messages are used to create and release primary and backup VCs on a requested network path, respectively. *Query_VC* is used to inquire about backup VCs working in an active or a non-active state (i.e., standby or failure state) in a backup node and link, in the event of a node or link fault, and primary VCs working in an active or a non-active state, in the case of a primary node or link repair. Each message includes information for the creation, release, and inquiry of VCs by telnet access to each network device. To receive SNMP trap messages from network devices in the event of a network node or link fault, an SNMP trap-based GUNI interface is also applied to GUMH.

3.3 VC Reservation and Provisioning for Protection

The NPS supports advance reservations of VCs at Layer 2 (VLANs) via an NSI interface, and Layer 3 (MPLS LSPs) via a GUI and an NSI interface. Two backup VCs (i.e., one bi-directional VC) in secondary (i.e., backup) links are reserved for protection, in addition to two primary VCs (i.e., one bi-directional VC) in primary links by an NSI reservation request. Since users can monitor a link or node fault in a topology map, they can reserve VCs on the remaining links and nodes, excluding fault nodes or links, as shown in **Fig. 2**.

Once a reservation is made, a VC provisioning step can be instantiated by the NSI provisioning request from an RA. Network device-specific GUNI messages in the dynamic VC network are used to initiate VC provisioning and release of the network devices. Two backup VCs (i.e., one bi-directional VC) in secondary links are provisioned for protection, in addition to two primary VCs (i.e., one bi-directional VC) provisioned in primary links by an NSI provisioning request message. Thus, a path-disjoint backup VC in standby state for protection of a primary VC in active state is enabled by the NPS. An active path is switched from a primary to a backup path in the event of a primary link or node fault [18].



Reservation service

Select from to

reservation type ● inside domain ○ multi-domain

source IP 192.168.150.7__

destination IP 192.168.100.7__

start time 2013.02.08 Calendar 16h 27m

end time 2013.02.08 Calendar 20h 0m

bandwidth Bandwidth in Maximum 800 M available

100

Fig. 2. Reservation request GUI.

4. Network and Protection Event Notification Service

For the reservation and provisioning of end-to-end VC services, NSI reflects the need for a widely accepted and standardized API messages. However, since NSI standardization does not yet address the user-driven and trusty protection management issues on NPSs, we have implemented RICE API messages, especially for the notifications of network fault/repair and protection/retrieval events between NSAs in multi-domain. The RICE API message represents a seed study case to show how network and protection events in a domain can be exchanged and monitored between NSAs in multiple networks. In this section, RICE API messages are presented in detail. Each network and protection event notification service mechanism for a local and multi-domain case is addressed between NSAs. The advantages of the notification service between NSAs for the trusty protection are finally explained.

4.1 RICE API Messages for Network and Protection Event Notification Service

1) InterfaceDown and InterfaceUp

When an interface of a network device has a fault, an SNMP trap message from a network device is delivered to the GUMH in NSA [19]. An *InterfaceDown* message is used to notify an RA of a fault interface on a network device in a domain. When a failure interface has been repaired, an SNMP trap message from the network device with the fault interface is delivered to the GUMH. An *InterfaceUp* message is used to notify an RA of a repaired interface on a network device in a domain.

2) NodeDown and NodeUp

To monitor a network device that has a fault, a periodic polling message from the GUMH is delivered to network devices. A *NodeDown* message is used to notify an RA of a fault network device in a domain. When a network device with a fault is retrieved, it can be monitored using both an SNMP trap and polling messages [19]. A *NodeUp* message is used to notify an RA of a repaired network device in a domain.

3) Primary2SecondarySuccess/Fail and Secondary2PrimarySuccess/Fail

Primary2SecondarySuccess and *Primary2SecondaryFail* messages are used to notify that backup VCs pre-assigned in secondary links in a domain are currently operating in the active state to protect VCs in a fault link and at least one backup VC is not operating in the active state, respectively. On the other hand, *Secondary2PrimarySuccess* and *Secondary2PrimaryFail* messages are used to provide notification that all VCs in a repaired primary link (i.e., interfaces) in a domain have been retrieved in the active state, and to notify that at least one VC has not currently been retrieved in the active state, respectively. The above events can be detected by sending *Query_VC* GUNI messages to network devices after receiving an SNMP trap message with a link or node fault/repair information from a network device.

4.2 Network and Protection Event Notification Service Mechanisms

1) Local domain case

RICE message flows for one network and protection notification service scenario in a local domain are shown in **Fig. 3**. GUNI and internal message flows between the PCRA, the CEMH, and the GUMH are also shown. It is assumed that a primary link with provisioned VCs has a fault. An SNMP trap message from the network device with a fault link is received at the GUMH. The GUMH notifies the CEMH of the fault link information and the CEMH requests the PCRA to renew the network topology and the primary VCs' data plane status in the VC reservation table. The CEMH sends an *InterfaceDownRequest* API message to the RA as a notification of a fault link and the RA renews the network topology and the reservation DB. An *InterfaceDownConf* API message is received at the CEMH in response. To detect protection success or failure, the CEMH makes a request to the GUMH to inquire about the data plane status information of the backup VCs. The GUMH detects the status information of backup VCs by sending a *Query_VC* GUNI message to the network device. If all the pre-assigned backup VCs are working in the active state, the GUMH sends a protection success notification message to the CEMH. The CEMH then requests that the PCRA renew the backup VCs' data plane status in the VC reservation table. Subsequently, the CEMH finally sends a *Primary2SecondarySuccessRequest* API message to the RA notifying it that protection was successful. The VC reservation DB then changes the data plane status of the backup VCs from standby to active state. Finally, the RA sends a *Primary2SecondarySuccessConf* API message to the CEMH in response.

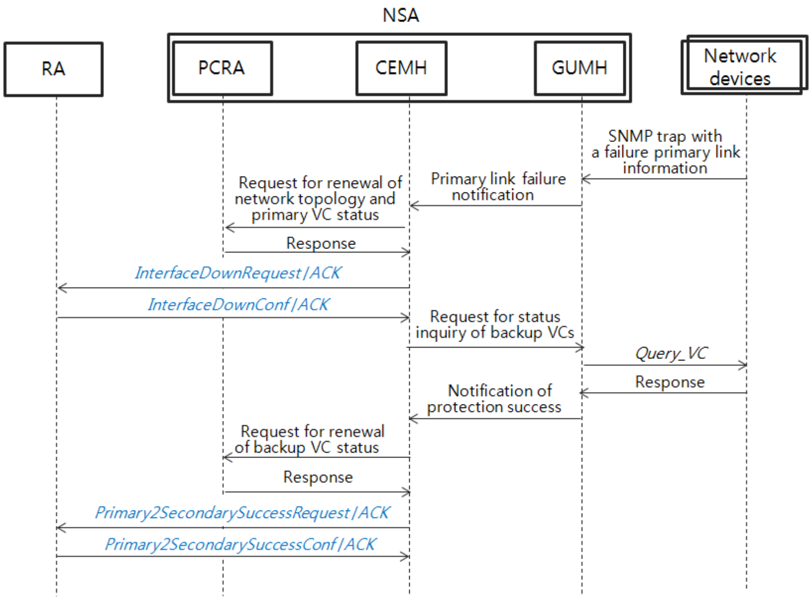

NSA
RA
PCRA
CEMH
GUMH
Network devices
SNMP trap with a failure primary link information
Primary link failure notification
Request for renewal of network topology and primary VC status
Response
InterfaceDownRequest/ACK
InterfaceDownConf/ACK
Request for status inquiry of backup VCs
Query_VC
Response
Notification of protection success
Request for renewal of backup VC status
Response
Primary2SecondarySuccessRequest/ACK
Primary2SecondarySuccessConf/ACK

Fig. 3. RICE message flows for one network and protection event notification service scenario in a local domain.

2) Multi-domain case

Fig. 4 presents a multi-domain network and protection notification service scenario between RA, Aggregator NSA and PA. A tree model case for multi-domain VC service and per-domain protection are assumed. In the case of the tree model for multi-domain VC service, a Requester Agent (RA) interfaces with an Aggregator (Global) NSA and an Aggregator NSA interfaces with Provider Agents (PAs) in the domains of interest. Each PA is only in charge of the VC service in its own domain [1]. Inter-domain VC connections in NSI are done by choosing appropriate Service Termination Points (STPs) such that the egress STP of one VC connection (STP b) corresponds directly with the ingress STP of the successive VC connection (STP c), as shown in **Fig. 4** [1]. A Service Demarcation Point (SDP) indicates a grouping of two STPs belonging to adjacent domains in **Fig. 4**. By using a shared point known as an SDP, NSI makes

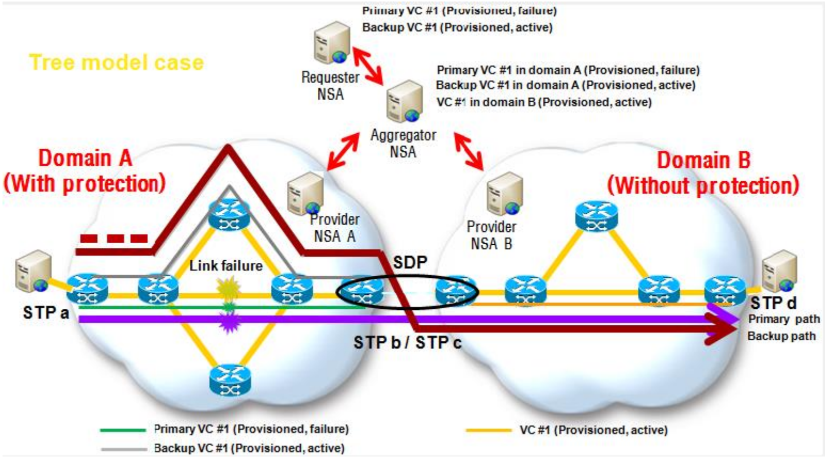

Primary VC #1 (Provisioned, failure)
Backup VC #1 (Provisioned, active)
Tree model case
Requester NSA
Primary VC #1 in domain A (Provisioned, failure)
Backup VC #1 in domain A (Provisioned, active)
VC #1 in domain B (Provisioned, active)
Aggregator NSA
Domain A (With protection)
Domain B (Without protection)
Provider NSA A
Provider NSA B
Link failure
SDP
STP a
STP b / STP c
STP d
Primary path
Backup path
Primary VC #1 (Provisioned, failure)
Backup VC #1 (Provisioned, active)
VC #1 (Provisioned, active)

Fig. 4. One network and protection event notification service scenario in multi-domain (per-domain protection is assumed).

inter-domain VC connections [1]. It is assumed that the disjoint backup VC #1 was reserved and provisioned to protect the primary VC #1 in domain A via NSI VC requests from an Aggregator NSA, which has received NSI VC requests from an RA from STP a to STP b. A primary and backup VC in domain A have same SDP for inter-domain VC connection in domain A and B. Domain B is a network without protection to reflect non-protection area in multi-domain and the VC #1 is only reserved and provisioned in domain B via NSI VC requests from an Aggregator NSA. Then the primary VC #1 for the RA comprises the primary VC #1 in domains A and the VC #1 in domain B, whereas the backup VC #1 for the RA consists of the primary VC #1 in domains A and the VC #1 in domain B. Domain A has a link fault and the primary VC #1 for the RA also has a failure as a result of a link fault in domain A, as shown in **Fig. 4**. User traffic on the primary VC #1 for the RA is switched to the disjoint backup VC #1. The PA-A needs to notify the Aggregator NSA and the RA of a link fault and protection success in domain A. RICE API messages between the PA-A, the Aggregator NSA, and the RA can be used to notify about a link fault and protection success in domain A, which leads to network and VC state monitoring in domains A and B for the RA and the Aggregator NSA.

Fig. 5 shows RICE message flows for the multi-domain network and protection notification service scenario in **Fig. 4**. When a link in domain A has a fault, the PA-A in domain A detects the link fault via an SNMP trap message from a network device in domain A. The PA-A notifies the RA of the link fault event in domain A through an Aggregator NSA using an *InterfaceDown* message. The RA then changes the data plane status of the primary VC #1 in its VC reservation DB from active to failure state. In the meantime, the PA-A queries the network devices in domain A about whether protection has been a success or a failure. In the case where protection success is detected (i.e., all backup VCs are active in domain A), the PA-A notifies the RA of protection success in domain A through an Aggregator NSA using a *Primary2SecondarySuccess* message. The RA then changes the data plane status of backup VC #1 in its VC reservation DB from standby to active state.

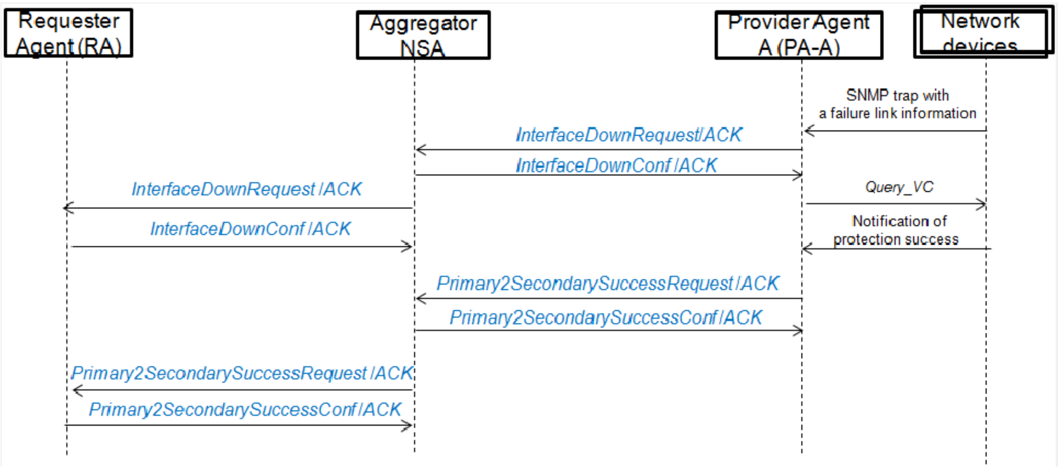
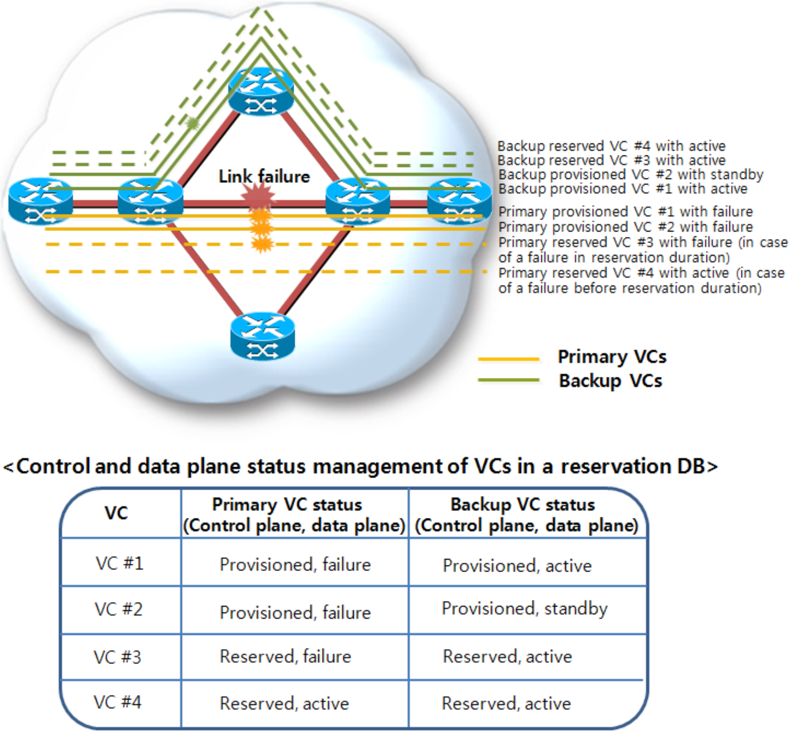

Requester Agent (RA)
Aggregator NSA
Provider Agent A (PA-A)
Network devices
SNMP trap with a failure link information
InterfaceDownRequest/ACK
InterfaceDownConf /ACK
InterfaceDownRequest /ACK
Query_VC
InterfaceDownConf /ACK
Notification of protection success
Primary2SecondarySuccessRequest /ACK
Primary2SecondarySuccessConf /ACK
Primary2SecondarySuccessRequest /ACK
Primary2SecondarySuccessConf /ACK

Fig. 5. RICE message flows for one network and protection notification service scenario in multi-domain.

4.3 Advantages of Network and Protection Event Notification Service

By using RICE API messages in local and multi-domain, an RA in one domain can observe whether its backup VCs are currently working in the active state or not in the event of a link or node fault in another domain, and whether primary VCs have been automatically retrieved or not after a link or node repair. With these messages, the data plane status of VCs in a RA's reservation DB can be managed in the event of a link or node fault/repair in a domain, as well as the control plane status of VCs, as shown in **Fig. 6**.


Link failure
Backup reserved VC #4 with active
Backup reserved VC #3 with active
Backup provisioned VC #2 with standby
Backup provisioned VC #1 with active
Primary provisioned VC #1 with failure
Primary provisioned VC #2 with failure
Primary reserved VC #3 with failure (in case of a failure in reservation duration)
Primary reserved VC #4 with active (in case of a failure before reservation duration)
Primary VCs
Backup VCs

<Control and data plane status management of VCs in a reservation DB>

VC	Primary VC status (Control plane, data plane)	Backup VC status (Control plane, data plane)
VC #1	Provisioned, failure	Provisioned, active
VC #2	Provisioned, failure	Provisioned, standby
VC #3	Reserved, failure	Reserved, active
VC #4	Reserved, active	Reserved, active

Fig. 6. Control and data plane VC status management in the event of a link fault.

In the case where a link or node fault event in a domain is not provided by a PA, an RA cannot detect that its VCs have data plane failures due to a link or node fault in one of domains of interest. With a protection success event message, an RA can switch its VCs control from primary VCs to backup VCs. In addition, if a protection failure event is not provided when there is a protection failure in a domain, an RA will assume that its backup VCs across the domains of interest are successfully operating in the active state even when some backup VCs in a domain are not active. Also, without a VC retrieval failure event message, the RA will assume that all primary VCs have been retrieved successfully in the active state following a link or node repair. It should be noted that a VC may not operate in the active state as a result of unexpected events, such as configuration intervention in network devices by an individual and an erroneously released VC by an administrator. An RA in one domain can request a network operator to recover unsuccessful VC protection/retrieval due to the unexpected events in another domain, as well as a network fault.

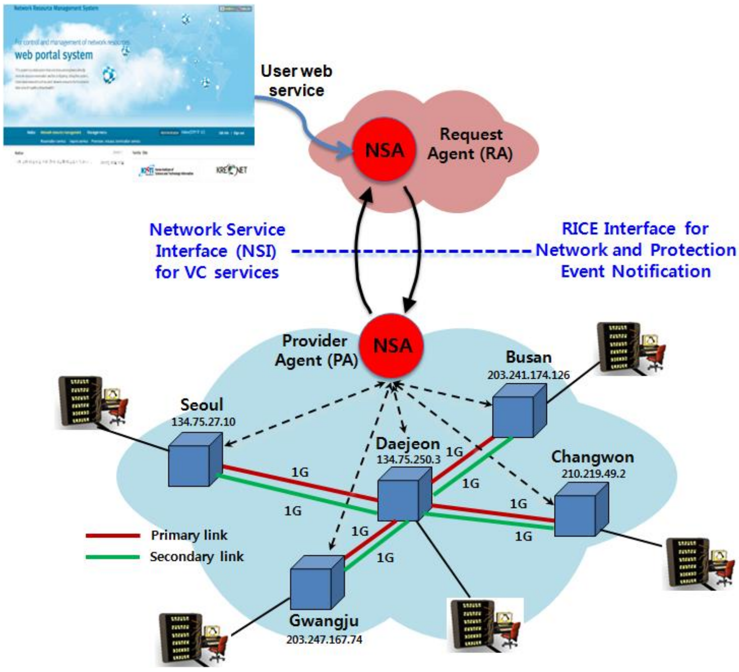


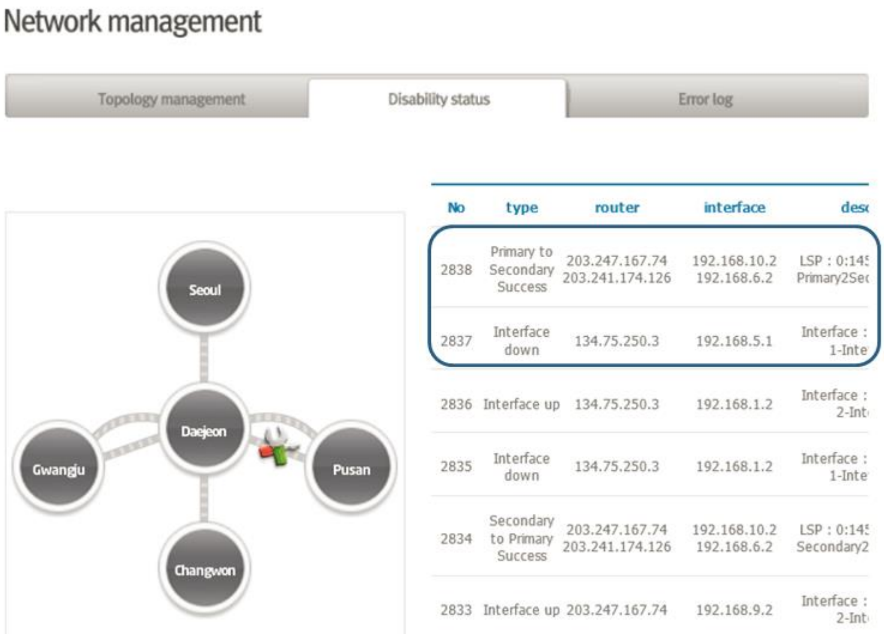
Fig. 7. Dynamic VC network controlled by the NPS with user-driven and trusty protection management.

5. Demonstration of Network and Protection Event Notification Service

5.1 Network Architecture

We have implemented a dynamic VC network with a user-driven and trusty protection management, which is physically distinct from the Korea Research Environment Open Network (KREONET) core network. NSI VC services can be made on the dynamic VC network, which comprises sections of five sites in the KREONET, as shown in **Fig. 7**. Each node is connected with two links (a primary and backup link) to provide a disjoint path for primary and backup VCs. A 1:1 protection for each VC in the network devices is enabled by the NPS. The PA operates as a server to process NSI request messages from the RA and operates as a client to create NSI confirmation messages, whereas it works as a client to create RICE request messages and a server to process RICE confirmation messages from the RA. The RA also operates as a client to create NSI request messages and a server to process NSI confirmation messages from the PA, whereas it works as a client to create RICE confirmation messages and as a server to process RICE request messages from the PA.

The dynamic VC network is designed to give disjoint paths for a link fault, although it does not provide disjoint paths for a node fault because of the star topology of the KREONET core network. A notification service demonstration focusing on a link fault/repair event is given in the next subsection.



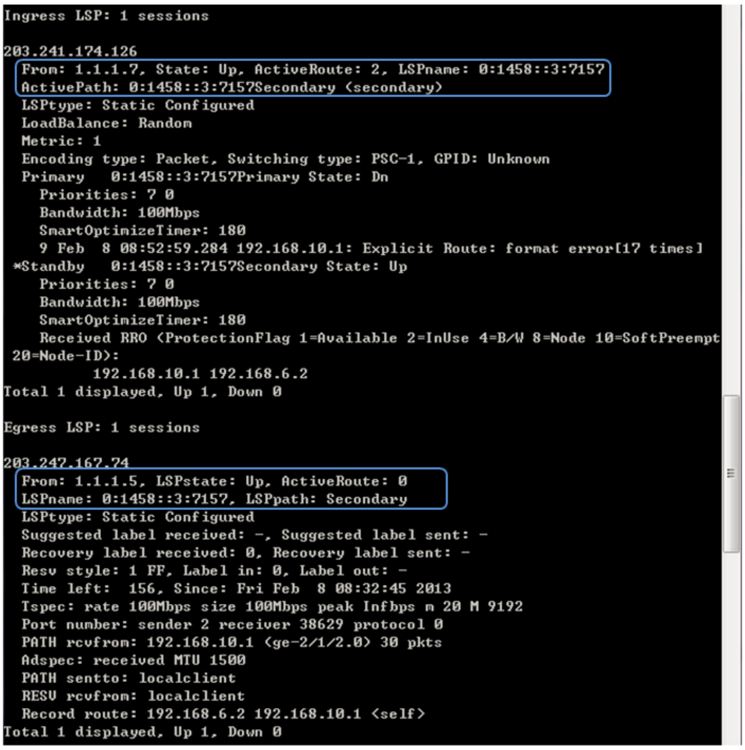
No	type	router	interface	des
2838	Primary to Secondary Success	203.247.167.74 203.241.174.126	192.168.10.2 192.168.6.2	LSP : 0:14 Primary2Sec
2837	Interface down	134.75.250.3	192.168.5.1	Interface : 1-Inte
2836	Interface up	134.75.250.3	192.168.1.2	Interface : 2-Int
2835	Interface down	134.75.250.3	192.168.1.2	Interface : 1-Inte
2834	Secondary to Primary Success	203.247.167.74 203.241.174.126	192.168.10.2 192.168.6.2	LSP : 0:14 Secondary2
2833	Interface up	203.247.167.74	192.168.9.2	Interface : 2-Int

Fig. 8. Link fault and protection success notification service GUI for an RA.

5.2 Experimental Results

A network and VC protection/retrieval notification service demonstration is addressed in the dynamic VC network. In this demonstration, an RA in one domain was implemented to communicate with a PA in another domain, because an RA in any domain should be able to request guaranteed VC services to a PA, as shown in **Fig. 7**.

Two unidirectional VCs (i.e., one bi-directional VC) with 100 Mbps BWs on a designated network path from the Gwangju site to the Pusan site were provisioned by the NPS on user request, together with two provisioned disjoint backup VCs. A primary link at the Daejeon site subsequently had a fault event. The link fault and VC protection success notification service GUI for an administrator in an RA is shown in **Fig. 8**. An administrator in an RA can be notified that a primary link at the Daejeon site has a fault via an *InterfaceDown* API message. In addition, the administrator in the RA can ascertain that backup VCs in secondary links are successfully working in the active state to protect primary VCs with a link fault via a *Primary2SecondarySuccess* message. A reservation DB in an RA changes the data plane status information of primary VCs from active to failure state, whereas it changes the data plane status information of backup VCs from standby to active state. By making use of these RICE API messages, an RA in one domain can detect whether its VCs in another domain are currently being protected as well as detect link or node fault event information in another domain. With these messages, the control and data plane status of each VC can be separately managed in the reservation DB of an RA when there is a network fault/repair event. **Fig. 9** verifies that two secondary VCs pre-assigned by the NPS are working in the active state in secondary links by exchanging signaling messages between network devices after a primary link fault.

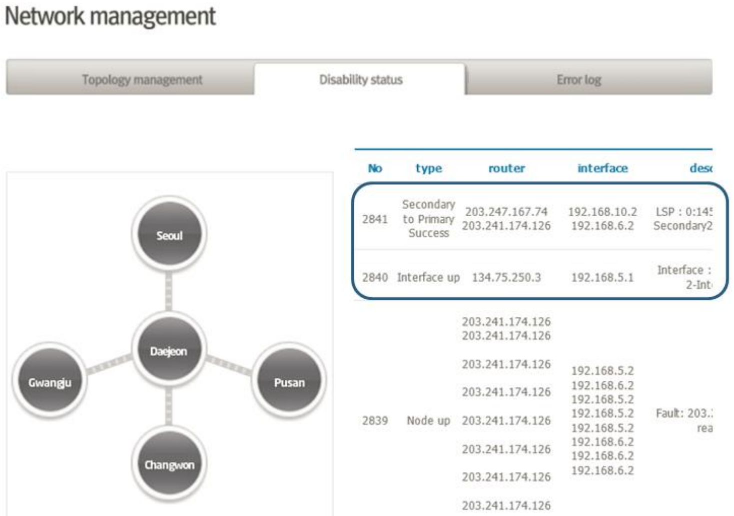


```
Ingress LSP: 1 sessions

203.241.174.126
  From: 1.1.1.7, State: Up, ActiveRoute: 2, LSPname: 0:1458::3:7157
  ActivePath: 0:1458::3:7157Secondary (secondary)
  LSPtype: Static Configured
  LoadBalance: Random
  Metric: 1
  Encoding type: Packet, Switching type: PSC-1, GPID: Unknown
  Primary   0:1458::3:7157Primary State: Dn
    Priorities: 7 0
    Bandwidth: 100Mbps
    SmartOptimizeTimer: 180
    9 Feb  8 08:52:59.284 192.168.10.1: Explicit Route: format error[17 times]
 *Standby   0:1458::3:7157Secondary State: Up
    Priorities: 7 0
    Bandwidth: 100Mbps
    SmartOptimizeTimer: 180
    Received RRO (ProtectionFlag 1=Available 2=InUse 4=B/W 8=Node 10=SoftPreempt
 20=Node-ID):
          192.168.10.1 192.168.6.2
Total 1 displayed, Up 1, Down 0

Egress LSP: 1 sessions

203.247.167.74
  From: 1.1.1.5, LSPstate: Up, ActiveRoute: 0
  LSPname: 0:1458::3:7157, LSPpath: Secondary
  LSPtype: Static Configured
  Suggested label received: -, Suggested label sent: -
  Recovery label received: 0, Recovery label sent: -
  Resv style: 1 FF, Label in: 0, Label out: -
  Time left:  156, Since: Fri Feb  8 08:32:45 2013
  Tspec: rate 100Mbps size 100Mbps peak Infbps m 20 M 9192
  Port number: sender 2 receiver 38629 protocol 0
  PATH rcvfrom: 192.168.10.1 (ge-2/1/2.0) 30 pkts
  Adspec: received MTU 1500
  PATH sentto: localclient
  RESV rcvfrom: localclient
  Record route: 192.168.6.2 192.168.10.1 <self>
Total 1 displayed, Up 1, Down 0
```

Fig. 9. Secondary VCs working in the active state following a primary link fault.

After the fault primary link at the Daejeon site has been repaired by a network operator, the link repair and VC retrieval success notification GUI for an administrator in an RA is as shown in **Fig. 10**. The administrator in the RA can be notified that the fault primary link was repaired via an *InterfaceUp* message. In addition, an RA in one domain can ascertain that its VCs in a repaired primary link of another domain have been successfully retrieved in the active state via a *Secondary2PrimarySuccess* message. The reservation DB changes the data plane status information of the primary VCs from failure to active state and renews the data plane status information of backup VCs from active to standby state. By making use of these API messages, an RA in one domain can ascertain whether its primary VCs in another domain were retrieved successfully or not, as well as obtain link or node repair event information in another domain.

Table 2 shows the failover/retrieval and failover/retrieval notification latency versus increase in the number of provisioned VCs with 20 *Mbps* BW per VC, on experiments of 1 *Gbytes* ftp data transfer over each primary VC from the Gwangju to the Pusan site. Failover latency provided by the network device in the case of a primary link fault at the Daejeon site was below 50 *ms*, which means average time that an active path for data transfer is switched from a primary to a backup path. The total failover notification latency of the RICE API messages (an *InterfaceDown* message and a *Primary2SecondarySuccess* message), composed of delivery time, processing time, and backup VCs inquiry time, was 2.2 *sec*, regardless of the number of provisioned VCs. On the other hand, retrieval latency of primary VCs due to a primary link repair increased according to the increase in the number of provisioned VCs before a link fault, because network devices require the time to rebuild primary VCs in a repaired link. Consequently, the total retrieval notification latency of the RICE API messages (an *InterfaceUp* message and a *Secondary2PrimarySuccess* message) between PA and RA,



No	type	router	interface	desc
2841	Secondary to Primary Success	203.247.167.74 203.241.174.126	192.168.10.2 192.168.6.2	LSP : 0:145 Secondary2
2840	Interface up	134.75.250.3	192.168.5.1	Interface : 2-Int
2839	Node up	203.241.174.126 203.241.174.126 203.241.174.126 203.241.174.126 203.241.174.126 203.241.174.126 203.241.174.126 203.241.174.126	192.168.5.2 192.168.6.2 192.168.5.2 192.168.5.2 192.168.5.2 192.168.6.2 192.168.6.2 192.168.6.2	Fault: 203. rea

Fig. 10. Link repair and VC retrieval success event notification GUI for an administrator in an RA.

Table 2. Failover/retrieval and failover/retrieval notification latency versus increase in the number of provisioned VCs with 20 *Mbps* BW per VC (1 *Gbytes* data transfer over each primary VC from the Gwangju to the Pusan site).

The number of provisioned VCs	VC protection due to a link fault		VC retrieval due to a link repair	
	Failover latency	Failover notification latency	Retrieval latency	Retrieval notification latency
2	≤50 msec	2.1 sec	3.8 sec	6.1 sec
4	≤50 msec	2.2 sec	7.7 sec	9.8 sec
6	≤50 msec	2.3 sec	11.4 sec	13.6 sec
8	≤50 msec	2.2 sec	15.3 sec	17.6 sec
10	≤50 msec	2.2 sec	19.2 sec	21.5 sec

composed of delivery time, processing time, and primary VCs inquiry time, also increased with the increase in the number of provisioned primary VCs, due to the increase of retrieval latency of primary VCs (i.e., primary VCs inquiry time).

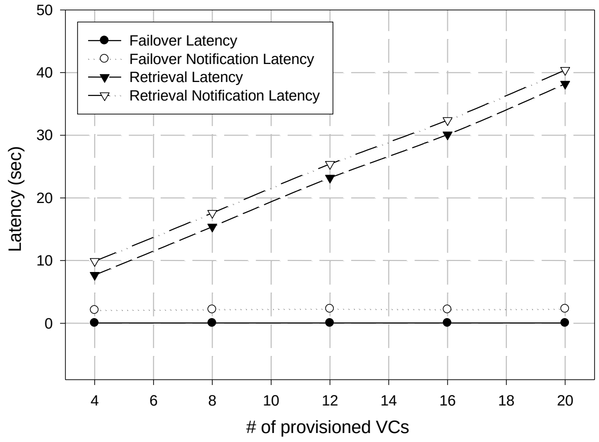


Fig. 11. Failover/retrieval and failover/retrieval notification latency versus increase in the number of provisioned VCs with 20 *Mbps* BW per VC (1 *Gbytes* ftp data transfer over each primary VC from the Gwangju to the Pusan site).

Fig. 11 shows the failover/retrieval and failover/retrieval notification latency versus increase in the number of provisioned VCs with 20 *Mbps* BW per VC, on experiments of 1 *Gbytes* ftp data transfer over each primary VC from the Gwangju to the Pusan site. Regardless of the number of provisioned VCs, failover and failover notification latency were 50msec and 2.2sec, respectively. On the other hand, retrieval latency of primary VCs due to a primary link repair increased according to the increase in the number of provisioined VCs before a link fault. Consequently, the total retrieval notification latency also increased with the increase in the number of provisioned primary VCs, due to the increase of retrieval latency of primary VCs (i.e., primary VCs inquiry time).

Finally we attempted to measure VC utilization during 1*Gbyte* ftp data transfer on a provisioned VC by NPS from the Gwangju to the Pusan site, as shown in **Fig. 12**. For data transfer on normal situation without a link fault (i.e., data transfer through a primary VC without a link fault), VC utilization was guarantted on a primary VC with each gurantted BW. In case of data transfer with a link fault, failover latency affected VC utilization reduction of about 0.336 *Mbps*, compared to VC utilization of data transfer on normal situation.

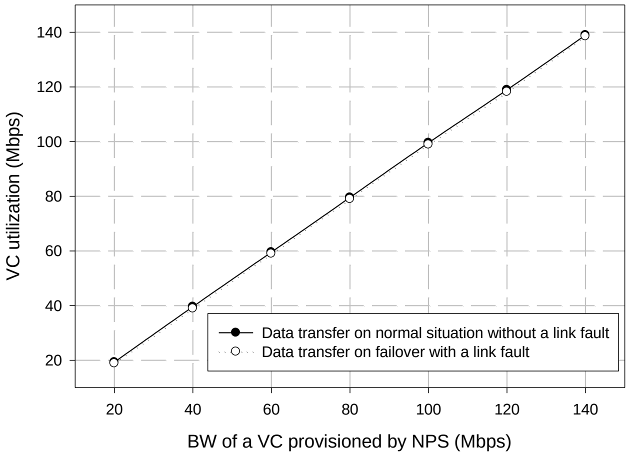


Fig. 12. VC utilization measurement without a link fault and with a link fault (1*Gbyte* ftp data transfer on a provisioned VC by NPS from the Gwangju to the Pusan site).

6. Conclusions and Future Works

Most of works on VC protection techniques in IP/MPLS and optical networks, that can be operated by an administrator rather than an application user, have exposed the limitations to reserve and control both the primary/backup VC in a user-driven manner, and to support the monitoring of primary/backup VC state for the trusty protection. Also, the majority of approaches in NPSs have not addressed the user driven and trusty protection management required for user-driven VCs. A user-driven and trusty protection management was implemented in an NPS, as a contribution to show how to implement protection management functions required for user-driven VC networks. It is capable of reserving and controlling a pair of path-disjoint VCs, to support protection in a user-driven manner. In particular, it provides network fault/repair and VC protection/retrieval notification service using RICE API messages between NSAs. With this capability, a Requester NSA in one domain could monitor whether VC protection in the event of a node or link fault in another domain has been successful or not and whether VC retrieval after a link or node repair has been successful or not. However, our demonstration needs to be extended to the dynamic multi-domain environment, to verify the trusty protection capability between NSAs more explicitly. The question of how to provide multi-layer protection management by NPSs forms part of our future work.

References

[1] G. Roberts, T. Kudoh, I. Monga and J. Sobieski, "NSI Connection Service Protocol v2.0," *GFD-173*, *OGF NSI-WG*, 2013. Article (CrossRef Link)

[2] I. Monga, C. Guok, W. E. Johnston and B. Tierney, "Hybrid Networks: Lessons Learned and Future Challenges Based on ESnet4 Experience," *IEEE Communications Magazine*, Vol. 49, Issue 5, pp. 114-121, May 2011. Article (CrossRef Link)

[3] F. Travostino, R. Keates, T. Lavian, I. Monga and B. Schofield, "Project DRAC: Creating an Application-aware Network," *Nortel Journal*, Vol. 22, No. 8, pp. 23-26, February 2005. Article (CrossRef Link)

[4] R. Krzywania, et al., "Network Service Interface: Gateway for Future Network Services," in *Proc. of Terena Networking Conference*, pp. 1-15, May 21-24, 2012. Article (CrossRef Link)

[5] N. Charbonneau, V. M. Vokkarane, C. Guok and I. Monga, "Advance Reservation Frameworks in Hybrid IP-WDM Networks," *IEEE Communication Magazine*, Vol. 49, Issue 5, pp. 132-139, May 2011. Article (CrossRef Link).

[6] Z. Zhao, J. Ham, A. Taal, R. Koning, C. Dumitru, A. Wibisono, P. Grosso and C. Laat, "Planning data intensive workflows on inter-domain resources using the Network Service Interface (NSI)," in *Proc. of SC Companion: High Performance Computing, Networking Storage and Analysis*, pp. 150-156, November 11-16, 2012. Article (CrossRef Link).

[7] R. Esteves, L. Granville, and R. Boutaba, "On the Management of Virtual Networks," *IEEE Communications Magazine*, Vol. 51, Issue 7, pp. 80-88, July 2013. Article (CrossRef Link)

[8] S. Peng, R. Nejabati, and D. Simeonidou, "Impairment-Aware Optical Network Virtualization in Single-Line-Rate and Mixed-Line-Rate WDM Networks," *Journal of Optical Communications and Networks*, Vol. 5, No. 4, pp. 283-292, April 2013. Article (CrossRef Link).

[9] H. Lim and Y. Lee, "Toward Reliability Guarantee VC Services in an Advance Reservation based Network Resource Provisioning System," in *Proc. of Int. Conf. on Systems and Networks Communications*, pp. 112-120, October 27-31, 2013. Article (CrossRef Link)

[10] R. Douville, J.-L. Roux, J. Rougier, and S. Secci, "A Service Plane over the PCE Architecture for Automatic Multi-domain Connection-Oriented Services," *IEEE Communication Magazine*, Vol. 46, Issue 6, pp. 94-102, June 2008. Article (CrossRef Link).

[11] M. Yannuzzi, et al., "On the Challenges of Establishing Disjoint QoS IP/MPLS Paths Across Multiple Domains," *IEEE Communication Magazine*, Vol. 44, Issue 12, pp. 60-66, December 2006. Article (CrossRef Link).

[12] A. Sprintson, M. Yannuzzi, A. Orda, and X. Masip-Bruin, "Reliable Routing with QoS Guarantees for Multi-Domain IP/MPLS Networks," in *Proc. of IEEE Int. Conf. on Computer Communications*, pp. 1820-1828, May 6-12, 2007. Article (CrossRef Link).

[13] C. Huang and D. Messier, "A Fast and Scalable Inter-Domain MPLS Protection Mechanism," *Journal of Communications and Network*, Vol. 6, No. 1, pp. 60-67, March 2004. Article (CrossRef Link).

[14] A. Urra, E. Calle, J. L. Marzo and P. Vila, "An Enhanced Dynamic Multilayer Routing for Networks with Protection Requirements," *Journal of Communications and Networks*, Vol. 9, No. 4, pp. 377-382, December 2007. Article (CrossRef Link).

[15] M. German, et al., "On the challenges of finding two link disjoint lightpaths of minimum total weight across an optical network," in *Proc. of European Conf. on Network and Optical Communications /Optical Cabling & Infrastructure*, pp 217-224, June 10-12, 2009. Article (CrossRef Link)

[16] H. Yu, V. Anand, C. Qiao, and H. Di, "Migration based protection for virtual infrastructure survivability for link failure," in *Proc. of Optical Fiber Communication Conference/National Fiber Optic Engineers Conference*, pp. 1-3, March 6-10, 2011. Article (CrossRef Link).

[17] Y. Tsukishima, M. Hayashi, T. Kudoh, et al., "Grid Network Service-Web Services Interface Version 2 Achieving Scalable Reservation of Network Resources Across Multiple Network Domains via Management Plane," *IEICE Transactions on Communications*, Vol. E93-B, No. 10, pp. 2696-2705, Oct. 2010. Article (CrossRef Link).

[18] R. Hughes-Jones, Y. Xin, G. Karmous-Edwards and J. Strand, *Grid Networks*, Wiley, 2nd Edition, New York, 2006.

[19] K. Ogaki, M. Miyazawa, T. Otani, and H. Tanaka, "Prototype demonstration of integrating MPLS/GMPLS network operation and management system," in *Proc. of Optical Fiber Communication Conference*, pp. 1-8, March 5-10, 2006. Article (CrossRef Link)

H. Lim received his Ph. D degree in Information and Communications from the Gwangju Institute of Science and Technology (GIST), South Korea, in 2006. He joined the Korea Institute of Science and Technology Information (KISTI) in 2006 as a senior researcher. He was a core network researcher in the architectural design and development of a user-driven network provisioning system deployed in a production research network in Korea. He is currently researching the management issues of user-driven virtual networks. He has published more than 50 refereed papers in the area of user-driven virtual networks, Information Centric Networking (ICN) and optical networks in archival journals and conference proceedings.