

## 대용량 데이터 전송을 위한 오픈 API 기반 가상 네트워크 프로비저닝 자동화 플랫폼 구현

김용환<sup>1</sup> · 박성진<sup>2</sup> · 김동균<sup>3\*</sup>

### Implementation of an open API-based virtual network provisioning automation platform for large-scale data transfer

Yong-hwan Kim<sup>1</sup> · Seongjin Park<sup>2</sup> · Dongkyun Kim<sup>3\*</sup>

<sup>1</sup>Senior Researcher, KISTI, Yuseong-gu, Daejeon, 34141 Korea

<sup>2</sup>Postdoctoral Researcher, KISTI, Yuseong-gu, Daejeon, 34141 Korea

<sup>3\*</sup>Principal Researcher, KISTI, Yuseong-gu, Daejeon, 34141 Korea

#### 요약

현재 선도국가연구망 그룹에서 SDN/NFV 기반의 네트워크 자동화 지능화 기술을 사용자에게 서비스 형태로 제공하기 위한 요구가 지속적으로 발생함에 따라 관련 연구개발이 추진되고 있다. 또한, 선도국가연구망에서는 일반 네트워크 환경보다 대용량 데이터 전송을 고성능으로 제공해야 한다는 요구사항이 점차 더 커지고 있는 실정이다. 이에 따라, 본 논문에서는 이러한 국가연구망의 네트워킹 요구사항들에 대응하기 위하여 연구 개발한 대용량 데이터 전송을 위한 오픈 API 기반 가상 네트워크 프로비저닝 자동화 플랫폼을 제안하고 이의 구현 결과를 제시한다. 해당 플랫폼에는 SDN 기반의 네트워크 가상화 기술을 제공하는 KREONET-S의 VDN 시스템, 컨테이너 중심의 서버 가상화 기술을 제공하는 Kubernetes 시스템, 그리고 고성능 데이터 전송 시스템인 Globus Online이 포함되어 있다. 본 논문에서는 이러한 상이한 시스템 간의 연계를 위한 환경 설정 및 시스템 연동 결과, 가상 네트워크 프로비저닝 자동화 구현 결과 및 성능에 대하여 보인다.

#### ABSTRACT

Currently, advanced national research network groups are continuously conducting R&D for the requirement to provide SDN/NFV-based network automation and intelligence technology for R&E users. In addition, the requirement for providing large-scale data transmission with the high performance networking facility, compared to general network environments, is gradually increasing in the advanced national research networks. Accordingly, in this paper, we propose an open API-based virtual network provisioning automation platform for large data transmission researched and developed to respond to the networking requirements of the national research network and present the implementation results. The platform includes the KREONET-S VDN system that provides SDN-based network virtualization technology, and the Kubernetes system that provides container-oriented server virtualization technology, and the Globus Online, a high-performance data transmission system. In this paper, the environment configurations, the system implementation results for the interworking between the heterogeneous systems, and the automated virtual network provisioning implementation results are presented.

**키워드** : 소프트웨어정의네트워킹, 가상화, 오픈 API, 자동화, 자원 할당

**Keywords** : SDN, Virtualization, Open API, Automation, Resource Allocation

Received 10 August 2022, Revised 25 August 2022, Accepted 29 August 2022

\* Corresponding Author Dongkyun Kim(E-mail:mirr@kisti.re.kr, Tel:+82-42-869-0516)

Principal Researcher, KISTI, Yuseong-gu, Daejeon, 34141, Korea

Open Access <http://doi.org/10.6109/jkiice.2022.26.9.1320>

print ISSN: 2234-4772 online ISSN: 2288-4165

© This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License(<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.  
Copyright © The Korea Institute of Information and Communication Engineering.

## I. 서 론

KREONET-S (KREONET Softwarization)[1]는 국가 과학기술연구망인 KREONET (Korea Research Environment Open Network)[2]의 소프트웨어화를 위하여 출범한 이니셔티브이다. KREONET-S는 SDN의 기본 구조인 데이터 전달 계층(Data Plane), 제어 계층(Control Plane), 응용 계층(가상화 응용 및 서비스 계층)을 토대로 설계되었으며, 새로운 네트워크 서비스에 대한 첨단응용 사용자들의 다양한 요구사항을 반영하기 위하여 하드웨어 기반의 폐쇄적인 제약에서 탈피하고 다양한 네트워크 환경의 변화에 대응할 수 있는 소프트웨어 기반의 네트워크 환경을 구축하고 있다. 그리고 이를 통하여 KREONET의 첨단연구 및 응용 기관을 대동한 새로운 네트워킹 인프라와 서비스 및 사용자 경험을 제공하고자 한다[3].

KREONET-S의 데이터 전달 계층을 담당하는 네트워크 장비는 SDN 표준 프로토콜인 오픈플로우를 탑재한 코어 및 에지 스위치로써, 국내외 8개 지역에 설치되어 SDN 광역망을 구축 운영 중이다. 제어계층의 경우 오픈소스 SDN 제어 플랫폼인 ONOS (Open Network Operating System)[4, 5] 컨트롤러 중심의 클러스터링 환경으로 동작하여 고가용성과 확장성을 확보할 수 있다. 그리고 ONOS의 노스바운드 API를 기반으로 개발된 가상전용네트워크(VDN, Virtually Dedicated Network) 시스템[6, 7]을 중심으로 KREONET-S의 가상화 응용 및 서비스 계층을 구성한다. VDN 시스템은 연구자들이 필요로 하는 고성능의 네트워크를 빠른 시간 내 프로비저닝함으로써 대용량 연구 데이터 전송과 관리를 요구하는 선도적 첨단협업연구를 적시에 지원하고, SDN 기반의 동적·온디맨드 네트워크 슬라이싱 서비스를 제공하고자 연구 및 개발되었다.

특히, 선도국가연구망 커뮤니티의 경우, SDN/NFV 기반의 네트워크 자동화·지능화 기술이 네트워크 분야의 중점 개발 동향 중 하나로 연구망의 과학기술자, 연구자, 실험자들이 요구하는 고성능 액세스망 연계형 에지/이용자 ScienceDMZ[8, 9] 환경과 자동화·지능화된 새로운 형태의 전용 네트워킹 서비스 개발에 대한 요구가 증대되는 실정이다[10, 11]. 여기서 ScienceDMZ는 대용량 데이터 중심형 거대과학(Data-Intensive Science) 및 관련 협업연구를 수행하기 위해 성능의 저하 없이 대

용량 데이터 전송이 가능한 혁신적인 네트워크 구조로 연구·교육 분야의 대용량 데이터를 고속으로 전송하기 위해 기존 네트워크 환경에서 연구 및 범용 네트워크로 트래픽 특성별 망 분리를 수행한다.

본 논문에서는 KREONET-S 기반의 새로운 네트워킹 서비스 요구사항들에 대응하기 위하여 SDN 기반의 네트워크 가상화 기술과 컨테이너 중심의 서버 가상화 기술을 연계 통합하고, 향후 KREONET-S를 자동화된 인프라로 고도화 추진하는 한편, 지능화된 차세대 자율 네트워킹 환경을 조성하기 위하여 개방형 응용 프로그래밍 인터페이스(API, Application Programming Interface)를 활용한 이종의 다중 시스템 연계 자동화·지능화 플랫폼 개발 결과에 대하여 소개한다. 이는 VDN 시스템을 통하여 첨단연구 및 응용사용자가 필요로 하는 고정적인 네트워크 자원을 빠른 속도로 제공하는데 그치지 않고, 상황에 따라 사용자의 가상전용네트워크 자원 수요를 측정 및 예측하여 요구되는 네트워크 자원을 동적으로 제공하는 자동화·지능화된 시스템을 포함한다. 이를 위하여 네트워크 가상화를 제공하는 KREONET-S의 VDN 시스템, 컨테이너 형태의 서버 가상화를 제공하는 Kubernetes[12, 13] 시스템, ScienceDMZ의 고성능 데이터 전송 시스템인 Globus Online[14]을 각각의 시스템에서 제공하는 오픈 API 기반으로 연계 및 구현했다. Kubernetes는 컨테이너 기반 서버 가상화 기술을 제공하며 컨테이너화된 애플리케이션을 자동으로 배포, 스케일링 및 관리해주는 오픈소스 시스템이다. Globus Online은 협업 연구기관 간 대용량 데이터의 고속 전송 기능을 SaaS(Software as a Service) 방식으로 제공하는 소프트웨어 기반 클라우드 플랫폼 서비스로 핵심 기능인 데이터 전송 기능 외에 협업 연구자들 간의 데이터 공유, 관리 및 포탈 연계 가능한 사용자 맞춤형 응용 개발 환경을 지원한다.

본 논문의 구성은 다음과 같다. 2장에서 Kubernetes 컨테이너 이미지 자동화와 VDN 및 Globus Online의 연계 환경 설정과 시스템 연동 방안에 대하여 소개하고, 3장에서는 오픈 API를 연계 활용하기 위한 사용자 인증 절차를 제시한다. 그리고 4장에서 대용량 데이터 전송을 위한 오픈 API 기반 가상 네트워크 프로비저닝 자동화 플랫폼 구현 결과와 성능시험 결과를 보인다. 마지막으로 5장에서 본 논문의 결론을 맺는다.



Fig. 1 Open API-based virtual network provisioning automation platform

## II. 환경 설정 및 시스템 연동

본 논문은 대용량의 데이터 송수신을 수행하는 데이터 전송 노드(DTN, Data Transfer Node)를 위한 가상 네트워크 프로비저닝 자동화 플랫폼의 구현 결과를 보인다. 여기서 DTN은 고품질의 하드웨어 장비들을 조합한 리소스들과 광역 데이터 전송을 위해 BIOS, OS, 소프트웨어의 튜닝을 통해 구성된 리눅스 서버이다. 제안하는 플랫폼의 전반적인 연계 구조도는 그림1에서 볼 수 있다. 네트워크 가상화를 제공하는 KREONET-S의 VDN 시스템, 컨테이너 형태의 서버 가상화를 제공하는 Kubernetes 시스템, ScienceDMZ의 고성능 데이터 전송 시스템인 Globus Online을 오픈 API로 연계하여 구현하였다. 본 논문에서 제안하는 DTN 간의 VDN (중단간 네트워크 슬라이싱 자동화 기술을 통하여 관리자나 사용자의 개입 없이 시스템 간의 오픈 API를 교환하고, 지정된 이벤트 처리만으로 VDN을 쉽고 빠르게 자동 생성할 수 있으며 KREONET-S 인프라의 대역폭 등 네트워크 자원을 보다 효율적으로 활용할 수 있다.

한편, DTN은 KREONET-S 인프라에 사전에 연결된 물리 서버 혹은 사용자 요구에 따라 동적으로 생성 및 관리되는 가상 머신(VM, Virtual Machine)이나 컨테이

너 형태로 구축이 가능하다. 서비스 자원의 효율화 및 시스템 단위의 연계를 통한 플랫폼 자동화 관점에서 DTN은 점차 컨테이너 DTN 형태로 발전할 필요가 있으며 이를 위하여 Kubernetes 기반의 DTN 컨테이너 이미지 생성 및 Globus Online과 자동화된 연계 등록이 요구된다. 또한, 생성된 DTN 컨테이너는 ONOS 기반의 VDN 시스템이 인식할 수 있어야 VDN을 통한 가상 네트워크 프로비저닝이 가능하게 된다.

Globus Online을 KREONET-S에서 설치한 DTN 서버들과 연계하여 자동화를 수행하기 위해서는 먼저 Docker[15] 이미지를 구성해야 한다. Docker 이미지를 이용하여 Globus Online 등록을 자동화하는 이유는 본 논문에서 구현하는 플랫폼에서 Globus Online에 등록될 DTN 서버가 Docker 컨테이너를 오케스트레이션하는 Kubernetes로 구성되기 때문이다. Docker를 이용하여 생성된 Globus Online의 DTN 이미지는 Kubernetes를 통해 컨테이너(POD) 형태로 생성되며, 전송 시스템인 Globus Online에 자동 등록된다. Docker에서 이미지를 생성하기 위하여 “docker build” 명령어를 이용할 수 있으며, 표 1은 Globus Online DTN을 생성하는데 사용된 Dockerfile이다.

**Table. 1** Dockerfile to create Globus Online DTN

```
FROM centos:7

RUN yum update -y
RUN yum install epel-release openssh-server polkit net-tools
at -y
RUN yum groupinstall -y "Development Tools"
ADD useradd.sh useradd.sh
RUN ./useradd.sh
RUN curl -LO https://downloads.globus.org/toolkit/globus
-connect-server/globus-connect-server-latest.noarch.rpm
RUN yum install -y yum-plugin-priorities
RUN yum install globus-connect-server-repo-latest.noarch.rpm
-y

ADD globus-connect-server.conf /etc/globus-connect-server.conf
RUN yum install -y globus-connect-server
ADD setup.sh setup.sh
RUN ./setup.sh
```

**Table. 2** YAML file for Kubernetes container creation

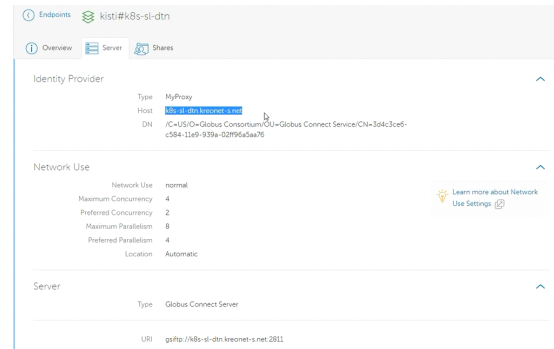
```
apiVersion: v1
kind: Pod
metadata:
  name: k8s-sl-dtn.kreonet-s.net
  labels:
    app: k8s-sl-dtn-app
  annotations:
    cni-plugins: '[ "host-device" ]'
spec:
  hostIPC: true
  containers:
    - name: k8s-sl-dtn-container
      image: globusonline-install:v1
      securityContext:
        privileged: true
        command: ["/sbin/init"]
      restartPolicy: Never
      dnsPolicy: None
      dnsConfig:
        nameservers:
          - 8.8.8.8
```

표 1의 이미지를 이용하여 Kubernetes에서 DTN 컨테이너를 생성하기 위해서는 Kubernetes에서 구성 파일 작성에 자주 사용되는 데이터 직렬화 언어인 YAML 파일을 구성해야 한다. 표 2는 Kubernetes에서 DTN 컨테이너를 생성하기 위한 YAML 파일이다. 해당 파일을 이용하여 컨테이너를 생성하면 생성된 컨테이너의 Hostname은 k8s-sl-dtn.kreonet-s.net으로 설정이 되고,

globusonline-install 이미지를 이용하여 Globus online 이 설치된 컨테이너가 생성된 후, Globus Online에 등록이 되는 것을 확인할 수 있다. 그림 2는 Kubernetes에 생성된 컨테이너 DTN 정보를 Kubernetes의 CLI인 kubectl로 확인한 결과이다. 해당 컨테이너의 Hostname이 k8s-sl-dtn.kreonet-s.net로 설정됨을 확인할 수 있다. 또한 그림3에서는 해당 DTN이 k8s-sl-dtn.kreonet-s.net로 Globus Online에 등록됨을 확인할 수 있다.

```
kreonet-s@sl-node:~$ kubectl get pod --all-namespaces
NAMESPACE   NAME                                     READY   STATUS    RESTARTS
AGE
default      k8s-sl-dtn.kreonet-s.net               1/1     Running  0
17d
kube-system  etcd-sl-master                         1/1     Running  3
290d
kube-system  kube-apiserver-sl-master               1/1     Running  3
290d
kube-system  kube-controller-manager-sl-master     1/1     Running  3
290d
kube-system  kube-proxy-7lds4                       1/1     Running  3
290d
kube-system  kube-proxy-qjz9                        1/1     Running  3
290d
kube-system  kube-scheduler-sl-master              1/1     Running  3
290d
kreonet-s@sl-node:~$ kubectl exec -it k8s-sl-dtn.kreonet-s.net -- /bin/bash
[root@k8s-sl-dtn /]#
```

**Fig. 2** Verification of container DTN registration in Kubernetes



**Fig. 3** Container DTN registration verification in Globus Online

Globus Online 시스템은 DTN 등의 서버를 등록할 때 DNS 주소를 이용하여 데이터 전송을 수행할 각 중단 지점을 연결한다. 이에 따라, Kubernetes에서 생성한 컨테이너 형태의 DTN을 등록할 때도 Hostname을 k8s-sl-dtn.kreonet-s.net과 같이 URL 주소로 설정해야 한다. 그리고 Globus Online 시스템에서 DTN 등의 실제 서버에 접속하기 위해서는 DNS 주소를 이용하여 서버에 접속해야 한다. 따라서 Globus Online 시스템에 등록되는 데이터 전송의 중단 지점들은 DNS 주소가 필수적으로 요구된다. 이를 위해 KREONET-S에서 Globus Online 시

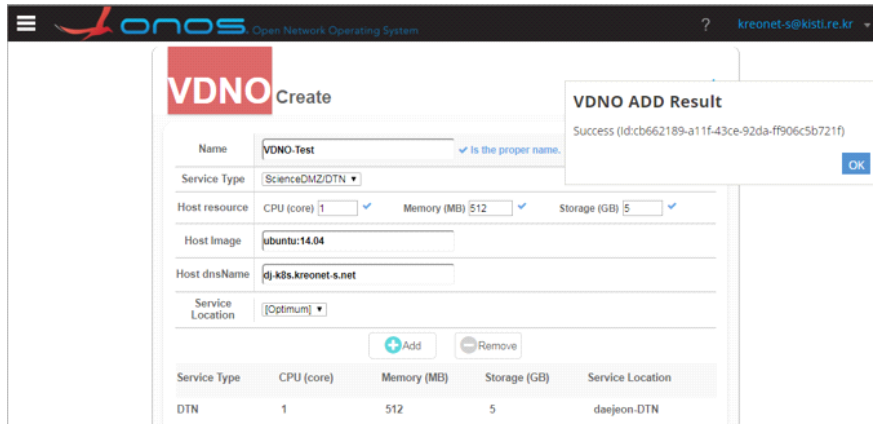


Fig. 4 Container DTN settings screen in KREONET-S/VDN

스택과 연계된 DTN 컨테이너를 생성 및 관리하기 위하여 별도의 DNS 서버를 구성하여 활용하였다. 그리고 KREONET-S/VDN 환경에서 각각의 컨테이너 DTN을 생성하기 위하여 Kubernetes 오픈 API 호출을 수행할 때, 그림 4와 같이 컨테이너 요구 자원 정보와 함께 Hostname을 입력받아 Kubernetes에게 요청하면 Kubernetes는 표 2와 같은 YAML 파일을 생성하고 이를 활용하여 Globus Online 시스템과 연계된 컨테이너 DTN을 생성한다. 그리고 이렇게 생성된 DTN은 KREONET-S/VDN에서 호스트 형태로 인지되어 4장에서 제안되는 프로비저닝 자동화를 수행할 수 있게 된다. 이에 대한 세부 내용은 4장에서 다룬다.

### III. Globus Online 애플리케이션 등록 및 오픈 API 인증

Globus Online 시스템은 오픈 API로 “Auth API”와 “Transfer API”를 제공한다. Auth는 Globus Online 시스템의 권한 부여 및 접근에 관한 API를 의미하며, Transfer는 DTN의 등록 및 해제, 리스트 등의 DTN 관리 및 데이터 전송에 관련된 API이다. 본 장에서는 Globus Online의 오픈 API를 사용하기 위한 환경 설정과 오픈 API를 활용한 애플리케이션 등록 및 구현 결과에 대하여 서술한다.

Python 형태로 제공되는 Globus Online의 Auth와 Transfer API를 사용하기 위해서는 먼저 API를 호출하

는 서버 환경에 Globus Online을 설치 및 설정하는 과정이 요구된다. 또한 Globus Online 상에 별도의 애플리케이션 등록 및 인증 과정을 거쳐야 한다. 그리고 이때, Globus Online은 OAuth2[16]에 따라 해당 과정을 수행한다. OAuth2는 인터넷 사용자들이 비밀번호를 제공하지 않고 다른 웹사이트 상의 자신들의 정보에 대해 웹사이트나 애플리케이션의 접근 권한을 부여할 수 있는 개방형 표준이다. 해당 과정을 수행하기 위해서는 Globus Online의 가입 계정이 요구되며, 이에 따른 시스템 인증이 필요하다. 해당 과정이 완료되면 오픈 API를 호출할 애플리케이션을 Globus Online에 등록해야 하며, API 사용 범위 및 보안 정책 등 다양한 옵션을 지정할 수 있다. 그림 5와 그림 6은 각각 애플리케이션 등록 화면과 이의 등록 결과를 보인다.

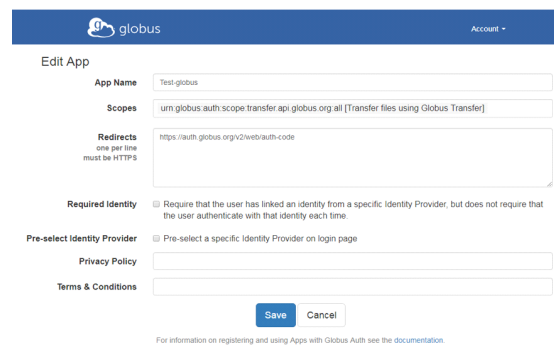


Fig. 5 Application registration in Globus Online



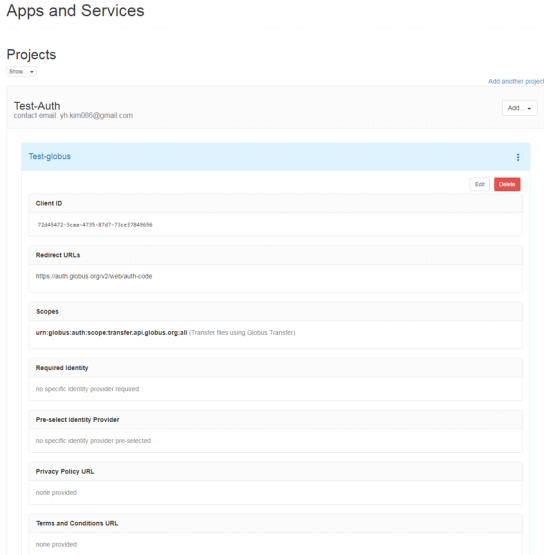


Fig. 6 Verification of application registration in Globus Online

Globus Online API를 사용하는 애플리케이션은 그림 6에서 확인 가능한 ClientID와 RedirectURLs를 활용하여 Globus Online 인증 절차를 수행해야 한다. 그림 7의 vdn-globus.py 또한 ClientID와 RedirectURLs를 활용한 인증 절차를 Auth API 호출을 통하여 수행한다. 해당 과정이 완료되면 그림 7과 같이 인증 URL이 자동 생성되며 이를 통한 애플리케이션 권한 부여 절차가 진행된다. 이를 위하여 그림 8과 같이 인증 절차가 끝나면 자동으로 사용자에게 팝업되는 URL 페이지에서 권한 부여 코드를 획득해야 한다.

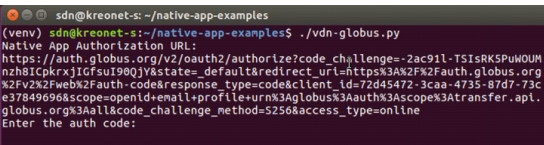


Fig. 7 Execution of Globus Online application and authentication

권한 부여 절차는 애플리케이션에 해당 코드를 입력함으로써 완료되며, 해당 과정은 애플리케이션 서버와 Globus Online 시스템의 Auth 서버 사이에 Auth API 호출을 통하여 이뤄진다. 권한 부여 절차가 완료되면 애플리케이션에서 Transfer API를 호출 가능한 상태로 변경

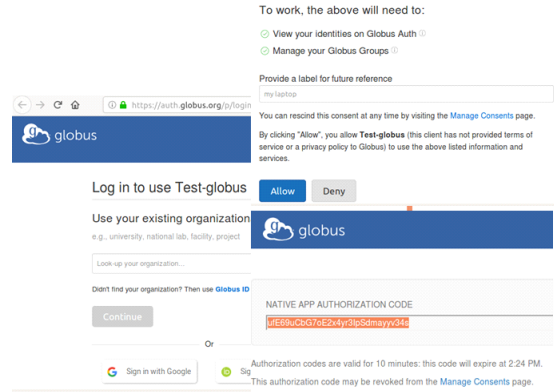


Fig. 8 Acquisition of Globus Online authorization code

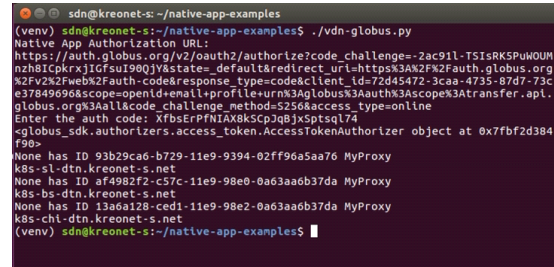


Fig. 9 Result of Globus Online Auth/Transfer based application execution

한다. 그림 9에서는 그림 8에서 획득한 권한 부여 코드를 애플리케이션에 입력함으로써 Transfer API 중 DTN 리스트를 획득하여 출력한 결과를 보인다. 그리고 이는 Globus Online 시스템에 등록된 Kubernetes 컨테이너 DTN 리스트 정보와 동일하다.

#### IV. 오픈 API 기반 가상 네트워크 프로비저닝 자동화 플랫폼

본 장에서는 2장과 3장에서 제시한 VDN, Kubernetes, Globus Online 시스템 연동 및 환경 설정과 Globus Online 애플리케이션 등록 및 오픈 API 인증 등을 토대로 대용량 데이터 전송을 위한 오픈 API 기반 가상 네트워크 프로비저닝 자동화 플랫폼을 구현한 결과와 이에 따른 DTN 간 데이터 전송 성능을 보인다.

KREONET-S의 VDN 시스템과 Kubernetes 시스템의 오픈 API는 REST 형태로 제공하며, 관련 래퍼 클래스(Wrapper Class)를 KREONET-S/VDN 상에 구현하

```
Deploy-M101:/home/sdn/vdn_jsondb>cat globus-dtn.json
{
  "apps": {
    "kr.re.kisti.vdn-fw": {
      "dtnIPList": [
        {
          "id": "0",
          "name": "k8s-bs-dtn.kreonet-s.net",
          "ip": "203.230.72.3"
        },
        {
          "id": "1",
          "name": "k8s-sl-dtn.kreonet-s.net",
          "ip": "203.230.72.4"
        },
        {
          "id": "2",
          "name": "k8s-bs-dtn.kreonet-s.net",
          "ip": "203.230.72.13"
        }
      ]
    }
  }
}
```

Fig. 10 DTN list information JSON format conversion

였다. Globus Online 시스템은 Python 언어 기반의 오픈 API를 제공하며, 3장에서 언급한 바와 같이 인증된 독립적인 Python 애플리케이션(ex, vdb-globus.py)이 요구된다. 이에 따라, 해당 API를 KREONET-S/VDN 환경에서 활용하기 위하여, 별도의 Python 애플리케이션을 실행하고 Globus Online 시스템에 등록된 DTN 리스트 정보를 획득하여 그림 10과 같이 KREONET-S/VDN가 인식할 수 있는 JSON 파일(ex, globus-dtn.json)로 변환하였다.

새로운 DTN 간 통신을 위한 가상 네트워크 프로비저닝 자동화를 위하여 먼저 네트워크상의 새로운 데이터 통신 발생을 감지해야 한다. 이를 위하여 제안 플랫폼에서는 ONOS가 제공하는 PacketProcessor 기능을 활용하여 SDN 프로토콜인 Openflow의 PACKET\_IN 메시지를 지속적으로 관측하였다. PacketProcessor는 임의의 호스트로부터 패킷이 발생하면 액세스 SDN 스위치에서 ONOS 컨트롤러로 보내는 PACKET\_IN 메시지 정보를 수신하였다. 그리고 이 때, 신규 데이터 통신이 DTN 간 이뤄진 것인지 확인하기 위하여 PACKET\_IN 메시지의 헤더에서 출발지와 목적지 정보를 추출하고 DTN 리스트 정보와 일치하는지 비교하였다. 그리고 만약 인가된 DTN 간 통신일 경우에 VDN API 호출을 통하여 가상 네트워크를 자동으로 생성하도록 구현하였다.

가령, Globus Online에 등록된 DTN 노드의 VDN 프로비저닝 자동화 기능을 활성화하고 203.230.72.3 주소를 지닌 DTN k8s-bs-dtn.kreonet-s.net에서 203.230.72.4 주소를 지닌 DTN k8s-sl-dtn.kreonet-s.net로의 통신을

```
[root@k8s-bs-dtn /]# ping 203.230.72.4
PING 203.230.72.4 (203.230.72.4) 56(84) bytes of data.
64 bytes from 203.230.72.4: icmp_seq=2 ttl=64 time=1123 ms
64 bytes from 203.230.72.4: icmp_seq=3 ttl=64 time=175 ms
64 bytes from 203.230.72.4: icmp_seq=4 ttl=64 time=7.12 ms
64 bytes from 203.230.72.4: icmp_seq=5 ttl=64 time=7.11 ms
64 bytes from 203.230.72.4: icmp_seq=6 ttl=64 time=7.11 ms
64 bytes from 203.230.72.4: icmp_seq=7 ttl=64 time=7.13 ms
64 bytes from 203.230.72.4: icmp_seq=8 ttl=64 time=7.11 ms
64 bytes from 203.230.72.4: icmp_seq=9 ttl=64 time=7.12 ms
64 bytes from 203.230.72.4: icmp_seq=10 ttl=64 time=7.09 ms
64 bytes from 203.230.72.4: icmp_seq=11 ttl=64 time=7.10 ms
64 bytes from 203.230.72.4: icmp_seq=12 ttl=64 time=7.09 ms
64 bytes from 203.230.72.4: icmp_seq=13 ttl=64 time=7.11 ms
64 bytes from 203.230.72.4: icmp_seq=14 ttl=64 time=7.12 ms
64 bytes from 203.230.72.4: icmp_seq=15 ttl=64 time=7.10 ms
64 bytes from 203.230.72.4: icmp_seq=16 ttl=64 time=7.11 ms
```

Fig. 11 Communication success by VDN provisioning automation

시도하면 PACKET\_IN 메시지 정보를 토대로 DTN 간 통신 여부를 확인 할 수 있으며, 이에 따라 VDN을 자동 생성하여 그림 11와 같이 통신이 수행됨을 알 수 있다. 또한 KREONET-S VDN 리스트 정보를 확인하면 그림 12와 같이 1Gbps 대역폭을 지닌 VDN-03739e90가 자동 생성되었음을 확인할 수 있다. 가상 네트워크 프로비저닝 자동화 플랫폼에서 생성되는 VDN 네트워크 대역폭은 사전에 1Gbps로 설정해 두었다. 한편, VDN NAME인 VDN-03739e90는 해당 VDN ID인 UUID에서 앞자리를 파싱하여 자동 생성된다.

한편, Globus Online에 등록된 DTN 간의 VDN 생성을 수행함에 있어 DTN의 SDN-IP 통신을 위하여 자동으로 해당 IP 대역의 게이트웨이를 포함하여 VDN을 생성하도록 구현하였다. 이 때, DTN과 동일 프리픽스를 지닌 SDN-IP 게이트웨이는 사전에 KREONET-S 네트워크에 생성해 두었다. 그림 12는 자동 생성된 VDN-03739e90의 네트워크 토폴로지를 보여주며, 서울 및 부산 지역의 DTN 노드와 함께 SDN-IP 통신을 위한 SDN-IP 게이트웨이가 포함되어 VDN이 구성됨을 확인

ID	NAME	IP	PORT	STATUS
BACKBONE	10.0.0.0/24	10.0.0.1	25565	OK
AUTO SCALE	10.0.0.0/24	10.0.0.1	25565	OK
PROACTIVE	10.0.0.0/24	10.0.0.1	25565	OK
VMAC GW	10.0.0.0/24	10.0.0.1	25565	OK
VDN1	10.0.0.0/24	10.0.0.1	25565	OK
VDN2	10.0.0.0/24	10.0.0.1	25565	OK
VDN3	10.0.0.0/24	10.0.0.1	25565	OK
VDN4	10.0.0.0/24	10.0.0.1	25565	OK
VDN5	10.0.0.0/24	10.0.0.1	25565	OK
VDN6	10.0.0.0/24	10.0.0.1	25565	OK
VDN7	10.0.0.0/24	10.0.0.1	25565	OK
VDN8	10.0.0.0/24	10.0.0.1	25565	OK
VDN9	10.0.0.0/24	10.0.0.1	25565	OK
VDN10	10.0.0.0/24	10.0.0.1	25565	OK
VDN11	10.0.0.0/24	10.0.0.1	25565	OK
VDN12	10.0.0.0/24	10.0.0.1	25565	OK
VDN13	10.0.0.0/24	10.0.0.1	25565	OK
VDN14	10.0.0.0/24	10.0.0.1	25565	OK
VDN15	10.0.0.0/24	10.0.0.1	25565	OK
VDN16	10.0.0.0/24	10.0.0.1	25565	OK

Fig. 12 Container DTN network information of Busan Kubernetes environment

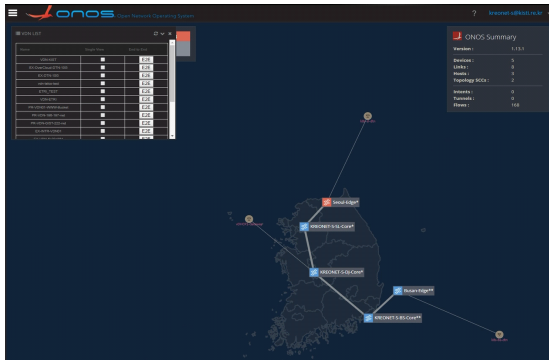


Fig. 13 Auto-generated VDN network topology

할 수 있다.

오픈 API 기반 가상 네트워크 프로비저닝 자동화 플랫폼은 대용량 데이터 전송 목적으로 개발되었기에 해당 플랫폼에 의하여 연결된 DTN간 데이터 전송 성능이 보장되어야 한다. 이를 확인하기 위하여 본 논문에서는 Iperf3와 Globus Online 시스템을 활용하였다. Iperf3는 일반적인 네트워크 전송 성능 측정 결과를 보이며, Globus Online 시스템은 실질적인 디스크 대 디스크 (D2D, Disk to Disk) 전송 성능 측정 결과를 보인다.

그림 14는 서울 Kubernetes에서 생성된 DTN k8s-sl-dtn.kreonet-s.net과 부산 Kubernetes에서 생성된 DTN k8s-bs-dtn.kreonet-s.net간의 네트워크 전송 성능 결과를 보인다. 이를 통하여 DTN간 네트워크 전송 성능이 물리적인 네트워크 인터페이스 카드(NIC, Network

Interface Card)의 성능에 거의 근접하게 달성됨을 확인할 수 있었다. 이는 KREONET-S/VDN이 1Gbps 물리적인 인터페이스를 가상 네트워크에 실질적으로 할당하였기에 달성 가능한 수치이다.

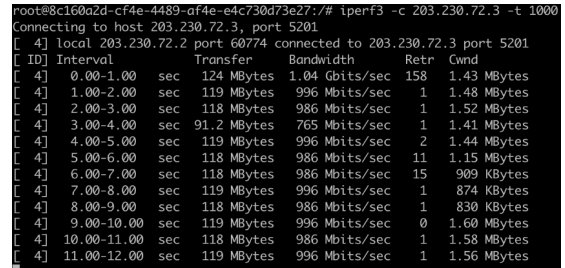


Fig. 14 Data transfer performance between DTN containers

그림 15는 서울 Kubernetes에서 생성된 DTN k8s-sl-dtn.kreonet-s.net과 부산 Kubernetes에서 생성된 DTN k8s-bs-dtn.kreonet-s.net간의 D2D 전송 성능 결과를 보인다. 이를 위하여 고성능 파일 전송 시스템인 Globus Online을 활용하여 DTN 간 10GB 파일 전송을 수행하였으며 측정 성능 결과는 98.34Mb/s로 약 800 Mbps의 D2D 전송 성능을 확인할 수 있었다. 일반적으로 D2D 전송 성능은 응용계층의 중단 시스템에서의 실질적인 데이터 송수신을 측정하기 때문에 네트워크 성능, 프로세싱 성능 등을 포함하며, 네트워크 전송 성능만의 측정 한 결과에 비하여 낮을 수밖에 없다. D2D 전송 성능 800

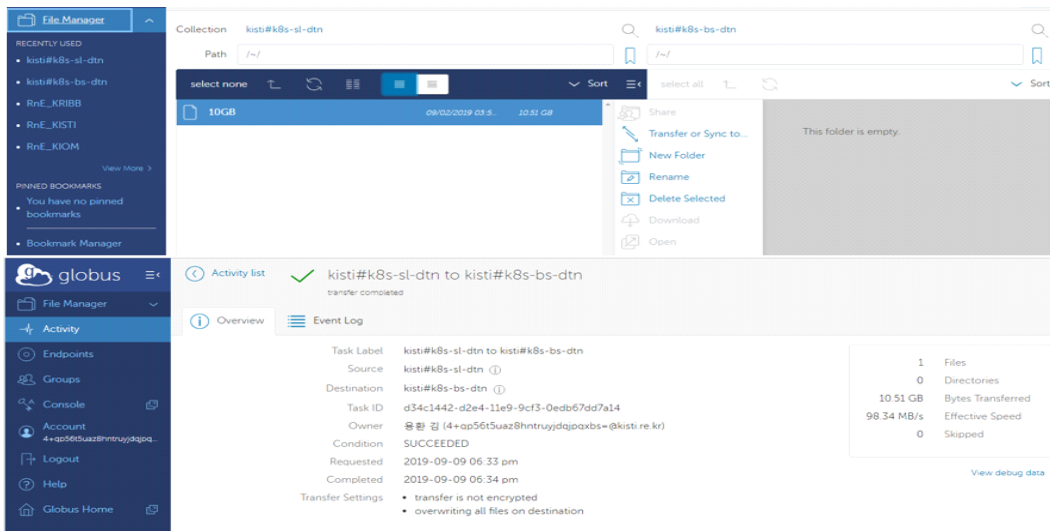


Fig. 15 D2D performance measurement results on Globus Online[17]



Mbps 수치는 대용량 데이터를 고속으로 전송하기 위해 기존 네트워크 환경에서 연구 및 범용 네트워크로 트래픽 특성별 망 분리하여 독립된 전용 회선을 제공하는 ScienceDMZ 환경하의 D2D 성능[8]과 유사한 결과이다.

## V. 결 론

본 논문에서는 새로운 네트워킹 서비스 요구사항들에 대응하기 위하여 SDN 기반의 네트워크 가상화 기술을 제공하는 KREONET-S의 VDN 시스템, 컨테이너 중심의 서버 가상화 기술을 제공하는 Kubernetes 시스템, 그리고 고성능 데이터 전송 시스템인 Globus Online을 오픈 API 기반으로 연계 통합하여 구현했다. 그리고 구현된 기술을 토대로 개발된 대용량 데이터 전송을 위한 오픈 API 기반 가상 네트워크 프로비저닝 자동화 플랫폼에 대하여 소개하였다. 이를 위하여 상이한 시스템 간의 연계를 위한 환경 설정 및 시스템 연동 방안에 대하여 제시하였고, Globus Online 시스템의 오픈 API를 연계 활용하기 위한 사용자 인증 과정 또한 보였으며, 오픈 API 기반 가상 네트워크 프로비저닝 자동화에 대한 구현 결과를 제시하였다. 이러한 결과들은 통하여 KREONET-S가 첨단연구 및 응용사용자가 필요로 하는 고정적인 네트워크 자원을 빠른 속도로 제공하는데 그치지 않고, 상황에 따라 사용자의 가상전용네트워크 자원 수요를 측정 및 예측하여 요구되는 네트워크 자원을 동적으로 제공할 수 있는 방안을 마련하였다.

이와 같은 가상전용네트워크의 자동화 프로비저닝 기술의 구현을 바탕으로, 현재 KREONET-S는 컴퓨팅(물리, 가상 자원 등), 네트워크(호스트, 디바이스, 링크, 포트 등) 자원 등의 물리망 계층 데이터, 시스템 운영 관리 등의 제어 계층 데이터, 가상망 및 관련 네트워킹 정보 등 가상화 계층 데이터 등을 수집 중에 있다. 향후 연구로는 해당 데이터를 기반으로 컴퓨팅, 네트워크 등의 다양한 자원을 인공지능 기반으로 다수의 사용자에게 적절하게 스케줄링하여 할당 및 적용하는 연구를 추진하고자 한다.

## ACKNOWLEDGEMENT

This work was supported in part by Institute of Information & communications Technology Planning & Evaluation (IITP, Development of virtual network management technology based on artificial intelligence) under Grant 2018-0-00749.

## REFERENCES

- [ 1 ] KREONET-S Website [Online]. Available: <http://www.kreonet-s.net/>.
- [ 2 ] KREONET Website [Online]. Available: <http://www.kreonet.net/>.
- [ 3 ] D. K. Kim and Y. H. Kim, "Dynamic Virtual Network Slicing and Orchestration for Selective MEC Services over Wide-Area SDN," *Algorithms*, vol. 13. no. 10, pp. 245, Sep. 2020.
- [ 4 ] Open Network Operating System Website [Online]. Available: <http://onosproject.org/>.
- [ 5 ] P. Berde, M. Gerola, J. Hart, Y. Higuchi, M. Kobayashi, T. Koide, B. Lantz, B. O'Connor, P. Radoslavov, W. Snow, and G. Parulkar, "ONOS: towards an open, distributed SDN OS," in *Proceedings of the third workshop on Hot topics in software defined networking*, Chicago: IL, USA, pp. 1-6, Aug. 2014.
- [ 6 ] Y. H. Kim, K. H. Kim, and D. K. Kim "Design and implementation of virtually dedicated network service in SD-WAN based advanced research & educational (R&E) network," *Journal of Korean Institute of Communications and Information Sciences*, vol. 42. no. 10, pp. 2050-2064, Oct. 2017.
- [ 7 ] D. K. Kim, Y. H. Kim, K. H. Kim, and J. M. Gil, "Cloud-Centric and Logically Isolated Virtual Network Environment Based on Software-Defined Wide Area Network," *Sustainability*, vol. 9. no. 12, pp. 2382, Dec. 2017.
- [ 8 ] E. Dart, L. Rotman, B. Tierney, M. Hester, and J. Zurawski, "The Science DMZ: A network design pattern for data-intensive science," *Scientific Programming*, vol. 22, no. 2, pp. 173-185, Aug. 2014.
- [ 9 ] J. Crichigno, E. Bou-Harb, and N. Ghani, "A Comprehensive Tutorial on Science DMZ," *IEEE Communications Surveys & Tutorials*, vol. 21. no. 2, pp. 2041-2078, Oct. 2018.

- [10] Q. Lu, L. Zhang, S. Sasidharan, W. Wu, P. DeMar, C. Guok, J. Macauley, I. Monga, S. Yu, J. H. Chen, J. Mambretti, J. Kim, S. -Y. Noh, X. Yang, T. Lehman, and G. Liu, "BigData Express: Toward Schedulable, Predictable, and High-Performance Data Transfer," in *2018 IEEE/ACM Innovating the Network for Data-Intensive Science (INDIS)*, pp. 75-84, Dallas: TX, USA, 2018.
- [11] M. Kiran, S. Campbell, F. B. Wala, N. Buraglio, and I. Monga, "Machine learning-based analysis of COVID-19 pandemic impact on US research networks," *ACM SIGCOMM Computer Communication Review*, vol. 51. no. 4, pp. 23-35, Oct. 2021.
- [12] Kubernetes Website [Online]. Available: <https://kubernetes.io/>.
- [13] D. Bernstein, "Containers and Cloud: From LXC to Docker to Kubernetes," *IEEE cloud computing*, vol. 1 no. 3, pp. 81-84, Sep. 2014.
- [14] Globus Online Website [Online]. Available: <https://www.globus.org/tags/globus-online>.
- [15] Docker Website [Online]. Available: <https://www.docker.com>.
- [16] Oauth Website [Online]. Available: <https://oauth.net/2/>.
- [17] Y. H. Kim and D. K. Kim, "Implementation of an Orchestration System for Distributed Cloud Environments Based on Software Defined Networking," *Journal of Korean Institute of Communications and Information Sciences*, vol. 46. no. 2, pp. 280-292, Feb. 2021.



**김용환 (Yong-hwan Kim)**

2010년 8월: 한국기술교육대학교 정보미디어공학과 석사  
 2015년 8월: 한국기술교육대학교 컴퓨터공학과 박사  
 2016년 2월~현재: 한국과학기술정보연구원(KISTI) 선임연구원  
 ※ 관심분야: SDN/NFV, 양자암호통신, 5G/6G



**박성진 (Seongjin Park)**

2012년 2월: 부산대학교 정보컴퓨터공학과 졸업  
 2014년 8월: 부산대학교 전자전기컴퓨터공학과 석사  
 2020년 2월: 부산대학교 전기전자컴퓨터공학과 박사  
 2020년 9월~현재: 한국과학기술정보연구원 박사후연구원  
 ※ 관심분야: 네트워크 자동화/지능화, 강화학습



**김동균 (Dongkyun Kim)**

1999년 2월: 충남대학교 컴퓨터공학과 석사  
 2005년 2월: 충남대학교 컴퓨터공학과 박사  
 2006년 4월~2007년 3월: 미국 테네시대학(UT)/오크리지국립연구소(ORNL) 방문연구원  
 2000년 6월~현재: 한국과학기술정보연구원(KISTI) 책임연구원  
 ※ 관심분야: SDN/NFV, 5G/6G, Advanced Research Networks