OpenFlow 기반 KREONET 가상 네트워크 플랫폼 연구

석승준, 정현욱
경남대학교 컴퓨터공학과

A Study on OpenFlow based Virtual Network Platform for KREONET

Seung-Joon Seok, Hyeonuk Jeong

Dept. of Computer Engineering, Kyungnam University

요 약 국제적으로 논의되고 있는 미래 인터넷의 핵심 특징 중의 하나는 네트워크 가상화 서비스 이다. 최근 논의되고 있는 네트워크 가상화 기술에는 네트워크 기능요소의 가상화와 가상 네트워크 서비스의 두 가지 접근 방법이 있다. 미국의 GENI를 비롯한 여러 국가의 연구망을 중심으로 미래 인터넷 기술을 실험하고 있으며 그 중 가상화 서비스가 주요 이슈에 포함되어 있다. 국내 대표적인 연구망인 KREONET에서도 미래 인터넷 서비스 도입을 위한 단계로서 미래 인터넷을 위한 핵심 네트워크 모델인 SDN/OpenFlow의 가상화 모델을 사용하여 가상 네트워크 프레임워크 구축을 추진하고 있다. 이를 위해 본 논문에서는 KREONET에서 가상 네트워크 서비스를 도입하기 위한 단계적 모델을 제시한다. 먼저 KREONET 사용자의 가상 네트워크 서비스 요구사항을 분석하고, 요구사항을 만족시킬 수 있는 자원 관리 방안 및 가상 네트워크 구축 방안을 제시한다. 마지막으로 KREONET 가상 네트워크 모델의 타당성을 검증한다.

주제어 : 가상 네트워크, 네트워크 가상화, 오픈플로우, 소프트웨어 정의 네트워크, 첨단연구망

Abstract Virtual Network service is a key characteristics of future Internet which is debate internationally. There are two kinds of network virtualization technologies considered lately: network functions virtualization and virtual network approaches. Several national wide research networks including US's GENI project have experimented technologies for future Internet and in particular network virtualization is one of key issues. Representative Korean research network, KREONET, is working on deploying virtual network framework as a preliminary for future Ineternet using the virtualization model of SDN/OpenFlow which is typical network model of future Internet. This paper proposes a stepwise model to bring virtual network services in KREONET. Firstly, we requirements of KREONET users' virtual network service and network resource management and network deploying virtual network. Finally, we verify the adequacy of our virtual network model for KREONET.

Key Words : Virtual Network, Network Virtualization, OpenFlow, Software Defined Network, KREONET

* 이 논문은 2013년도 정부(미래창조과학부)의 지원으로 한국과학기술정보연구원의 슈퍼컴퓨팅서비스 기반 첨단연구망 구축/운영 및 서비스 고도화 사업의 지원을 받아 수행된 연구임

Received 11 June 2014, Revised 21 July 2014

Accepted 20 August 2014

Corresponding Author: Seung-Joon Seok(Dept. of Computer Engineering, Kyungnam University)

Email: sjseok@kyungnam.ac.kr

ISSN: 1738-1916

© The Society of Digital Policy & Management. All rights reserved. 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.

1. 서론

미래 인터넷 연구용 테스트베드를 위해 미국 스탠퍼드 대학에서 처음 제안된 OpenFlow 기술은 이 후 SDN (Software Defined Network) 이라는 범용 네트워크 모델로서 발전되어 왔다. 특히, 최근에는 미래 인터넷을 위한 네트워크 기술로서 SDN이 적극적으로 고려되고 있다. SDN/OpenFlow 기술의 가장 큰 특징은 전체 혹은 일부 네트워크 영역 내 스위치들의 내부 제어 기능을 분리하여 집중화된 하나의 제어 노드에서 수행되도록 한다는 것이다. 또한 중앙 제어 노드는 네트워크 내 정보를 수집하고 스위치를 직접 제어할 수 있으며 기능 추가를 위한 오픈 인터페이스를 갖추고 있다. 따라서 개발자 혹은 네트워크 서비스 제공자는 제어기 내 기능 블록을 추가함으로써 스위치 구조의 변경 없이 서비스 추가 및 개선이 가능하게 된다. 여러 인터넷 장비 벤더 및 연구소들에서 SDN 모델을 연구하고 있으나 현재까지는 스탠퍼드 대학의 OpenFlow가 가장 널리 고려되고 있는 모델이다. 미래 인터넷을 위한 SDN에서는 제어기를 통해서 네트워크 운영체제의 개념을 구현하도록 하고 있다. 몇몇 벤더에서 개발한 SDN 제어기는 OpenFlow 프로토콜/기술을 기본으로 지원하면서 자체적인 네트워크 운영체제 기능을 구현하고 있다 [1, 2, 3, 4].

최근 네트워크 분야에서 활발히 논의되고 있는 주제는 가상화(Virtualization)이다. 무엇을 가상화 할 것인가 (가상화 대상) 혹은 어떻게 가상화 할 것인가 (가상화 방법) 등에 있어서 많은 아이디어 및 솔루션들이 하루가 다르게 발표되거나 개발되고 있다. 특히 최근에는 클라우드 컴퓨팅(Cloud Computing) 기술과 네트워크 가상화를 결합하고자 하는 시도까지 진행되고 있다 [8].

네트워크 가상화 서비스 중 특히 가상 네트워크(Virtual Network)는 인터넷 서비스 제공자(ISP) 입장에서 매우 매력적인 문제이다. 즉 별도의 회선, 교환기, 라우터 장치 등을 구축하지 않고 사용자가 요구하는 독립적인 네트워크를 자유롭게 제공할 수 있게 되는 것이다. 이는 혼잡한 네트워크 자원을 공유하는 방식으로 사용자별 서비스 품질(QoS)을 관리하는 과거의 방법과는 근본적으로 다른 접근 방법이 될 수 있다.

가상 네트워크는 네트워크를 논리적으로 분할하는 것을 의미한다. 이러한 네트워크 분할은 네트워크 장치를

논리적으로 분할하는 방식과 네트워크 장치에 입력되는 트래픽을 논리적으로 분할하는 방식이 있다. 또한 데이터 평면에서 트래픽을 직접 다루는 방법과 제어 평면에서 간접적으로 분할을 시도하는 접근 방법이 있다. 이 중 OpenFlow/SDN 기술은 제어평면에서 트래픽을 분할하는 방식으로 가상 네트워크 서비스를 구현한다. 이 OpenFlow 기술은 지금까지의 여러 가상화 기술 중 가장 구체적으로 발전되고 있으며 미국, 일본, 중국 벤더들 중심으로 활발한 움직임을 보이고 있다. OpenFlow 기술이 현실적으로 당장 고려될 수 있는 네트워크 가상화 기술이다 [5, 6].

하나의 OpenFlow 네트워크는 다수의 스위치와 하나의 제어기로 구성되며, 제어기는 네트워크 내 모든 스위치에 대한 제어 평면 기능을 수행해야 한다. 일반적으로 물리 OpenFlow 스위치들은 하나의 제어기와만 연동되며, 단일 OpenFlow 네트워크만을 구성한다. 하지만 OpenFlow 네트워크 가상화 기술은 물리 스위치들이 다중의 OpenFlow 네트워크를 구성할 수 있도록 한다. 이 경우 각 네트워크는 별도의 OpenFlow 제어기를 둔다 [5, 6].

OpenFlow에서 사용하는 가상화 방식은 다른 가상화 방식보다는 좀 더 소프트웨어적인 방식으로 물리 장치(스위치)들의 자원을 직접 분할하지 않고 플로우바이저(FlowVisor) 노드가 스위치와 제어기 중간에서 에이전트 역할을 수행해서 네트워크 사용자에게 네트워크가 가상화된 것처럼 보이게 하는 기술이다 [5, 6, 7, 8]. 플로우바이저는 스위치와 제어기가 상호 교환하는 모든 제어 메시지를 분석하고 미리 설정된 내부 정책에 맞도록 메시지를 변환한 후 상대방 스위치 혹은 제어기들에게 전달한다.

OpenFlow 시스템에서 플로우바이저는 가상 네트워크 별로 동작하는 제어기와 스위치 사이에서 모든 제어 정보를 중간에서 수집, 분석, 변환한 후 스위치에 전달되도록 한다. 이 제어 정보에는 각 제어기의 스위치의 플로우 테이블(FlowTable) 항목 생성 메시지도 포함된다. 이 과정에서 미리 정의된 가상 네트워크 정책(Slice)이 반영되어 망 자원이 가상 네트워크별로 분할될 수 있도록 한다. OpenFlow 네트워크에서는 네트워크 구성요소(스위치, 포트 등)로 슬라이스를 정의하지 않고 가상 네트워크를 통해 서비스되는 트래픽으로 슬라이스를 정의한다.

[Fig. 1] OpenFlow network virtualization service model

즉 슬라이스는 단일 정책이 적용되는 트래픽 요소인 플로우 스페이스(FlowSpace)을 하나이상 갖추고 있어야 한다.

본 논문에서는 KREONET에서 가상 네트워크 서비스를 도입하기 위한 단계적 모델을 제시한다. 먼저 KREONET의 가상 네트워크 서비스 요구사항을 분석하고 요구사항을 만족시킬 수 있는 자원 관리 방안 및 가상 네트워크 구축 방안을 제시한다. 마지막으로 OpenFlow 네트워크를 활용한 KREONET 가상 네트워크 서비스 모델의 타당성을 검증한다.

2. KREONET 가상화 네트워크 요구사항

국내에서 항상 선도적인 인터넷 기술을 도입해온 KISTI의 첨단연구망(KREONET)에서 가상화 플랫폼 도입을 본격적으로 준비하고자 본 연구에서는 OpenFlow 기반 가상화 네트워크 모델을 기반으로 KREONET 가상 네트워크 프레임워크 구축을 추진한다.

아래 <Table 1>은 현재 주요 KREONET 활용 분야별 네트워크 서비스 요구 사항을 정리하고 있다. 이 중 기상, 천체, 고에너지, Bio 생물연구 분야는 사용량, 데이터 타입, 서비스 타입 에서 유사한 특성을 갖는다. 정리하면 이들 서비스는 대용량 컴퓨팅 데이터 송수신을 위하여 안정적인 대역의 데이터 전송 서비스를 지속적으로 KREONET에 요구한다. 하지만 원격 교육, 원격 의료, 문화 공연 분야는 이벤트가 있을 때 대용량 멀티미디어 트래픽 전송을 KREONET을 통해 시도한다.

[Fig. 2] The concept of KREONET virtual network service

<Table 1> Network service requirements of KREONET major user fields

KREONET User Field	Application Field	Required Bandwidth	Flow Type	Service Type
Climate	GRID, Super Computing	Broad	TCP	Constant
Astronomy	GRID, Super Computing	Broad	TCP	Constant
High Energy Phisics	GRID, Super Computing	Broad	TCP	Constant
Bio	GRID, Super Computing	Medium	TCP	Constant
Super Computer Applications	Super Computing	narrow	TCP	Event
Remote Education	Multimedia	narrow	TCP+UDP	Event
Remote Medicine	Multimedia	Broad	TCP+UDP	Event
Cultural Performance	Multimedia	Broad	UDP	Event

<Table 1>의 KREONET 사용 분야별 네트워크 요구 사항을 수용하기 위한 OpenFlow 기반 가상 네트워크 플랫폼 구축 시 다음 세 가지 사항을 고려해야 한다.

- 사용자 요구별 가상 네트워크(슬라이스)를 제공한다.
- 사용자 요구를 동적으로 수용하기 위하여 동적 가상 네트워크(슬라이스) 구성이 가능하도록 한다.
- 구성된 가상 네트워크(슬라이스) 간 상호 영향이 최소화 되도록 독립성(Isolation)을 보장한다.

첫 번째 항목은 사용자가 가상 네트워크를 요청할 수 있고 KREONET에서는 사용자 요구별 슬라이스를 제공해야 한다는 것이다. 두 번째는 사용자의 가상 네트워크 요구를 언제든지 수용하기 위한 방안이 KREONET 가상

네트워크 플랫폼 구축 시 고려되어야 한다는 의미이다. 마지막으로 가상 네트워크(슬라이스)는 상호 독립성이 보장되도록 해야 한다. 즉 가상 네트워크 사용자는 자신이 받는 서비스가 물리 네트워크와 구분하지 못하도록 가상 네트워크의 서비스 품질이 관리되어야 한다. 이는 사용자는 일반적으로 가상 네트워크를 통해서 물리 네트워크 특성을 요구하기 때문이다.

위에서 언급된 고려사항과 <Table 1> KREONET 사용자 요구 특성을 반영하여 본 논문에서는 OpenFlow 기반 KREONET 가상 네트워크 플랫폼에 대한 요구사항을 아래와 같이 정리한다.

- OpenFlow 스위치 간 1Gbps 이상의 링크 구축
- 백본 OpenFlow 스위치의 안정적인 QoS 기능 동작
- 사용자 요구 특성을 고려한 OpenFlow 컨트롤러 응용 개발
- VMWare 혹은 Cloud 컴퓨팅 서비스를 통한 동적 OpenFlow 컨트롤러 구성
- 사용자 요구 수용과 가상 네트워크 서비스 플랫폼 제어를 실시간으로 수행할 수 있는 통합 제어 시스템 구축
- 국가 간 연구를 지원하기 위해 해외 OpenFlow 기반 연구망과의 가상 네트워크(Slice) 연동

3. KREONET 가상화 네트워크 모델

본 연구에서는 가상화 네트워크 서비스를 KREONET 사용자들에게 동적으로 제공하기 위해 OpenFlow 네트워크 플랫폼으로 [Fig. 3]의 모델을 제안한다.

OpenFlow 네트워크에서 가상 네트워크(슬라이스)를 생성하기 위해서는 먼저 플로우바이저(FlowVisor)에 슬라이스를 정책이 등록되어야 한다. 또한 OpenFlow 시스템에서는 가상 네트워크(슬라이스)별로 전용의 제어기가 필요하기 때문에 플로우바이저 슬라이스 정책에 등록 시 제어기 정보가 지정되어야 한다. 따라서 제안하는 모델에서는 사용자 요구에 따라 언제든지 사용가능한 OpenFlow 제어기를 준비해야 하는 방안이 마련되어야 한다. 하지만 슬라이스별로 필요한 제어기를 무한정 첨단 연구망 내에 미리 준비해 놓을 수가 없기 때문에 본

[Fig. 3] KREONET virtual network service framework

논문에서는 VMWare 시스템을 구축하고 VM 상에서 OpenFlow 제어기/컨트롤러를 설치하고 구동하는 방법을 제안한다. 논문에서 사용한 VMWare는 Xen 서버 시스템이고 다중의 VM을 사용하여 별도의 하드웨어 장비 없이도 OpenFlow 제어기를 생성하도록 한다. 하지만 고성능 제어기가 필요한 경우를 위하여 제어기 시스템을 별도로 구축하고 제어 네트워크 구성을 효과적으로 수행하기 위하여 다중 플로우바이저를 계층적으로 구성하도록 한다. 플로우바이저에서 슬라이스 정책별 망 자원을 할당해야하기 때문에 응용 서비스별 자원 특성을 고려하여 유사 서비스별로 플로우바이저를 설치할 수 있도록 한다. 마지막으로 가상 네트워크 관리 시스템을 통하여 사용자의 가상 네트워크 요청을 수용하고 가상 네트워크를 구성 및 관리할 수 있도록 한다.

슬라이스를 생성하는 동작은 플로우바이저에 슬라이스 정책이 등록되는 과정을 통해 진행된다. 이때 가장 핵심 내용은 슬라이스와 연동되는 제어기를 등록하는 동작과 슬라이스를 구성하는 트래픽을 정의하는 것이다. 앞서 1장에서 언급한 것과 같이 OpenFlow에서는 가상 네트워크를 구분하는 기준으로 트래픽 범위를 사용하기 때문에 플로우바이저에는 슬라이스 트래픽 정의를 하나 이상의 플로우 스페이스(Flow Space)를 통해 등록하도록 한다. 정리하면 동적으로 가상 네트워크를 제공하기 위해 본 논문에서 제안하는 방안은 먼저 슬라이스 정책을 플로우바이저에 등록하고 Xen 서버에 OpenFlow 컨트롤러를 동작시키는 절차 순으로 진행된다.

제안하는 플랫폼에서는 슬라이스 생성 동작을 진행하

기 위한 가상 네트워크 관리 시스템(Virtual Network Management System)을 둔다. 사용자 요구가 있을 경우 관리자는 가상 네트워크 관리 시스템 혹은 관리 시스템 스스로 플로우바이저와 VMWare의 관련 절차를 수행하도록 한다.

3.1 슬라이스 구성 단계

OpenFlow 네트워크에서 가상 네트워크/슬라이스는 플로우바이저 구성 파일에 슬라이스 정보를 포함함으로써 설정된다. 플로우바이저 구성 파일에서 OpenFlow 슬라이스는 플로우 스페이스(FlowSpace) 집합으로 정의된다. 플로우 스페이스는 슬라이스를 이용할 수 있는 트래픽 범위를 정의하는 방법으로 <Table 2> 항목을 모두 혹은 일부분만을 사용한다.

<Table 2> FlowSpace Components

FlowSpace Component	Usage Description
Switch	FlowSpace should indicate switches which compose a virtual network / slice. All or partial switches of physical network can be included in a slice. However, switches of slice should make a kind of connected graph.
Switch's I/F(port)	A FlowSpace can be defined by specific I/Fs(input and/or output ports) of switches. If FlowSpace doesn't specify I/Fs, all it includes all I/Fs. Indicating I/Fs(ports) means that virtual network service is provided to all traffic incoming via I/Fs as well as that slice uses port resources.
MAC address	A FlowSpace can be defined by Source and/or destination MAC addresses.
EtherType	A FlowSpace can be defined by the type of Ethernet's payload information.
IP address	A FlowSpace can be defined by specific source and/or destination IP addresses or IP address blocks.
Protocol, Port #	Source/Destination Port number should be used with Protocol field.

[Fig. 3]의 제안하는 프레임워크에서는 가상 네트워크 관리 시스템을 통해 사용자 가상 네트워크 요구를 받아 [Fig. 4]의 절차를 거쳐 슬라이스를 설정한다.

[Fig. 4] Slice setup process of virtual network management system

3.2 Xen을 이용한 다중 제어기 구성

KREONET의 가상화 네트워크 요구 사항 중 동적 가상 네트워크 구성을 위해서는 제어기 생성에 관한 고려가 필요하다. 앞서 언급하였듯이 가상 네트워크(슬라이스)별 제어기 노드가 필요하기 때문에 KREONET 내부에서 제어기 노드 생성에 관한 방안이 고려되어야 한다. 본 연구에서는 컴퓨터 시스템 가상화 솔루션인 Xen VMware가 구축된 제어기 서버를 두고 슬라이스 생성 시 VM(가상 머신)을 동적으로 생성하고 Floodlight를 설치하도록 한다.

3.3 가상 네트워크 자원 제공 방안

네트워크 가상화 서비스를 제공하기 위해서는 가상화 네트워크(슬라이스) 사이가 연결성 측면에서 분할되는 것에 더해서 링크 자원 즉 대역이 분할되도록 해야 한다. OpenFlow 기술에서는 스위치에서 고유하게 가지고 있는 QoS/Scheduling 기능을 활용하여 슬라이스별 자원 분할을 구현한다. 이 경우 스위치 성능에 따라서 자원 분할 성능이 결정된다. KREONET에 설치된 Pronto 329x 스위치에서 사용되는 OVS(Open vSwitch) 프레임워크는 최소 대역폭 보장이 가능한 스케줄러(Scheduler) 기능

을 갖추고 있다.

본 연구에서 가상 네트워크를 위한 자원 분할을 위한 방법으로는 슬라이스에 부여되는 큐 및 속성(최소/최대 대역폭)이 스위치별로 미리 생성되어야 하고 플로우 스페이스 정의 시 해당 트래픽을 위한 큐가 스위치별로 지정되도록 해서 OpenFlow 스위치 FlowTable 엔트리 생성 시 Action 필드에서 해당 큐로 패킷 입력을 표시하여 스위치의 대역 보장 서비스를 패킷이 받을 수 있도록 한다. 이 경우 플로우 스페이스를 스위치(& 인터페이스) 별로 지정하여 각 링크별로 다른 큐를 지정할 수 있다. [Fig. 5]는 OpenFlow 가상 네트워크 링크 대역 제공 모델을 나타내고 있다.

[Fig. 5] OpenFlow virtual network resource assignment model

[Fig. 6] The first phase of KREONET virtual network service platform

[Fig. 7] The second phase of KREONET virtual network service platform

[Fig. 8] The third phase of KREONET virtual network service platform

4. KREONET 가상 네트워크 서비스 플랫폼 단계적 구축 방안

논문에서는 앞서 제안한 OpenFlow 기반 KREONET 가상 네트워크 서비스 플랫폼에 대한 3단계 구축방안을 제안한다. 먼저 1단계 모델([Fig. 6])에서는 Pronto 3295 스위치를 사용하여 OpenFlow 네트워크를 구축한다. 호스트 단말 중심으로 사용자를 연동하며 이벤트 응용중심으로 가상 네트워크 서비스를 제공한다. 또한 동적 제어기를 구동하기 위하여 Xen 서버를 두도록 한다. 사용자

는 오프라인으로 관리자에게 가상 네트워크 생성을 요청하면 관리자는 플로우바이저와 스위치 그리고 컨트롤러를 설정하여 가상 네트워크(슬라이스)를 생성한다. 슬라이스 그룹별로 스위치 큐를 배정한다.

2단계 모델([Fig. 7])에서는 Pronto 3295 스위치와 더불어 QoS 성능이 우수한 백본 스위치를 도입한다. 호스트와 더불어 네트워크 단위로 사용자를 연동하며 이벤트 응용과 상시 응용을 모두 지원하도록 한다. 다중 플로우바이저를 설치하여 슬라이스의 체계적 관리를 시도한다. 사용자는 웹 시스템을 통해 가상화 서비스 요청하며 네트워크 관리자용 시스템도 구축한다. 마지막으로 슬라이스별로 큐를 배정하여 대역할당을 보장하도록 한다.

마지막 3단계 모델([Fig. 8])에서는 QoS(Quality of Service) 성능이 우수한 백본 스위치로 네트워크를 구성한다. 네트워크 중심으로 사용자를 연동하며 이벤트 응용과 상시 응용을 모두 지원하도록 한다. 사용자 관리 시스템과 네트워크 관리 시스템을 통합하여 실시간 슬라이스 할당이 가능하도록 한다. 마지막으로 응용서비스별 슬라이스 구성과 대역할당을 보장하도록 한다.

5. 테스트베드를 통한 가상화 네트워크 성능 실험

가상 네트워크 서비스 제공의 핵심 성능은 슬라이스 분할(Isolation) 성능이다. 가상 네트워크(슬라이스) 사용자는 독립된 물리 네트워크와 같은 성능을 망 서비스 제공자에게 요구한다. 즉 같은 물리 네트워크를 공유하는 슬라이스 사용자들은 다른 슬라이스 혼잡 상황에 무관하

[Fig. 9] Local test-bed data plane diagram

[Fig. 10] Local test-bed control plane diagram

게 망 서비스를 받기를 원한다. 따라서 KREONET에서 제공하고자 하는 가상 네트워크 슬라이스 분할 성능을 서비스 제공 전에 검증하는 것은 매우 중요한 의미가 있다.

OpenFlow 기술을 이용한 가상 네트워크 서비스의 타당성을 검증하기 위하여 본 논문에서는 먼저 로컬 테스트베드 실험을 수행하였다. 이 후 현재 구축 중인 KREONET의 OpenFlow 네트워크에서 가상 네트워크(슬라이스) 구성 실험을 수행하였다. [Fig. 9]와 [Fig. 10]은 로컬 테스트베드의 데이터평면과 제어평면의 구성도이다. 로컬 네트워크에서는 OpenFlow@KREONET에 설치된 Pronto 3290 스위치와 동일한 OVS 1.10.0 모듈을 Linux(Ubuntu 12.04 TLS)에 설치하여 스위치로 사용하고 있으며, Pronto 스위치와 혼합하여 네트워크를 구성한다. 제어평면에서는 컨트롤러로서 Floodlight 0.9 그리고 플로우바이저 1.4 Maint 버전을 사용하고 다중 플로우바이저 성능을 검증하기 위하여 플로우바이저 1과 2를 계층적으로 구성한다. 마지막으로 논문에서 제시한 가상 네트워크 서비스 프레임워크 모델과 같이 다중의 컨트롤러를 동적으로 구동하기 위한 VM Ware로서 본 실험에서는 Xen Server 6.2를 설치한다.

첫 번째는 슬라이스 3에 입력되는 트래픽을 0, 50, 100, 150, 200, 250, 300, 350, 400Mbps로 점차 증가시키면서 슬라이스 3의 혼잡상황을 측정하는 실험이다. 이때 슬라이스 1, 2, 4는 1Gbps 트래픽을 입력시켜 극심한 혼잡을 유도한다. 슬라이스 1과 2가 입력되는 큐1은 Min 500 Mbps, Max 1Gbps 대역폭을 할당하고 슬라이스 3가 입력되는 큐2는 Min 200 Mbps, Max 1Gbps 대역폭을 설정

한다.

[Fig. 11] Slice isolation test results in case of UDP traffic

[Fig. 12] Slice isolation test results in case of TCP traffic

[Fig. 11]의 실험결과에서 슬라이스 1, 2, 4에 의한 극심한 혼잡 상황에서도 슬라이스 3는 일반적인 네트워크 동작을 수행하고 있다. 즉 최소 대역까지의 입력 상황에서는 혼잡과 지연이 극히 작은 값을 갖고 보장 대역 이상에서 급격히 지연과 손실이 급격히 증가하는 것을 확인한다.

두 번째 실험에서는 TCP 트래픽의 경우 슬라이스 분할 성능을 측정한다. 슬라이스 1, 2의 TCP 플로우 수를 0, 1, 5, 10, 15, 20개 순으로 점차 증가시키고, 대역을 할당받지 못한 슬라이스 4에도 TCP 플로우 수를 0, 1, 2, 3, 4, 5개 순서로 점차 증가시켜서 망 혼잡을 유도한다. 이 환경에서 슬라이스 3에 TCP 플로우를 입력하는 경우와 그렇지 않은 경우 슬라이스 지연과 손실률을 비교한다. 큐 설정은 슬라이스 1과 2가 입력되는 큐1의 대역을 Min 600Mbps, Max 1Gbps로 하고 슬라이스 3가 입력되는

[Fig. 13] Slice performance tests between slices sharing a queue in case of UDP traffic

큐2는 Min 300Mbps, Max 1Gbps로 설정하여 두 큐의 최대 대역이 상호 중첩되도록 환경을 구축하여 두 큐의 대역을 상호 영향 받는 정도를 관찰한다.

[Fig. 12]는 슬라이스 3에 TCP 플로우가 입력되는 경우 슬라이스 1, 2, 4에 TCP 플로우 수 증가에 따른 혼잡 상황과 무관하게 슬라이스 1과 2의 Throughput의 합은 최소 대역폭 600Mbps에 근접한 값을 슬라이스 3는 최소 대역폭 300Mbps와 유사한 Throughput을 일정하게 얻는 것을 보이고 있다.

세 번째 실험에서는 스위치 큐를 공유하는 두 슬라이스 사이의 상호 영향을 측정한다. 이번 실험에서는 슬라이스 1에 입력되는 트래픽을 UDP 0, 100, 200, 300, 400, 500, 600, 700Mbps로 증가시키고 슬라이스 2는 고정된 UDP 200Mbps 트래픽을 입력하여 큐1의 대역을 경쟁하도록 한다. 그리고 슬라이스 3는 UDP 1Gbps 트래픽을 고정적으로 입력하여 링크의 혼잡을 유도한다. 슬라이스 1과 2가 사용하는 큐1은 Min 600Mbps / Max 700Mbps를 할당하고 슬라이스 3가 사용하는 큐2는 Min 200Mbps / Max 300Mbps로 설정한다.

[Fig. 13]의 결과에서는 슬라이스 3는 슬라이스 1과 2에 의한 큐1의 혼잡에 영향을 받지 않고 있으며 슬라이스 1과 큐를 공유하는 슬라이스 2도 슬라이스 1의 혼잡 상황에 영향을 받지 않고 있다. 이는 스위치 스케줄러에서 큐를 공유하는 슬라이스들 간에 큐의 최소 대역을 균등분할 하고 있기 때문이다. 또한 스위치 스케줄러는 슬라이스 1과 2의 혼잡 상황에 무관하게 슬라이스 3의 트래픽을 큐2의 최대 대역 이하로 제한하는 것으로 보인다.

[Fig. 14] Slice performance tests between slices sharing a queue in case of TCP traffic

네 번째 실험에서는 세 번째 실험환경에서 TCP 트래픽을 고려한 경우이다. 슬라이스 1에 입력되는 트래픽을 TCP 플로우 0, 1, 5, 10, 15, 20개로 증가시키고 슬라이스 2와 슬라이스 3는 각각 TCP 플로우 20개와 1개로 고정한다. 슬라이스 1과 2가 사용하는 큐1은 Min 600Mbps / Max 700Mbps를 할당하고 슬라이스 3가 사용하는 큐2는 Min 200Mbps / Max 300Mbps로 설정한다.

[Fig. 14]는 큐1을 공유하는 슬라이스 1과 2의 TCP 트래픽의 성능을 보인다. 슬라이스 1의 트래픽이 없는 경우 슬라이스 2의 TCP는 큐1의 최대 대역에 가까운 Throughput을 얻는다. 이 후 슬라이스 1의 TCP 플로우가 입력되면 그 수에 상관없이 슬라이스 2이 300Mbps 정도로 축소된다. 슬라이스 2는 큐1의 최소 대역 600Mbps의 절반을 보장 받는 것으로 판단된다. 슬라이스 1은 슬라이스 2보다 약간 높은 대역을 얻는다. 이는 큐를 공유하는 슬라이스 간에 최소 대역 이상에서 우선순위가 존재함을 알 수 있다. 슬라이스 3는 큐2의 최소대역 이상을 큐1의 혼잡에 상관없이 보장받는다.

마지막 실험에서는 동적 컨트롤러 구동을 테스트하기 위하여 Xen 서버 VM(Virtual Machine)에 컨트롤러를 설치하는 경우 기존 컨트롤러에 미치는 영향을 실험한다. 실험에서는 Xen 서버를 구동하고 VM(350MB RAM)을 동적으로 생성한다. 생성된 VM에 컨트롤러인 Floodlight를 설치하면서 기존의 VM에 미치는 영향을 VM별 CPU와 RAM 사용량으로 평가한다.

[Fig. 15]는 7개 VM에 컨트롤러가 안정적으로 구동되고 VM 8에 새롭게 컨트롤러를 실행시키는 경우 VM의 CPU 사용량과 메모리 사용량에 영향을 거의 미치지 않

Name	CPU Usage	Used Memory	Disks (avg / max Kbs)	Network (avg / max Kbs)
xen Default install of XenServer	10% of 2 CPUs	3695 of 4094 MB	-	1/3
fl1	1% of 1 CPU	218 of 350 MB	1/2	0/0
fl2	1% of 1 CPU	268 of 350 MB	2/4	0/0
fl3	1% of 1 CPU	268 of 350 MB	0/0	0/0
fl4	1% of 1 CPU	171 of 350 MB	4/7	1/2
fl5	8% of 1 CPU	184 of 350 MB	1/1	0/0
fl6	1% of 1 CPU	162 of 350 MB	1/2	0/0
fl7	1% of 1 CPU	174 of 350 MB	0/0	0/0
fl8	1% of 1 CPU	161 of 350 MB	0/0	0/0
fl9	1% of 1 CPU	162 of 350 MB	0/1	0/0

[Fig. 15] The system status of 8 VMs running Floodlight controllers on Xen server

[Fig. 16] OpenFlow@KREONET topology

음을 확인할 수 있다. <Table 3>은 8개의 VM에서 Floodlight 컨트롤러가 모두 동작한 후 Xen 서버 내 VM별 자원 사용량을 정리한다. Xen 서버의 사용메모리는 VM들에 의해서 사용하는 메모리를 포함한 값이다. <Table 3> 결과로부터 컨트롤러가 안정적으로 동작한 이후 각 VM의 CPU 사용량은 작고 안정적으로 유지됨을 보이고 있다. 하지만 메모리 사용량은 평균 51% 정도의 높은 사용량을 요구한다. 그 결과 OpenFlow@KREONET 가상 네트워크 서비스를 위해 사용되는 Xen 서버 시스템은 RAM 크기만 갖춰진다면 CPU 혹은 다른 자원의 제약은 많지 않다는 것을 확인한다.

<Table 3> Comparison of VM resource consumption after running Floodlight OpenFlow controllers on 8 VMs with 350MB memory

Resources	VM 1	VM 2	VM 3	VM 4	VM 5	VM 6	VM 7	VM 8	XenServer	Avg.
CPU [%]	1	1	1	8	1	1	1	1	10	1.8%
Used Memory [MB]	218	268	171	184	162	174	161	162	3695/4094	187.5
Memory Using Ratio [%]	62	76	48	52	46	49	46	46		51.12

[Fig. 16]은 KREONET의 OpenFlow 테스트베드 구성도이다. 본 연구에서는 OpenFlow@KREONET의 가상 네트워크 서비스 타당성을 검증한다. 제어평면은 [Fig. 10]의 로컬 테스트베드 제어평면을 KREONET에 연동하여 사용한다.

OpenFlow@KREONET의 슬라이스 구성 실험에서는 <Table 2>의 OpenFlow 기술에서 제공하는 슬라이스 구성 방식이 모두 가능함을 확인하기 위하여 <Table 4>의 슬라이스를 구성하고 동작을 확인한다.

<Table 4> Virtual network / slice configuration results on OpenFlow@KREONET

①	A slice can be configured by all traffic on OpenFlow@KREONET.
②	A slice can be configured by all I/Fs of Busan and Seoul OpenFlow switches.
③	Slices can be configured by MAC addresses of Seoul and Changwon hosts.
④	Slices can be configured by IP addresses of Seoul and Busan hosts.
⑤	Slices can be configured by network IP addresses / IP address blocks.
⑥	Slices can be configured by transport layer Port numbers.
⑦	Slices can be configured by protocol types.

본 논문에서는 OpenFlow@KREONET의 데이터 전송 성능을 검증하기 위하여 <Table 5>의 실험을 수행한다.

<Table 5> Data transmission results through slice(s) of OpenFlow@KREONET (KJ-Kwangju, CW-Changwon, SL-Seoul, DJ-Deajeon)

Test	Slice Configuration with	Sending [bps]	Receiving [bps]
①	All I/Fs of KJ, CW, SL switches	KJ→SL 1G CW→SL 1G	KJ→SL ≈ 450M CW→SL ≈ 450M
②	All I/Fs of KJ, CW, SL switches	SL→KJ 1G SL→CW 1G	SL→KJ ≈ 450M SL→CW ≈ 450M
③	All I/Fs of SL, CW, DJ, KJ switches	SL→KJ 1G DJ→CW 1G	SL→KJ ≈ 850M DJ→CW ≈ 850M
④	All I/Fs of SL, KJ switches	SL→KJ 1G	SL→KJ ≈ 850M
	All I/Fs of DJ, CW, SL switches	DJ→CW 1G	DJ→CW ≈ 850M

실험 ①의 경우 매트릭스 인터넷 스위치와 서울 스위치 사이 1Gbps 링크에서 광주와 창원에서 보낸 트래픽 각각 1Gbps의 병목이 발생한다. 이는 실험 ②의 경우에도 마찬가지로 발생하는 현상이다. 이러한 이유는 [Fig. 16]의

OpenFlow@KREONET이 매트릭스 인터넷 스위치를 통해 각 OpenFlow 스위치가 스타 형태로 구성되어 있기 때문에 병목구간이 생기기 때문이다. 하지만 실험 ③에서 서울, 광주 구간과 대전, 창원 구간은 동일한 매트릭스 인터넷 링크를 공유하지 않기 때문에 병목이 발생하지 않는다. 실험 ④은 실험 ③과 같은 실험을 두 개의 슬라이스를 구성하여 결과를 나타내고 있다. 데이터 전송 실험 결과 현재의 OpenFlow@KREONET의 스위치를 메쉬형으로 재구성하도록 할 계획이다.

6. 결론

국내 대표 첨단연구망인 KREONET에서는 고도화 방안으로 SDN/OpenFlow 기술을 이용한 가상 네트워크 서비스 인프라를 구축하고자 한다. SDN/OpenFlow 기술은 최근 전 세계적으로 미래 인터넷을 위한 대표적인 네트워크 모델로 주목받고 있다. 본 논문에서는 먼저 KREONET의 현재 사용자 특성을 분석하여 가상 네트워크 서비스 요구사항을 도출하였다. 그리고 이를 만족시킬 수 있는 KREONET OpenFlow 기반 가상 네트워크 프레임워크를 제시하고 단계적 구축모델을 정의하였다. 마지막으로 로컬 테스트베드 및 OpenFlow@KREONET 네트워크에서 OpenFlow 가상 네트워크 서비스의 타당성을 검증함으로써 제시한 가상 서비스 제공 방안이 KREONET에서 적용 가능함을 확인하였다.

ACKNOWLEDGMENTS

This research was supported by the Building and Operation KREONET and advancement of Service of the Korea Institute of Science and Technology Information (KISTI) funded by the Ministry of Science, ICT & Future Planning.

REFERENCES

- [1] A. Lara, A. Kolasani, B. Ramamurthy, "Network Innovation using OpenFlow: A Survey," *IEEE*

- Communications Surveys & Tutorials*, Vol. 16, Issue 1, pp. 493-512, 2014.
- [2] Pang-Wei Tsai, Yu-Ting Lai, Pei-Wen Cheng, Chu-Sing Yang, Mon-Yen Luo, "Design and develop an OpenFlow Testbed within virtualized architecture," in Proc. of APNOMS 2013, pp. 1-3, 2013.
- [3] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, J. Turner, "OpenFlow: Enabling Innovation in Campus Networks," OpenFlow White Paper, Stanford Univ., 2009.
- [4] R. Sherwood, G. Gibb, K. Yap, G. Appenzeller, M. Casado, N. McKeown, G. Parulkar, "FlowVisor: A Network Virtualization Layer," OpenFlow Technical Report 2009-01, Stanford Univ., 2009.
- [5] E. Salvadori, R. Doriguzzi Corin, A. Broglio, M. Gerola, "Generalizing Virtual Network Topologies in OpenFlow-Based Networks," in Proc. of GLOBECOM 2011, pp. 1-6, 2011.
- [6] S. Azodolmolky, R. Nejabati, Shuping Peng, A. Hammad, M.P. Channegowda, N. Efstathiou, A. Autenrieth, P. Kaczmarek, D. Simeonidou, "Optical FlowVisor: An OpenFlow-based optical network virtualization approach," in Proc. of OFC/NFOEC 2012, pp. 1-3, 2012.
- [6] B. Sonkoly, A. Gulyas, F. Nemeth, J. Czentye, K. Kurucz, B. Novak, G. Vaszun, "OpenFlow Virtualization Framework with Advanced Capabilities," in Proc. of EWSDN 2012, pp. 18-23, 2012.
- [7] R. Doriguzzi Corin, M. Gerola, R. Riggio, F. De Pellegrini, E. Salvadori, "VeRTIGO: Network Virtualization and Beyond," in Proc. of EWSDN 2012, pp. 24-29, 2012.
- [8] N. B. Melazzi, A. Detti, G. Mazza, G. Morabito, S. Salsano, L. Veltri, "An OpenFlow-based testbed for information centric networking," in Proc. of FutureNetW 2012, pp. 1-9, 2012.
- [9] R. Narisetty, L. Dane, A. Malishevskiy, D. Gurkan, S. Bailey, S. Narayan, S. Mysore, "OpenFlow Configuration Protocol: Implementation for the of Management Plane," in Proc. of GREE 2013, pp. 66-67, 2013.
- [10] P. Calyam, S. Rajagopalan, A. Selvadurai, S. Mohan, A. Venkataraman, A. Berryman, R. Ramnath, "Leveraging OpenFlow for resource placement of virtual desktop cloud applications," in Proc. of IM 2013. pp. 311-319, 2013.

석 승 준(Seok, Seung-Joon)

- 2003년 2월 : 고려대학교 대학원 전자공학과 졸업(공학박사)
- 2004년 3월 ~ 현재 : 경남대학교 컴퓨터공학부 교수
- 관심분야 : 무선인터넷 프로토콜, 미래 인터넷, USN 프로토콜 etc.
- E-Mail : sjseok@kyungnam.ac.kr

정 현 욱(Jeong, Hyeonuk)

- 2014년 2월 : 경남대학교 컴퓨터공학과 졸업 (공학사)
- 관심분야 : 미래 인터넷
- E-Mail : vktod25@gmail.com