Toward Reliability Guarantee VC Services in an Advance Reservation based Network Resource Provisioning System

H. Lim
Dept. of High Energy Physics
California Institute of Technology
Pasadena, CA, USA
hklim@caltech.edu

Y. Lee
Dept. of Computer Engineering
Mokpo National University
Mokpo, Jeon-Nam, South Korea
yhlee@mokpo.ac.kr

***Abstract*— The most representative research network in Korea, KREONET, has developed DynamicKL, an advance reservation based network service agent for user driven virtual circuit services. DynamicKL provides reservation, provisioning, release, termination, and inquiry web services for network resources by using an open standard network service interface (NSI), as well as web services for network resources by using a GUI interface. In addition, it has the RESTful web service Interface for Configuration and Event management (RICE) to support a protection management function for virtual circuits and reservations. In this paper, a protection management per virtual circuit (VC) for provisioned VCs and reservations is addressed in the DynamicKL framework, as a contribution to the VC protection management issue, which results in more manageable and reliable VC services compared to other advance reservation frameworks. An administrator can detect successful or unsuccessful VC protections in the event of a primary link failure and successful or unsuccessful VC retrievals after a primary link repair, by using RICE.**

Keywords- Advance reservation, Network Service Agent (NSA), Network Service Interface (NSI), Dynamic provisioned network resource, DynamicKL, Virtual circuit protection, Primary link, Secondary link, Link failure.

I. INTRODUCTION

Recently, most dynamic provisioning in advanced research networks have developed and deployed advance reservation based network resource provisioning systems for big data transfers to support various application areas, for example, DRAGON, OSCARS, DRAC, AutoBHAN, EnLIGHTened, PHOSPHOROUS, and G-Lambda, [1][2][3][4][5][6][7]. They have their own framework for only network resources or for both grid and network services. Some of them have a standard interface for multi-domain services and the others have their own interface.

The Network Service Interface (NSI) developed by the Open Grid Forum (OGF) is a standard interface for network resource reservation and control in intra or inter-domain [8][9][10]. An NSI based resource reservation and provisioning system can improve productivity for data intensive research projects, for example, reserving and allocating available network resources (i.e., virtual circuits) automatically for large-scale data applications between multi-domains, such as Large Hadron Collider (LHC) in the field of High Energy Physics (HEP). G-Lambda A/K (the latest version of G-Lambda), OSCARS, AutoBHAN, and OpenDRAC (the latest version of DRAC) have been implementing the standard NSI interface in their frameworks [11].

Dynamically provisioned network resources, such as VCs, are recognized as extremely useful capabilities for many types of network services. However, to date the majority of approaches to such services do not address protection management per VC to protect data traffics in VCs in the event of node or link failures, which provide manageability and reliability guarantees VC services in advance reservation frameworks.

We present Dynamic circuit based advance reservation system of KRLight (DynamicKL) based on web services, which consists of the Network Service Agent (NSA) and a web portal server. In particular, DyamicKL provides the RESTful web service Interface for Configuration and Event management (RICE) web service interface for the protection management per VC for virtual circuits (VCs) and reservations in the event of a link failure, as well as the NSI and Graphic User Interface (GUI) web services for reservation, provisioning, release, termination, and inquiry. Protection management in DynamicKL is provided per VC for provisioned VCs and reservations in case of a link failure, a feature that contributes to managing failure and protection status information per VC in a primary and backup VC reservation DB. This capability constitutes a dominant, important difference from other advance reservation systems. With this capability, an administrator can detect when backup VCs are successfully or unsuccessfully working as active paths to protect primary VCs and primary VCs are successfully or unsuccessfully retrieved as active paths after a primary link repair. Because a primary and backup reservation DB separately manage failure information per each primary/ backup VC, it is possible to establish VCs and to terminate reservations in backup links for provisioning and termination requests of reservations with a primary link failure, by delivering NSI provisioning messages with backup interface information.

In Section II, other advance reservation frameworks are introduced as related works. The DynamicKL framework with protection management is addressed in Section III. In Section IV, protection management function per VC is addressed to provide more manageable and reliable VC

 ISBN: 978-1-61208-305-6

services more detail, which differentiates this approach from other advance reservation frameworks. To verify its capabilities, a protection management demonstration for virtual circuits with a link failure is presented in Section V. Finally Section VI gives conclusions and implications.

II. Related Works For Advance Reservation Framework

In this section, we introduce architecture and development of other advance reservation frameworks selected as representative related research. They are mainly focused on VC service technology issues other than management issues such as protection management per VC, for reliability and manageability guarantee VC services to protect data traffics to disjointed VCs in the event of a link or node failure.

A. DRAGON

Dynamic Resource Allocation via GMPLS Optical Networks (DRAGON) [1] is a project that allows dedicated network resources dynamically to link computational clusters, storage arrays, and other instruments into distributed topologies. GMPLS [12] is used to create virtual circuits for both optical and Ethernet domains and DRAGON creates Layer 1 and Layer 2 virtual circuits. The key components of DRAGON software consist of Virtual Label Switched Router (VLSR), Network Aware Resource Broker (NARB), Application Specific Topology Builder (ASTB), and Resource Computation Engine (RCE). To create virtual circuits that cover various domains, the NARB acts as the entity that represents a local autonomous system or a domain [1]. The main role of VLSR is to control Ethernet switches via the GMPLS control plane. RCE and ASTB are used to compute network resources needed for virtual circuit provisioning.

B. OSCARS

On-Demand Secure Service and Advance Reservation System (OSCARS) is a user driven network software developed to support dynamic virtual circuits (VCs), for large scale data applications such as Large Hadron Collider (LHC) research using the Energy Science Network (ESnet) in the US [2][13]. The main objective of OSCARS is to allow application programmers and end-users to set up advance reservations for VCs [2]. MPLS-TE and RSVP-TE protocols are used to make advance reservations and to allocate dedicated bandwidth on demand. The Label Switched Paths (LSPs) are created for Layer 2 and Layer 3 circuits using OSCARS software [2][13]. The key components of OSCARS consist of the Reservation Manager (RM), Path Setup Subsystem (PSS), Bandwidth Scheduler Subsystem (BSS), Authentication, Authorization and Accounting Subsystem (AAA) [2]. OSCARS is currently implementing open standard interface (NSI) in its framework.

C. AutoBHAN

AutoBAHN is a GEANT-provided provisioning tool that integrates with an NREN's own systems to facilitate the multi-domain dynamic circuit provisioning service [3]. AutoBAHN eliminates the long established problem of manually provisioning multi-domain circuits, reducing this time from weeks to a matter of minutes, even seconds. AutoBAHN easily negotiates the different networking technologies deployed by the different domains. Interoperability with other BoD systems is achieved through the Inter Domain Controller Protocol (IDCP), and the Network Services Interface (NSI) Protocol, fully enabling global connections.

The AutoBAHN system is based on the Inter-Domain Manager (IDM), a module responsible for the inter-domain operation of circuit reservation on behalf of a domain [3].

D. DRAC

The Dynamic Resource Allocation Controller (DRAC) [4] is a network service to support network resources automatically and dynamically, to meet application requirements in SURFnet (the national education and research network in the Netherlands). DRAC acts as an agent of the various applications, brokering and configuring on an end-to-end basis all the necessary pieces of the network, regardless of the type of network – circuit or packet, wireless or wired network [4]. DRAC allows very large number of flows of packets or low-latency applications dynamically through Layer 1 circuit instead of Layer 3. DRAC simply create and release optical circuits as application requirements [4].

DRAC has been extended to OpenDRAC. Open DRAC is an open source project that is developing a middleware that allows network control by users and applications [13][14]. It is currently compatible with open standard interface (NSI).

E. G-Lambda

G-Lambda [7] is a project to provide users with virtual dedicated circuits for both grid and network resources in Japan. The GNS-WSI [15] in G-LAMBDA defines a set of messages to be sent to Network Resource Manager (NRM) from Grid Resource Scheduler (GRS). These include messages to reserve, activate, release, and inquire VCs.

The G-Lambda framework consists of a GRS, which behaves as a Grid Scheduler, and Resource Managers (NRM for network resources and CRM for computing resources), which manage each local network or computing resource. GRS and RMs work together to provide users with co-allocation and resource reservations. The GRS provides a web service interface to user clients using the web service resource framework (WSRF) [7][15]. The NRM is responsible for path virtualization between endpoints, local scheduling, and activation/de-activation of VCs [7][13][15]. G-Lambda is currently extended to G-Lambda-A/K with open standard interface (NSI).

 ISBN: 978-1-61208-305-6

III. DYNAMICKL FRAMEWORK WITH PROTECTION MANAGEMENT

A. DynamicKL System Block

DynamicKL consists of a web portal server and a NSA, as shown in Fig. 1. The web portal server provides a web-based user interface for users to make advance reservations. It has a primary and backup 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. Primary and backup reservation DBs separately manage failure and protection status information per each primary and backup VC. The NSA system consists of an NSI Handler (NSIH) with a Provider Agent (PA) and an Requester Agent (RA) to support network resource service in intra or inter domain, a Path Computation and Resource Admission (PCRA) and a G-UNI Message Handler (GUMH) to manage network resource in intra domain, and a Configuration and Event Management Handler (CEMH), as shown in Fig. 1. The web portal server interfaces with the NSIH through the Network Service Interface (NSI) and the CEMH through the RICE, respectively.

The NSIH executes advance reservations based connection management in intra or inter-domain with the NSI interface. Also, the PA in the NSIH interfaces with the PCRA through the GNSI interface for connection management for intra domain network resources. The PA delivers requested reservation information to the PCRA through the GNSI interface and the PCRA performs path computation and an admission control for local network resource reservation. The PCRA reflects node or link failure information received from the CEMH in network topology information and has a primary and backup VC reservation table managed with ResvID. By using them, the PCRA controls admission for new VC reservation request. The PCRA interfaces with the GUMH for the creation and release of virtual circuits on a network path requested by a user. The GUMH exchanges control messages for creation, release and inquiry of primary and backup virtual circuits with network devices through the GUNI interface. The GUMH receives network failure/repair events by SNMP trap messages from network devices. To detect a router (node) failure event, periodic polling messages from the GUMH are received at network devices. VC protection/retrieval events can be detected by using a *Query_VC* GUNI message from the GUMH. The CEMH provides a management plane with network event and VC protection/retrieval event information received from the GUMH through RICE API messages. The CEMH internally interfaces with the PCRA, to initialize and apply network topology information received from the web portal server, and to request renewal of network topology and backup/primary VC reservation table information. Also, the CEMH internally interfaces with the GUMH, to detect network event information from network devices and to request VC protection/retrieval information inquiry.

The NSI is a standard interface for network resource reservation and control between inter domains defined by Open Grid Forum (OGF), in partnership with the Global Lambda Integrated Facility (GLIF) organization. The Grid Network Service Interface (GNSI) is an interface between

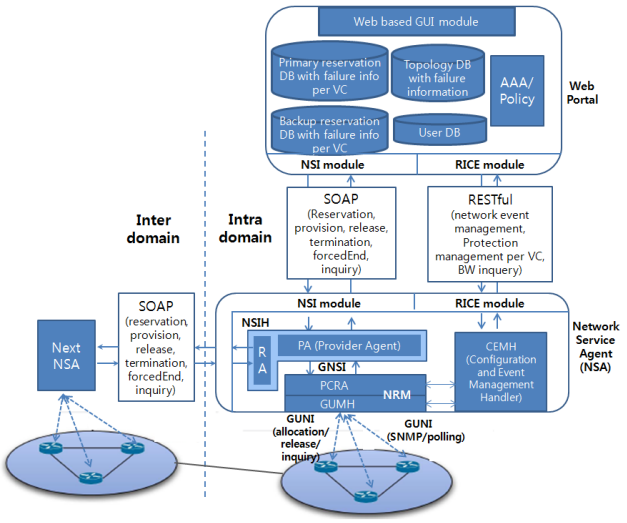


Fig. 1. DynamicKL framework with a protection management per VC.

NSIH and PCRA used for the reservation and connection management of the local domain [16]. The Grid User Network Interface (GUNI) is an interface for VC creation, release and inquiry. The RICE is an interface for network event management, especially for protection management per VC.

B. Interfaces in DynamicKL

1) NSI interface

NSI messages [8][9][10] for network resource control are shown in Table I. A request message from the RA is delivered to the PA. The web portal server plays a role of the RA for user VC services. The PA sends a confirmation or failure response message to the RA as identification of success or failure for a request message from the RA. A NSI message is delivered through Simple Object Access Protocol (SOAP). All of the NSI messages have a correlation ID as an identifier for a request and response message and a request message has a next NSA address for inter domain VC service. All of the NSI messages are defined in Table I.

TABLE I. NSI V1.1 MESSAGES FOR NETWORK RESOURCE SERVICE

Message	Description
reserveRequest	A message from RA that requests a network resource to PA for a connection between two nodes
provisonRequest	A message from RA that allocates a reserved network resource to PA
releaseRequest	A message from RA that releases an allocated network resource and maintains reservation for the network resource to PA
terminateRequest	A message from RA that terminates an allocated or a reserved network resource to PA
queryRequest	A message from RA that inquiries connection status to PA
forcedEndRequest	A message from PA that notifies RA that PA administratively terminated a reservation

 ISBN: 978-1-61208-305-6

A *reserveRequest* message is used to request a VC reservation. A reservation request message has following factors: globalReservationId, connectionId (CID), service parameter and path information. A starting time, an end time and a bandwidth are included in service parameter. Direction (bi-direction as a default), addresses of source/destination nodes are included in path information. A *reserveConf* message is used to response for a reservation request. [8][9][10]. For provisioning of a reserved VC identified as a CID, a *provisionRequest* message is used. For release of a provisioned VC, a *releaseRequest* message is used and a *releaseConf* message is used for response. Also, a *terminateRequest* message is used for termination of a reserved or a provisioned VC and a *terminateConf* message is used for a response. Finally, a *queryReqeust* message is used for inquiry of a reserved or provisioned VC and a *queryConf* message is used for response. A *forcedEndRequest* message is used to notify RA that PA administratively terminated a reservation [8][9][10]. If any service for reservation, provisioning, release, termination, and inquiry has failed, PA sends a failure message to RA.

2) GNSI interface

GNSI is an interface for a network resource reservation service defined by the GLIF organization [16]. GNSI has been previously implemented in a network resource provisioning system for local domain VC service by us [16]. We make use of it as an internal interface in DynamicKL for intra domain VC services. The GNSI messages implemented for resource reservation service is as follows. *GreateResourceResv* is used for resource reservation and *ProvisionResourceResv* is used to allocate a reserved network resource. *ReleaseResourceResv* and *ReleaseResourceProv* are used to release a reserved resource and a provisioned resource, respectively. *GetResourceProperty* is used to return attribute information corresponding to a reserved resource. The *GetAllReservedResources* message is used to inquire about an available BW between a requested reservation starting time and a requested reservation end time in a designated network path [16]. To provide interoperability between the NSI and GNSI interfaces, a CID2ResvID mapping table is used in NSIH.

3) GUNI interface

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

4) RICE interface

We designed and implemented the RESTful web service Interface for Configuration and Event management (RICE) for the application of network topology information to the NSA, BW inquiry for a specific path with a reservation duration, and network event management, especially for protection management per VC for provisioned VCs and reservations with a link failure. The RICE API messages for protection management per VC will be described in detail in section III.

C. Virtual Circuit Provisioning by DynamicKL

DynamicKL supports advance reservations of VCs at layer 2 (VLANs) through a NSI interface, and layer 3 (MPLS LSPs) through a NSI and GUI interface. In this capacity, DynamicKL is used as an intra domain controller for network resources within KREONET. DynamicKL also functions as a inter-domain controller which has the capability to communicate with other intra domain NSAs through NSI interface, as shown Fig. 1. DynamicKL is used to allow application programmers and end-users to set up reservations for VCs in advance, with NSI and GUI interfaces.

Once a reservation is made, a VC provisioning step can be instantiated either by the NSI provisioning request from a user. Network device specific GUNI messages for each bender are used to initiate RSVP signaling, and MPLS LSP provisioning and release on the network devices. LSPs are established based on Forward Equivalent Class (FEC) for VC provisioning between two storage hosts. For both layer 2 and layer 3 VCs, where reservations are bidirectional, the configuration GUNI messages are delivered to both edge routers at the start and end of the intra domain VC. 2 backup VCs (i.e., 1 bidirectional VC) in secondary links are internally provisioned for protection, in addition to 2 primary VCs (i.e., 1 bidirectional VC) provisioned in primary links by a user request.

D. VC Reservation Request by DynamicKL

A user can select source/destination nodes on the topology map and provide source/destination host addresses, which is internally mapped to Service Termination Point (STP) addresses. If a user provides a starting time and an end time of reservation and inquires about a residual bandwidth, a maximum available bandwidth from a source node to a destination node is shown to a user. If a user provides a bandwidth smaller than that, a reservation request is ended. So, users do not have to experience reservation failures when searching for a needed specific BW. 2 backup VCs (i.e., 1 bi-directional VC) in secondary links are internally reserved for protection by DynamicKL, in addition to 2 primary VCs (i.e., 1 bi-directional VC) in primary links by a user request.

 ISBN: 978-1-61208-305-6

Since users can notice primary link or node failure in a topology map, they can reserve VCs on the rest of nodes and primary links, except failure nodes or links.

E. Monitoring

An administrator can monitor network events and all VCs and reservations at all sites in a dynamic VC network, while a user can monitor his/her VCs or reservations only. Thus, an administrator has an authorization to terminate reservation abuses in all sites, in case of unexpected events, such as router or interface failures or suspected reservation abuses. An administrator can request a network operator in a dynamic VC network to recover physical network failures and VC failures due to unsuccessful protection or unexpected events.

IV. PROTECTION MANAGEMENT PER VC

Since NSI standardization does not yet address network management issues, such as protection management, we have implemented the RICE, especially for protection management for VCs and reservations with a link failure.

A backup virtual circuit for protection is internally reserved and provisioned by DynamicKL, together with a primary reservation and virtual circuit for a user request. An active path is switched from a primary VC to a backup VC in a dynamic VC service network, in the event of a primary link failure (i.e., a 1:1 path based protection [17] implemented in network devices is used). A protection management function per VC is addressed to provide manageable and reliable VC services, by using RICE.

A. RICE API messages for network event management

1) InterfaceDown

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 [18]. An *InterfaceDown* message is used to notify a web portal server a fault interface of a network device.

2) InterfaceUp

When a failure interface is repaired, an SNMP trap message from a network device with a fault interface is delivered to the GUMH. An *InterfaceUP* message is used to notify a web portal server a retrieved interface of a network device.

3) NodeDown

To monitor a network device with a fault, a periodic polling message from the GUMH is delivered to network devices. A *NodeDown* message is used to notify a web portal server a fault of a network device.

4) NodeUp

When a network device with a fault is retrieved, it can be monitored by using both an SNMP trap and polling messages [18]. A *NodeUp* message is used to notify a web portal server a retrieved network device.

5) Primary2SecondarySuccess/Primary2SecondaryFail and Secondary2PrimarySuccess/Secondary2PrimaryFail

Primary2SecondarySuccess and *Primary2SecondaryFail* API messages are used to notify that backup VCs pre-assigned in secondary links currently operate as working paths to protect VCs in a failure primary link and at least a backup VC does not operate as a working path, respectively. On the other hand, *Secondary2PrimarySuccess* and *Secondary2PrimaryFail* messages are used to provide notification that all of VCs in a repaired primary link (interfaces) are retrieved as active paths, and to indicate that at least a VC is not currently retrieved as an active path, respectively. The above events can be detected by sending Query_VC GUNI messages to network devices, after receiving an SNMP trap message with primary link failure or repair information from a network device.

B. A protection management service scenario

RICE and GUNI message flows for a protection management service scenario are shown in Fig. 2. Internal message flows between the PCRA, the CEMH and GUMH are also shown. It is assumed that a failure primary link has provisioned VCs beforehand. A SNMP trap message from a network device with a failure link is received at the GUMH. The GUMH notifies the CEMH failure link information and the CEMH requests the PCRA to renew network topology and primary reservation table information. The CEMH sends an *InterfaceDownRequest* message to the web portal server for a notification of a failure link and the web server renews network topology and primary reservation DBs. An *InterfaceDownConf* message is received at the CEMH. Because provisioned VCs in a failure primary link have also failures, the CEMH requests the GUMH to inquire about status information of backup VCs. The GUMH detects status information of backup VCs by using a *Query_VC* GUNI message. If all backup VCs pre-assigned are working on active paths, the GUMH notifies the CEMH a protection success. The CEMH requests the PCRA to renew a backup

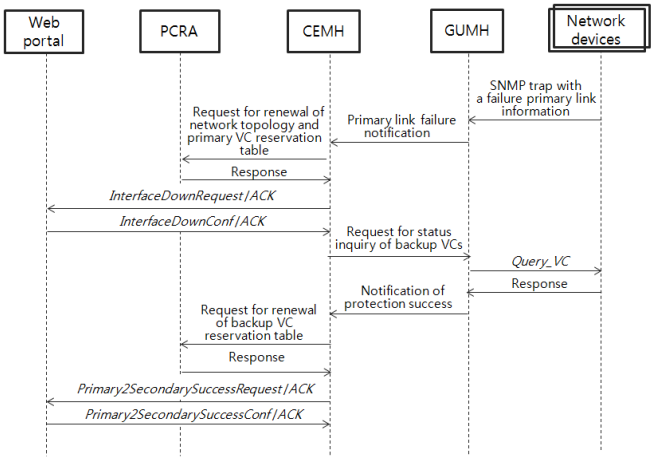


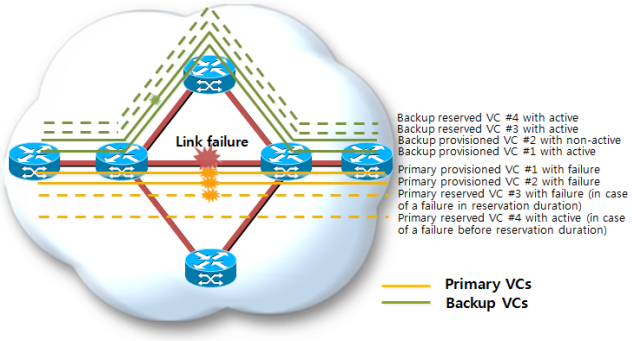
Fig. 2. RICE and GUNI message flows for one protection management service scenario.

VC reservation table. The CEMH finally sends a *Primary2SecondarySuccessRequest* message to the web portal server for a notification of a protection success. A backup VC reservation DB changes status of secondary VCs to active status. Then the web portal server sends a *Primary2SecondarySuccessConf* message to the CEMH.

An administrator can identify each network event by receiving a distinguished RICE message from NSA.

C. Advantages of Protection Management per VC

By using the RICE API messages, an administrator can observe that backup VCs are currently working as active paths or not working in the event of a primary link failure, and primary VCs are automatically retrieved or not retrieved after a primary link repair. With these messages, failure and protection status information is managed per VC, as well as per network link. In other words, NSA notifies successful or unsuccessful VC activations as working paths in the backup link and successful or unsuccessful VC retrievals as working paths after a primary link repair, and failure and protection status information per VC (i.e., per CID) in a primary and backup reservation DB is separately managed, as shown in Fig. 3. If protection status information is not provided per VC, an administrator will assume that all backup VCs are successfully operating as working paths in the event of a failure primary link with VCs and that all primary VCs operates successfully as working paths in the event of a primary link repair. We also note that a provisioned VC may not operate as an active path, due to unexpected events, such as a network device's configuration intervention by an individual and an OS problem in a network device. In addition, a provisioned VC may be released by an administrator's mistake.



<Protection management per VC in primary/backup reservation DBs>

VC	Primary VC status	Backup VC status
VC #1	Provisioned with failure	Provisioned with non-active
VC #2	Provisioned with failure	Provisioned with active
VC #3	Reserved with failure	Reserved with active
VC #4	Reserved with active	Reserved with active

Fig. 3 Protection management per VC in a primary and backup reservation DB in the event of a primary link failure with VCs and reservations.

Because primary and backup reservation DBs separately manage failure and protection status information per each primary and backup VC, as shown in Fig 3, DynamicKL can still release and terminate user virtual circuits in backup links, in the event of a primary link failure with VCs. Also, it is possible to establish VCs and to terminate reservations in backup links for reservations with a primary link failure, by delivering NSI provisioning messages with backup interfaces information to network devices. Users wanting to reserve VCs can monitor a failure link or a failure node and create reservations for the rest of links and nodes.

D. Comparison of Advance Reservation Frameworks

Table II provides a comparison of advance reservation frameworks. G-Lambda A/K, OSCARS, AutoBHAN, and OpenDRAC have been implementing the standard NSI interface in their frameworks. Few of the works in literatures have been introduced [3][11][19]. As an additional function compared to other advance reservation frameworks, DynamicKL provides a protection management function per VC, for protection from primary VCs to disjointed VCs and its specific management, in the event of a failure link.

Protection management in an advance reservation framework for VC services should be provided per VC, in protecting provisioned VCs and reservations with a link failure. That is, an advance reservation framework should be able to detect successful or unsuccessful VC activation information in backup links and successful or unsuccessful VC retrieval information in primary links. With this information, it is possible to renew and manage a primary and backup reservation DB with failure and protection status information per VC (CID), which leads to a protection management per VC. These capabilities in the DynamicKL result in more manageable and reliable VC service.

V. Demonstration of Protection Management per VC

A. Service Network Architecture

To address the growing need for guaranteed bandwidth by large-scale collaborations, such as the LHC in the field of HEP, the KREONET has designed and implemented the dynamic virtual circuit network, which is physically distinct from the IP core network (KREONET). NSI VC services (i.e., reservation, provision, release, termination, and inquiry services) can be made on the dynamic VC service network, which consists of some part of 5 sites in the KREONET, as shown in Fig. 4. The KREONET core network is architected primarily to transport IP packets as a best effort service, while the KREONET dynamic virtual circuit network is engineered to support only dynamic virtual circuits (VCs). The NSA system implemented as web service operates as a server to process request messages and operates as a client for the creation of confirmation messages. The web portal

 ISBN: 978-1-61208-305-6

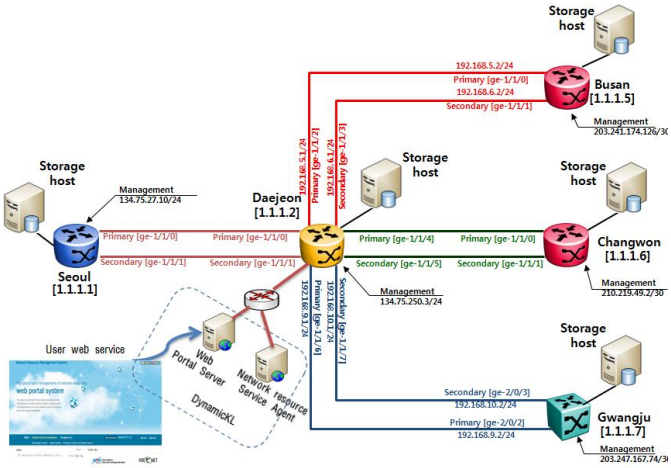


Fig. 4. Dynamic VC service network by the DynamicKL with a protection management per VC.

server also operates as a client for request messages and a server for confirmation messages.

Even though a secondary (backup) and primary VC in the dynamic VC network do not have disjointed paths in the event of a node failure, due to the star topology of KREONET core network, they have disjointed paths each other in the event of a link failure. A demonstration for protection management per VC is focused on the event of a link failure.

B. Demonstration of Protection Management per VC

In this subsection, a demonstration of protection management per VC for VCs with a link failure is demonstrated in the dynamic VC service network. Bidirectional VC (2 unidirectional VCs) with 100 *Mbps* BW on a designated network path from a host at Gwangju to a host at Pusan is provisioned by a user request. A primary link at Daejeon site has a failure event. A protection management GUI for an administrator is shown in Fig. 5. An administrator can notice that a primary link at Daejeon site has failure, by receiving an *InterfaceDown* message. Also an administrator

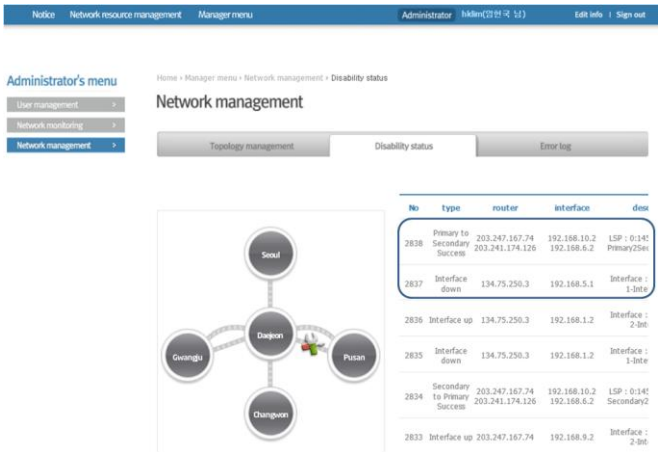


Fig. 5. Protection management GUI for an administrator.

can recognize that backup VCs in secondary links are successfully working as active paths to protect primary VCs with a link failure, by receiving a *Primary2SecondarySuccess* API message. A primary reservation DB creates failure information for primary VCs and a backup reservation DB changes status information for backup VCs from non-active (standby) status to active status. By making use of these API messages, protection management per VC is possible in a primary and backup reservation DB. Figure 6 verifies that two secondary VCs pre-assigned are working as active paths, by exchanging signaling messages between network devices after a primary link failure.

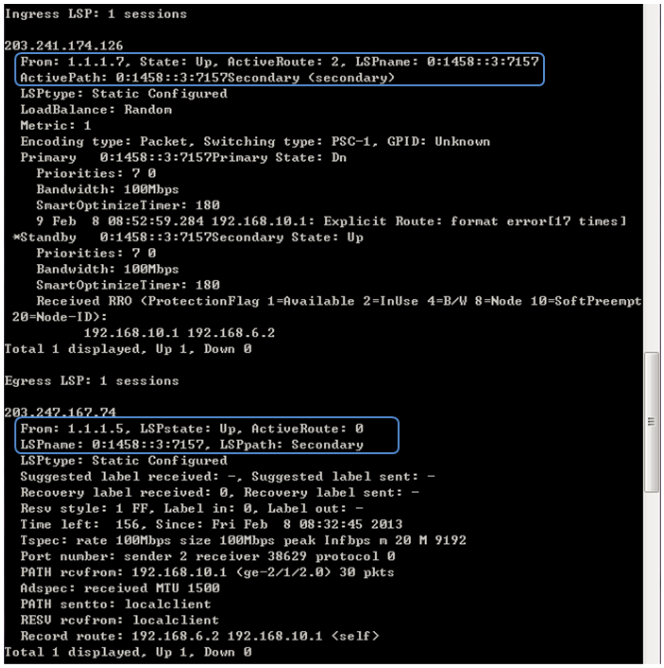


Fig. 6. Secondary VCs working on active paths in the event of a primary link failure.

When a failure primary link at Daejeon site is repaired by a network operator, a management GUI for an administrator is shown in Fig. 7. An administrator can recognize that a failure primary link was repaired by receiving an *InterfaceUp* message. Also, an administrator can notice that VCs in primary links are successfully retrieved as active (working) paths, by receiving a *Secondary2PrimarySuccess* API message. A primary reservation DB changes status information of provisioned VCs with failures to active status and a backup reservation DB renews backup VCs with active status to them with non-active (standby) status. These RICE API messages demonstrate that a retrieval management per VC is possible in DynamicKL.

VI. CONCLUSIONS

To date the majority of approaches in advance reservation frameworks do not address a number of required management issues, such as fault and protection managements for VC services, to provide manageability and

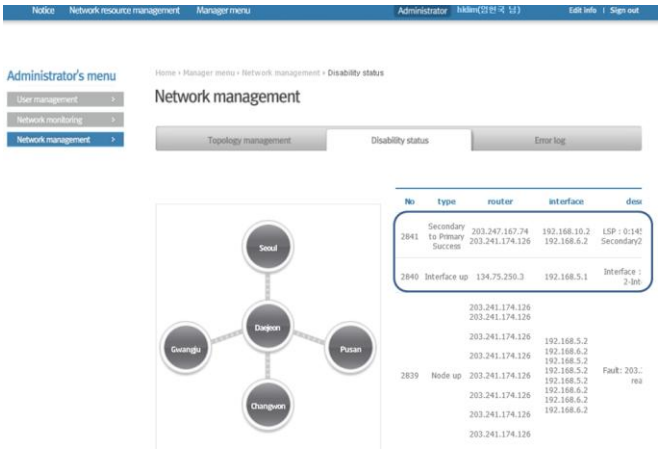


Fig. 7. VC retrieval management GUI for an administrator.

reliability guarantees. Here, we demonstrated that the DynamicKL provides the protection management per VC by using RICE interface, for virtual circuits and reservations with a primary link failure, which leads to more manageable and reliable VC services. Also, DynamicKL provides reservation, provision, release, termination, and inquiry services for virtual circuits by using a standard network service interface (NSI). With the protection management capability, an administrator can notice successful or unsuccessful VC protections in the event of a primary link failure and successful or unsuccessful primary VC retrievals as active paths after a primary link repair. Because a primary and a backup reservation DB separately manage failure and protection status information per each primary and backup VC, DynamicKL is able to release and terminate user virtual circuits in backup links, in the event of a primary link failure with VCs. In conclusion, DynamicKL could contribute to improve manageability and reliability of VC services.

REFERENCES

[1] T. Lehman, J. Sobieski, and B. Jabbari, "DRAGON: A Framework for Service Provisioning in Heterogeneous Grid Network," IEEE Communi., Magazine, Vol. 44, Issue 3, Mar. 2006, pp. 84-90.

[2] C. P. Guok, D. W. Robertson, E. Chaniotakis, M. R. Thompson, W. Johnston, and B. Tierney, "A User Driven Dynamic Circuit Network Implementation", IEEE Globe Communi., Oct. 2009, pp. 1-6.

[3] J. Lukasik, O. Neofytou, A. Sevasti, S. Thomas, and S. Tyley, "Installation and Deployment Guide: AutoBAHN System Book", Published by DANTE, June 2008.

[4] F. Travostino, R. Keates, T. Lavian, I. Monga, and B. Schofield, "Project DRAC: Creating an Application-aware Network", Nortel Journal, vol. 22 No. 8, Oct. 2006, pp. 23-26.

[5] G. Zervas, et al., "Phosphorus grid-enabled GMPLS control plane (GMPLS): architectures, services, and interfaces," IEEE Communication Magazine, vol. 46, no. 6, Jun. 2008, pp. 128 –137.

[6] L. Battestilli, et al., "EnLIGHTened computing: An architecture for co-allocating network, compute, and other Grid resources for high-end applications," International Symposium on High Capacity Optical Networks and Enabling Technologies, Nov. 2007, pp. 1–8.

[7] A. Takefusa, et al., "G-Lambda: Coordination of a grid scheduler and lambda path service over GMPLS", Future Generation Computing Systems, vol. 22 No. 8, Oct. 2006, pp. 868-875.

[8] G. Roberts, T. Kudoh, I. Monga, J. Sobieski, and J. Vollbrecht, "Network Service Framework v1.0", Technical Report GFD 173, OGF NSI-WG, 2010.

[9] J. Sobieski, "GLIF 2011 Rio NSI PlugFest Guide and Interoperability Challenge", OGF NSI WG, 2011.

[10] G. Roberts, T. Kudoh, I. Monga, and J. Sobieski, "NSI Connection Service Protocol v1.1," GFD-173, OGF NSI-WG, 2011.

[11] R. Krzywania, et al., Network Service Interface: Gateway for Future Network Services, Terena Network Conference, Jun. 2012, pp. 1-15.

[12] I. W. Harbib, Q. Song, Z. Li, and N. S. V. Rao, 'Deployment of the GMPLS Control Plane for Grid Applicaitons in Experimental High-Performance Networks', *IEEE Communications Magazine*, Vol. 44, Issue 3, March 2006, pp. 65-73.

[13] 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, May 2011, pp. 132 –139

[14] Handling Dynamiche Lichtpaden Manual, Version 0.2, Published by SURFnet, Nov. 2008.

[15] Grid Network Service-Web Services Interface (GNS-WSI), version 3, 2008, "https://www.g-lambda.net" [retrieved: Sep. 2013].

[16] Y. Cha, K. Lee, C. Kim, J. Kong, J. Moon, and H. Lim," Grid Network Management System Based on Hierarchical Information Model", Communications in Computer and Information Science, 1, Vol. 206, Part 4, Sep. 2011, pp. 249-258.

[17] R. Hughes-Jones, Y. Xin, G. Karmous-Edwards, and J. Strand, "Network Performance Monitoring, Fault Detection, Recovery, and Restoration", Grid Networks, Editors: F. Travostino, J. Mambretti and G. Karmous-Edwards, Wiley, pp. 253-275.

[18] K. Ogaki, M. Miyazawa, T. Otani, and H. Tanaka, "Prototype demonstration of integrating MPLS/GMPLS network operation and management system", OFC 2006, Mar. 2006, pp. 1-8.

[19] Z. Zhao, et al., "Planning data intensive workflows on inter-domain resources using the Network Service Interface (NSI)," 2012 SC Companion: High Performance Computing, Networking Storage and Analysis, Nov. 2012, pp. 150-156.

 ISBN: 978-1-61208-305-6

TABLE II. COMPARISON OF ADVANCE RESERVATION FRAMEWORKS

Framework	Provisioning Layer	Network Resource Provisioning System	Grid Co-scheduling Capabilities	Protection management per VC	With NSI
DynamicKL	Layer 2 & 3	Integrated (MPLS/VLAN)	No	Yes	Yes
OSCARS	Layer 2 & 3	Integrated (MPLS/VLAN)	No	No	IDC/NSI
OpenDRAC	Layer 1 & 2	Integrated	No	No (protection only)	Yes
EnLIGHTened	Layer 1 & 2	Integrated (GMPLS based)	Yes	No	No
G-Lambda A/K	Layer 1 & 2	Integrated (GMPLS based)	Yes	No	Yes
PHOSPHORUS	Layer 1 & 2	ARGON, DRAC and UCLP	Yes	No	No
AutoBAHN	Layer 2 & 3	Integrated	No	No	IDC/NSI
DRAGON	Layer 1 & 2	VLSR (GMPLS based)	No	No	No

 ISBN: 978-1-61208-305-6