

ISBN 978-89-6211-472-0 98560

OpenFlow 기반 네트워크 가상화 서비스 프레임워크 설계서

일자: 2009년 11월 12일

부서: 슈퍼컴퓨팅본부 기반기술개발실

제출자: 홍원택, 문정훈

{wthong, jhmoon}@kisti.re.kr



한국과학기술정보연구원
Korea Institute of Science and Technology Information

305-806 대전광역시 유성구 어은동 52번지
TEL (042)869-0676 / FAX (042)869-0679
www.kisti.re.kr

목차

1. 서론
2. OpenFlow 기반의 가상화 기법
3. 단계별 가상화 테스트베드 프로토타입 구축
4. 네트워크 가상화 서비스 프레임워크 설계
5. 결론
6. 참고문헌

1. 서론

네트워크 가상화의 요구사항을 통하여서 네트워크 연구자들을 대상으로 기존 실험실 위주의 연구 환경을 실제 네트워크 환경으로 확대하고, 스위치, 라우터 등의 네트워크 장비에 대해서도 연구자의 환경으로 구성 또는 충분한 제어 권한을 통하여 효과적인 연구를 수행할 수 있게 한다. 로컬 환경의 한계를 벗어나 백본 환경에서의 실제적인 테스트를 통해 연구 성과 및 효과를 증대시킬 수 있게 된다. 따라서 이러한 획기적인 네트워크 테스트베드를 제공할 수 있는 네트워크 가상화 핵심 기술과 가상화 환경 하에서 효과적인 연구 수행을 위해 트래픽 모니터링, 가시화, 아카이브 등과 같은 연구자를 위한 서비스와 가상화 네트워크 관리자를 위한 서비스 기술의 개발이 필요하다.

- 네트워크 가상화 환경 사용자 : 네트워크 연구자
 - . 실제 네트워크 환경에서 연구가 필요한 네트워크 연구자
 - . 실제 운영중인 네트워크 장비의 활용/제어가 필요한 네트워크 연구자
 - . 실제 네트워크와 네트워크 장비의 임의 수정 및 재구성이 필요한 네트워크 연구자

네트워크 가상화 기술의 사용자는 네트워크 연구자로서 실험망 환경이 아닌 실제 네트워크 상에서의 테스트베드에 대한 요구가 증가하고 있다. 이러한 요구는 새로운 네트워크 기술들이 실험실의 제한적인 환경 하에서의 제약으로 인해 효과적인 기술임에도 불구하고 기존 망 환경에 적용이 되지 않을 뿐만 아니라 실제 환경에서의 검증도 되지 않고 있기 때문이다.

따라서 실제 환경 하에서의 실제 운영측면의 테스트는 기존 네트워크 환경의 높은 진입장벽으로 인해 지금까지는 시뮬레이션 또는 실험 망 규모의 환경에서 테스트가 진행되었지만, 가상화 기술을 통하여 백본 네트워크 환경을 포함한 대륙 간 네트워크 실험도 가능하게 되었다. 이러한 환경을 가능케 하는 기술들로서 지금까지 제안되고 진행되고 있는 프로젝트들이 PlanetLab, VINI 등과 같은 기술들이며, 특히 백본 네트워크에 대한 권한이 있는 KREONET 환경에서는 OpenFlow Switch 환경이 효과적인 가상화 기술이다.

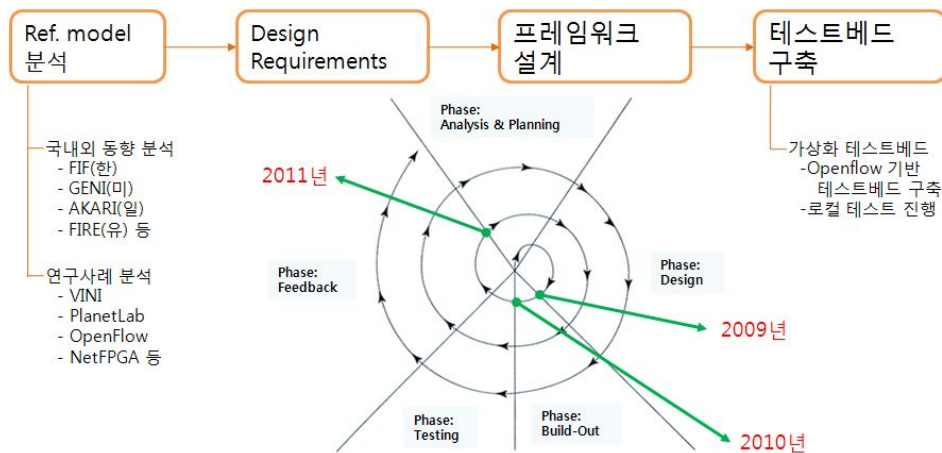


Figure 1 Network Virtualization Roadmap in KISTI

네트워크 가상화 테스트베드 제공은 위의 그림과 같은 로드맵을 통하여 진행되는데 2009년도의 첫 번째 Spiral-1에서는 국내외 관련 기술의 동향분석과 연구사례 분석하여 모델을 정립하고, 설계 요구사항을 분석, 도출하여 프레임워크를 설계한다. 또한 네트워크 가상화 테스트베드 제공을 위하여 OpenFlow 기반의 로컬 테스트베드를 구축한다.

2. OpenFlow 기반의 가상화 기법

기본적인 OpenFlow기반의 네트워크 구성은 다음 그림과 같이 구성된다. 하나의 Controller를 중심으로 OpenFlow switch들의 조합으로 이루어지는데 OpenFlow switch들은 flow-table의 flow-entry에 따라 들어오는 패킷들에 대해서 특정 action를 통해 라우팅을 하게 되고 flow-table에서 정의되어 있지 않는 패킷들은 Controller로 보내져서 그 패킷에 대한 조치를 취하게 된다. 이러한 Centralized Control 기능을 가지고 연구자들에게 새로운 기능들을 실험할 수 있는 환경과 제어를 제공한다. Controller와 OpenFlow switch간의 기본적인 메시지 전송을 통하여 다음과 같은 기능들을 수행하게 된다.

- Controller-to-switch message
 - . Configuration the switch
 - . Managing the flow table
 - . Figuring out switch capabilities
- Asynchronous message
 - . From switch to controller
 - . Report event (switch state, network state, etc)
- Symmetric message
 - . Sent in either direction (Hello, Echo, etc)
 - . Diagnose problems in switch-Controller connection

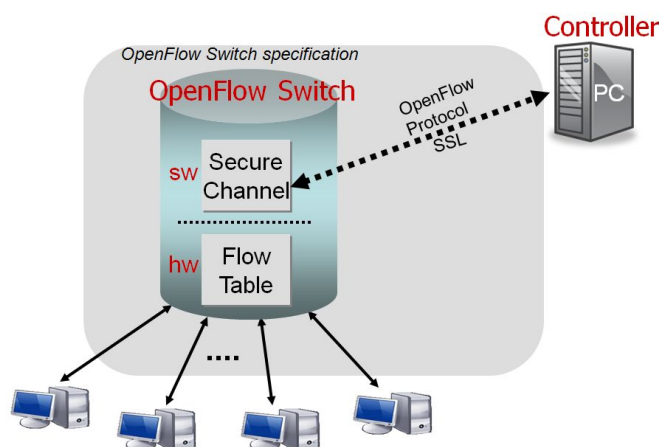


Figure 2 Basic concept of OpenFlow switch and Controller

기존의 OpenFlow switch환경에서 Controller의 성능은 한 번에 하나의 Controller만 작동

하는 형태였다. 다수 사용자가 동시에 OpenFlow switch환경의 네트워크를 사용할 수 있도록 개발된 것이 FlowVisor이다. 또 FlowVisor는 NOX의 S/W 구조의 복잡성과 성능의 문제등으로 인해 일종의 Controller로서 개발되었다.

FlowVisor는 네트워크 자원들을 flow 기반으로 slicing해서 사용하는 것으로서 다수의 Controller가 동시에 작업을 할 수 있게 되었다. 또한 FlowVisor가 개발되면서 본격적으로 OpenFlow switch기반의 네트워크 환경도 flow 기반의 slicing을 통한 가상화 환경의 구현이 가능하게 되었다. 따라서 FlowVisor의 목적은 다수의 연구자가 동시에 OpenFlow switch환경에 접근하여 자신들의 환경에 따른 연구 및 실험을 가능케 한다. 아래의 그림은 FlowVisor의 기본 구조도이다.

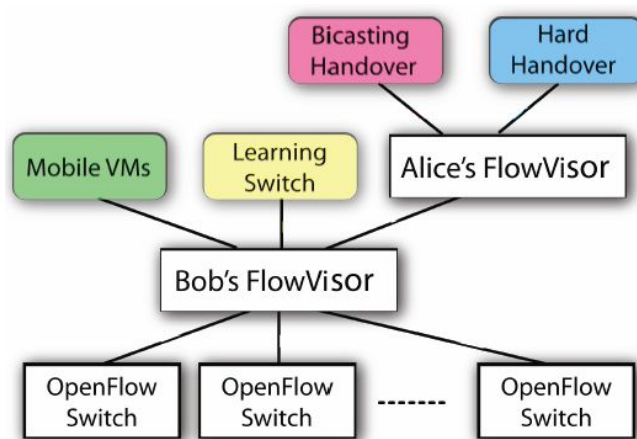


Figure 3 Multiple Controllers based on FlowVisor

위의 그림에서 Bob's FlowVisor를 통해 하단에 OpenFlow Switch 네트워크가 관리가 되고 Bob's FlowVisor는 Alice's FlowVisor, Mobile VM Controller, Learning Switch Controller를 지원하고 있다. Alice's FlowVisor는 Bicasting Handover Controller와 Hard Handover Controller를 지원하면서 Bob's FlowVisor를 통해 OpenFlow switch 네트워크를 활용하고 있다.

FlowVisor를 활용하여 Plug-n-Serve, OpenPipes, OpenRoads, Aggregation 등의 기술들이 구현되어 있다. 이러한 각각의 기술들은 OpenFlow switch 기반 하에서 하나의 Controller상에서 구현되는 것들로서 FlowVisor를 통하여 동시에 하나의 OpenFlow switch 네트워크상에서 작동이 가능하게 된다.

- . Plug-n-Serve : Flow 기반 네트워크 상의 Load-balancing 기능
- . OpenPipes : OpenFlow switch 기반으로 구성된 네트워크 상에서 H/W설계를 line rate으로 할 수 있는 기능 (경제적으로 우수)
- . OpenRoad : WiFi-WiMAX사이에서 loss가 없는 Handover 기능
- . Aggregation : 트래픽 관리를 위해 동적으로 flow를 모으는 기능

KREONET 환경에서도 OpenFlow switch 환경을 FlowVisor를 통해 구현함으로써 네트워크 가상화 환경으로서 다수의 사용자에게 OpenFlow switch 환경의 테스트베드를 제공할 수 있다.

3. 단계별 가상화 테스트베드 프로토타입 구축

3.1 1단계 : OF/NOX 기반 테스트베드 구축

네트워크 가상화 1단계로서 OpenFlow/NOX 기반의 테스트베드의 구축이다. NOX는 Network Operation System의 기본 개념을 수용한 Controller의 일종으로 연구자들의 원하는 시나리오를 application으로 작성하여 실행 시키면 해당 시나리오대로 OpenFlow switch를 제어한다.

따라서 네트워크 가상화 프로젝트에서 1단계로서 미래인터넷 및 네트워크 가상화 기술에 대한 분석과 선진국의 기술동향을 분석하고, 이에 따라 기본적인 OpenFlow 기반의 네트워크 가상화 테스트베드 구축한다. 아래 그림은 기본적인 테스트를 위한 OpenFlow 기반의 네트워크 가상화 테스트베드이다.

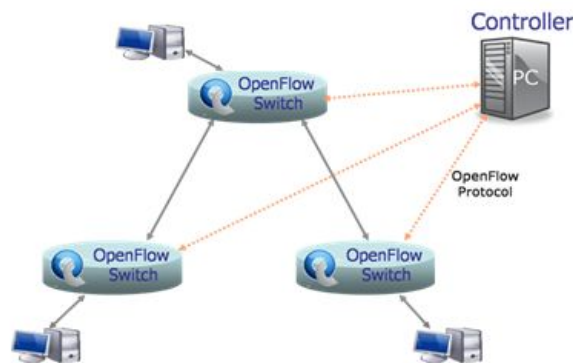


Figure 4 Basic OpenFlow network

3.2 2단계 : OF/NOX/FV 기반 Slicing 제공

네트워크 가상화 테스트베드 제공의 2단계로서 OpenFlow기술에서 언급했듯이 FlowVisor 기능의 추가로 OpenFlow/NOX/FlowVisor 기반의 네트워크 가상화 테스트베드를 구축한다. 이렇게 구축할 경우 실제적으로 다수의 Controller의 수용이 가능하며, FlowVisor를 통하여 flow 기반의 slicing이 가능함으로 다수의 Controller의 수용이 가능해진다. 따라서 FlowVisor 기반의 Slicing을 통한 본격적인 네트워크 가상화 테스트베드로서 로컬 환경에서 국내 연구망 백본을 활용한 테스트베드 범위를 확대한다. 아래 그림은 FlowVisor를 통한 국내 거점 지역중심의 OpenFlow 기반 네트워크 가상화 환경 구성도이다.

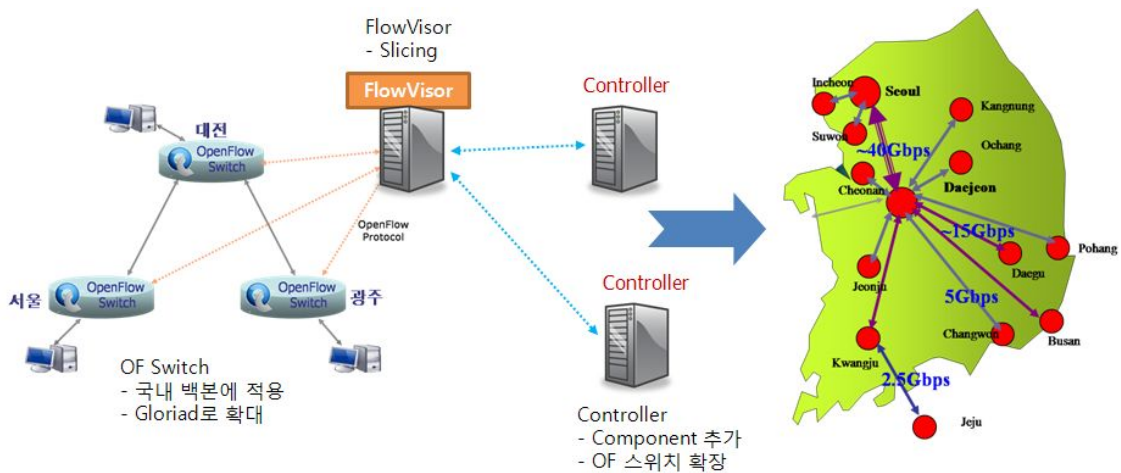


Figure 5 OpenFlow switch and FlowVisor deployment for KREONET

3.3 3단계 : 국내의 연구망 기반의 네트워크 가상화 시범 서비스

네트워크 가상화 시범 서비스 단계로서 2단계의 OpenFlow/FlowVisor기반의 네트워크 가상화 테스트베드 환경에 NMS (Network Management System) 기능의 적용으로 사용자, 운영자 측면의 특화된 서비스 기능을 구현하고, 가시화 기능을 구현하여 효과적인 실험환경을 제공한다. 멀티 도메인 환경에서의 효과적인 운영을 위하여 Bridging 기술을 활용하여 네트워크 가상화 서비스 범위를 확대한다. Bridging 기술로는 Mac-in-IP, Overlay 등의 기술들을 활용할 수 있다. 또한 GLORIAD 망을 활용하여 국제망 환경에서의 네트워크 가상화 테스트베드가 제공된다.

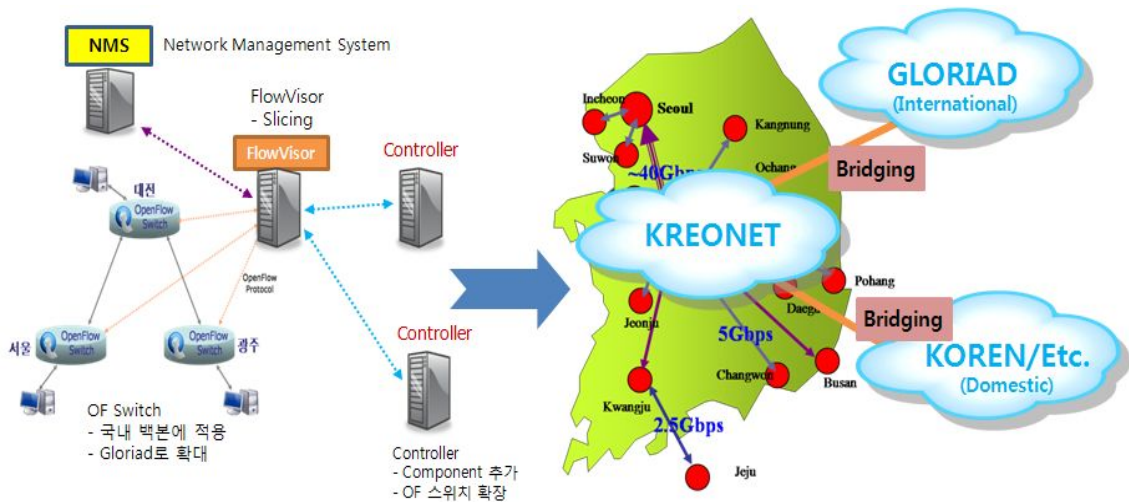


Figure 6 OpenFlow switch and FlowVisor deployment for KREONET and GLORIAD

4. 네트워크 가상화 서비스프레임워크 설계

OpenFlow 스위치로 구성된 연구망에서 네트워크 연구자들은 각자의 실험 시나리오를 반영하는 Controller를 개발하여 원하는 시간에 적용하기 원할 것이다. 이러한 요구를 동시에 수용하기 위해 OpenFlow 망에서의 자원 슬라이싱이 요구되고, 궁극적으로 망의 슬라이스들을 모니터링하기 위한 서비스 프레임워크가 필요하다.

현재, OpenFlow 스위치들로 구성된 망을 기반으로 OpenFlow 선도 연구 그룹들이 여러 가지 실험 시나리오를 기반으로 Controller를 개발하여 적용하고 있고, 이러한 과정 중에 필요한 모니터링 위한 GUI 툴들을 병행하여 개발하고 있다. 이러한 모니터링 툴 개발에 공통적으로 많이 이용하고 있는 것이 ENVI(Extensible Network Visualization & Control) 프레임워크이다.

ENVI는 OpenFlow 관련 GUI 툴을 개발하는 데 기반을 제공하는 확장 가능한 플랫폼이다. 네트워크 토폴로지 및 관련 정보들을 질의할 수 있는 인터페이스를 제공한다. 현재 ENVI는 Java 라이브러리, Javadoc 형태로 공개되어 있고, OpenFlow protocol의 메시지를 기반으로 하여 확장의 여지가 많은 상태이다.

다음 그림은 ENVI 플랫폼의 구조를 보여준다. 서버는 OpenFlow 스위치로 구성된 망의 접근 및 제어와 관련된 기능을 제공하고, "Dashboard"라 불리는 클라이언트는 OpenFlow 망을 가시화시키고, 제어하기 위한 인터페이스를 제공한다. 일반적인 네트워크 모니터링에서 많이 사용하는 클라이언트-서버 구조를 따랐으며, 데이터의 수집과 가시화를 각각 분리시켰다.

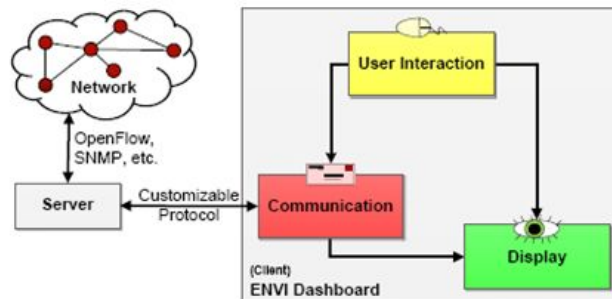


Figure 7 ENVI framework

다음 그림들은 ENVI 플랫폼에서 제공하는 Java 전체 라이브러리 및 클래스의 일부를 보여 준다. 전체 라이브러리는 크게 OpenFlow protocol 메시지들을 처리하는 패키지와 관련 데이터를 가지화시키는 패키지로 나누어진다. 이들 패키지들 중에서 "org.openflow.gui.net.protocol" 패키지 내에 Flow, Link, Node에 대한 정의와 추가, 삭제에 관련된 클래스들이 정의되어 있다.

Packages	
org.openflow.gui	Core components of the OpenFlow GUI.
org.openflow.gui.chart	Extensions to JFreeChart.
org.openflow.gui.drawables	Objects which can be drawn by the Pan-Zoom GUI.
org.openflow.gui.net	Provides classes to manage connections and protocols.
org.openflow.gui.net.protocol	Core messages for communicating with the OpenFlow GUI backend.
org.openflow.gui.net.protocol.auth	Authentication messages.
org.openflow.gui.stats	Defines statistics-processing classes.
org.openflow.protocol	Defines messages in the OpenFlow protocol.
org.openflow.util	Defines basic structures throughout the project.
org.openflow.util.string	Utility classes for string processing.
org.pzgui	Provides a GUI system which supports multiple windows which can be panned and zoomed.
org.pzgui.icon	Provides a framework for drawing and efficiently managing resizable graphics.
org.pzgui.layout	Extends the base Pan-Zoom GUI system to provide automatic node layout.
org.pzgui.math	Defines some basic math functions and data structures.
org.tame	Provides a multi-knob slider component.

Figure 8 ENVI java packages

Class Summary	
Flow	Structure to describe a flow.
FlowHop	A hop that a flow goes through.
FlowsAdd	Flow(s) added message.
FlowsDel	Flow(s) deleted message.
FlowsList	A list of flows.
Link	Structure to specify a link.
LinksAdd	Link(s) added message.
LinksDel	Link(s) deleted message.
LinksList	A list of links.
Node	Structure to specify a node.
NodesAdd	Node(s) added message.
NodesDel	Switch(es) deleted message.
NodesList	A list of nodes.
OFGMessage	Header for OpenFlow GUI protocol messages.
PollStart	OFGMessage which tells the backend to periodically send a message for us.
PollStop	Tells the backend to stop polling for a given message.
Request	A generic request.
RequestLinks	A request regarding links.
StatsHeader	Stats related request or reply.
SwitchDescriptionRequest	Request for info describing a switch.

Figure 9 Core messages (org.openflow.gui.net.protocol)

OpenFlow 망과 클라이언트 사이에 존재하는 서버는 “Standalone” 라이브러리 형태와 NOX-based Module 형태로 구현될 수 있고, 각각의 특징은 다음과 같다.

- Standalone library

ENVI 플랫폼의 소스 코드를 포함하여 제공되는 "lightweight & extensible"한 구현물로서 클라이언트의 주요 기능을 검증하는데 테스트 서버로 사용될 수 있다. Standalone 라이브러리는 처음부터 ENVI 플랫폼의 구현을 1차 목표로 하여 구현되었으므로, 각각의 Controller와의 연동을 위해서는 통합되어야 할 것이다.

- NOX-based module

NOX 플랫폼 기반의 Controller를 위한 ENVI 서버의 구현물로서 LAVI라는 이름으로 알려져 있다. LAVI 자체가 NOX 기반의 모듈 형태로 제공되므로, NOX를 기반으로 구현된 Controller와 연동을 하는 경우에 적합한 선택이 될 것이다.

다음 그림은 ENVI 플랫폼을 기반으로 확장되어 OpenFlow 망을 모니터링하는 Network monitor GUI와 ElasticTree의 GUI를 캡처하여 보여준다. Network monitor는 ENVI 플랫폼의 많은 확장 없이 간단한 모니터링 기능을 제공한다.

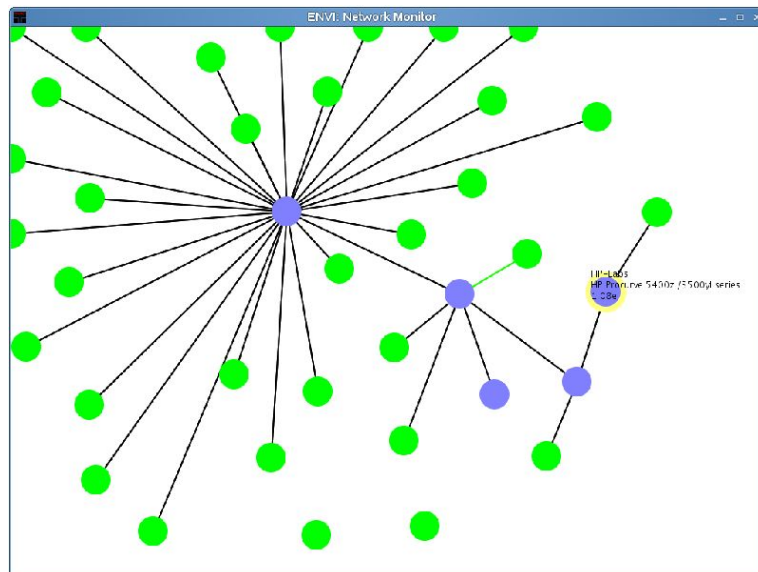


Figure 10 Network monitor GUI based on ENVI

ElasticTree는 다음 그림에서처럼 데이터센터 네트워크의 전력 최적화를 위해 만들어진 Controller로서 단순한 모니터링 기능 외에 edge, aggregation, core 단에서의 트래픽 제어를 위한 제어 기능도 함께 제공한다. 이러한 ElasticTree의 기능 확장은 ENVI 플랫폼을 기반으로 한 모니터링 외에 제어를 위한 확장 시 참고 모델로 도움이 될 것이다.

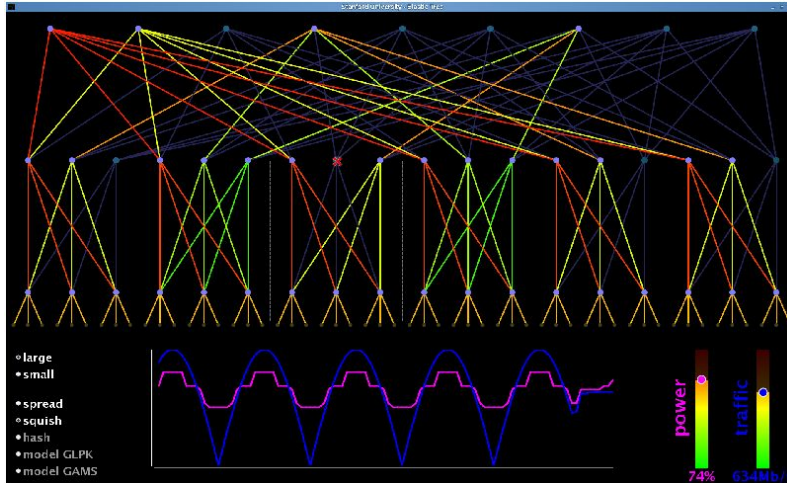


Figure 11 ElasticTree GUI based on ENVI

연구망에서는 OF 망 자원의 슬라이싱을 위해 FlowVisor가 사용될 예정이며, 기본적으로 FlowVisor에서 제공하는 GUI의 많은 부분을 도입하여 확장할 예정이다. FlowVisor GUI는 슬라이스별 토폴로지의 가시화, 복수 슬라이스들 간의 조합된 가시화를 위한 간단한 제어 기능 등을 다음 그림에서처럼 제공한다.

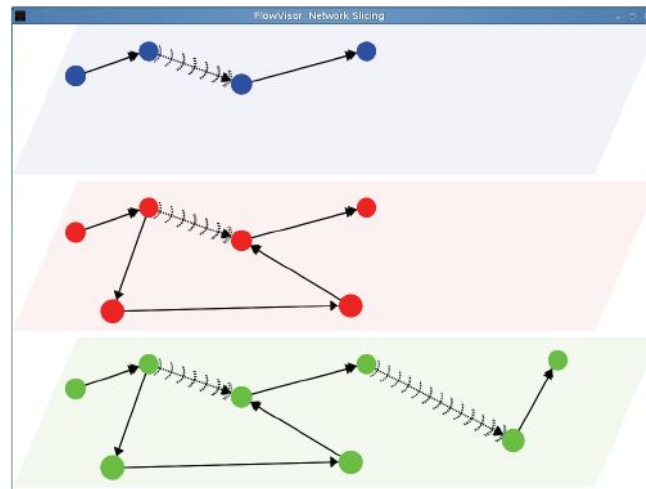


Figure 12 FlowVisor GUI

현재, 연구망에서 고려하고 있는 OpenFlow switch로 구성된 망 기반의 가상화 서비스 프레임워크는 다음 그림에서와 같다. 주요 기능으로는 Controller 등록 기능, 슬라이스 모니터링 기능, 네트워크 토폴로지/링크 모니터링 등이다.

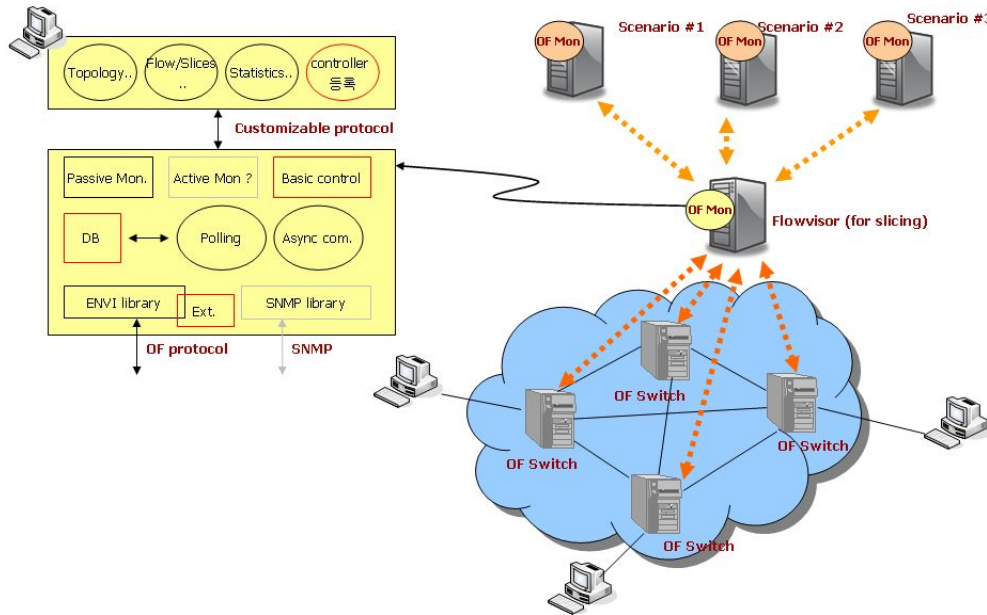


Figure 13 Network virtualization service framework

연구망을 위한 OF 망 기반의 서비스 프레임워크는 크게 서버 부분과 클라이언트 부분으로 나눠 정리할 수 있다.

- 클라이언트 주요 기능

. 토폴로지 모니터링

OpenFlow 스위치들로 구성된 물리적 토폴로지를 제공하는 기능으로, OpenFlow 스위치, 스위치간의 연결(links), 각 스위치에서 유지하는 flow 정보들을 제공한다. 이들 중에서 flow 정보는 flow 테이블상의 flow entry expiration에 따라 동적으로 변하는 정보로 분류할 수 있다.

. 슬라이스 모니터링

기본적으로 FlowVisor GUI에서는 슬라이스별 토폴로지의 가시화 기능을 제공한다. 현재, EVNI 플랫폼에서는 Flow, Link, Node 중심의 표현을 위한 라이브러리가 정의되어 있는 상태이고, 이들 중에서 Flow 및 link 단위의 정보를 토대로 각 Controller 별로 할당된 슬라이스들을 표현하는 새로운 방법을 고안할 예정이다.

. Network statistics 정보 제공

OpenFlow 스위치 스펙에서는 다음 표와 같이 OpenFlow 망의 statistics 정보를 "Read-State" 메시지를 이용하여 request/reply할 수 있다. 이러한 OpenFlow 메시지들은

서버를 통하여 클라이언트로 전송 및 가공하여 네트워크 성능 파라미터를 도출이 가능하다.

Table 1 Request/Reply parameters in ofp_stats_types

종류	request	reply
OFPST_DESC	empty	struct ofp_desc_stats
OFPST_FLOW	struct ofp_flow_stats_request	an array of struct ofp_flow_stats
OFPST_AGGREGATE	struct ofp_aggregate_stats_request	struct ofp_aggregate_stats_reply
OFPST_TABLE	empty	an array of struct ofp_table_stats
OFPST_PORT	empty	an array of struct ofp_port_stats

. Controller 등록 기능

OpenFlow 망을 동시에 이용하는 다수의 Controller들을 등록하여 서비스 프레임워크에서 통합적으로 관리할 수 있는 기능이다. Controller의 위치 정보 및 각 Controller 별 슬라이싱 정책을 반영하여 서버 상에서 슬라이스를 할당할 때에 기초 자료로 활용될 수 있도록 한다.

- 서버 주요 모듈

. ENVI 라이브러리 연동 및 확장 모듈

대부분의 OpenFlow 프로토콜의 메시지를 처리하는 부분으로 모니터링과 관련된 많은 부분의 기본 정보들을 제공하는 역할을 한다. ENVI 라이브러리와 연동을 위해 필요한 모듈로써 클라이언트에서의 기능 추가로 인한 ENVI 라이브러리 확장에 따라 변화가 불가피한 부분이다.

. SNMP 라이브러리

대부분의 configuration 및 모니터링 정보는 OpenFlow 프로토콜의 메시지 해석을 기반으로 이루어지지만, OpenFlow 스위치에서 제공하는 표준 MIB을 통해 관리 정보를 얻고자 할 경우 필요한 정보를 수집 및 처리할 수 있는 모듈이다. OpenFlow-enabled된 상용 장비와 연동 시 필요할 것이다.

. Polling/Async 통신 모듈

기본적으로 OpenFlow와 Controller 사이의 연결은 메시지 타입 중 symmetric 메시지인 "Echo" 메시지를 이용하여 연결 상태를 주기적으로 파악할 수 있으나, OpenFlow 스위치들 간의 연결은 정의된 메시지가 따로 존재하지 않는다. 이러한 경우, "Packet In/Out" 메시지를 주기적으로 주고받음으로써 연결 상태를 파악할 수 있다. 반면, 각 스위치에서 유지하는 flow entry를 클라이언트 GUI에서 동적으로 제공하는 경우, asynchronous 메시지인 "Flow Expiration" 메시지를 통해 각 OpenFlow 스위치들로부터 이벤트 발생 시 정보를 얻을 수 있다.

. Passive/Active 모니터링 모듈

일반적인 모니터링의 기법들 중에서 Passive/Active 모니터링을 선택적으로 사용할 수 있는데, Controller의 시나리오에 따라 필요한 모니터링 방법의 선택을 반영할 수 있는 모듈이다.

. 기본 제어 및 데이터베이스 처리 모듈

클라이언트로부터 입력 받은 Controller의 등록 정보를 저장하고, 각 Controller 별로 일정 기간 이상 network statistics 정보들 중에서 과거 정보를 유지하여 모니터링 하고자 할 경우, 일정 기간 동안 관리 정보를 유지하기 위한 모듈이다. 수집 대상이 되는 정보는 Controller에서 관심을 갖는 파라미터를 선택하여 반영할 수 있도록 한다.

그 외에, 종합적으로 OpenFlow 스펙에서 정의된 메시지들 중에서 모니터링에 필요한 메시지들은 다음 표에서와 같이 정리할 수 있다.

Table 2 OpenFlow messages for monitoring

메시지 종류	서브 타입	의미
Controller-to-Switch	Features	스위치에서 지원되는 Capabilities (포트, 인터페이스에 대한 물리적 정보)
	Read-State	스위치의 flow-tables, 포트, flow entries들에 대한 statistics 정보
Asynchronous	Flow Expiration	flow table로 부터의 flow entry 삭제 관련
	Port-status	port 설정 상태의 변화 시
Symmetric	Echo	Controller와 switch 간의 지연, 대역폭, 연결 관련

5. 결론

백본 망에서의 네트워크 가상화를 지원하기 위해 KREONET에서는 OpenFlow 스위치로 구성된 테스트베드 제공을 계획하고 있다. 이러한 OpenFlow 스위치 기반의 망에서 FlowVisor를 이용한 네트워크 슬라이싱 및 다수의 Controller들을 동시에 수용하는 통합적인 서비스 프레임워크 설계를 진행해 오고 있다.

테스트베드 제공 측면에서는 OpenFlow switch/NOX 기반의 1단계, OpenFlow switch/NOX/FlowVisor 기반 네트워크 자원 슬라이싱 제공의 2단계, 국내외 연구망 기반의 확장된 네트워크 가상화 시범 서비스 3단계로 구성된 단계별 목표를 진행할 예정이다. 또한, 이러한 테스트베드 상의 효율적인 운영을 위한 통합적인 서비스 프레임워크를 제공하기 위해 ENVI 플랫폼을 기반으로 네트워크 토폴로지/링크/슬라이스 모니터링 및 다수의 Controller 등록 기능 등을 포함하는 서비스 프레임워크를 설계 진행 중에 있다.

6. 참고문헌

- [1] FlowVisor. <http://www.openflowswitch.org/wk/index.php/FlowVisor>
- [2] Rob Sherwood, Glen Gibby, Kok-Kiong Yapy, Guido Appenzellery, Martin Casado, Nick McKeowny, and Guru Parulkary, FlowVisor: A Network Virtualization Layer, Technical Report
- [3] Rob Sherwood, Michael Chan, Glen Gibb, Nikhil Handigol, Te-Yuan Huang, Peyman Kazemian, Masayoshi Kobayashi, David Underhill, Kok-Kiong Yap, Guido Appenzeller, and Nick McKeown, Carving Research Slices Out of Your Production Networks with OpenFlow, ACM SIGCOMM Demo, Aug 2009
- [4] D. Underhill. An Extensible Network Visualization and Control Framework. Master's thesis, Stanford University, May 2009
- [5] D. Underhill et al. Network monitor. <http://www.openflowswitch.org/wp/gui>
- [6] ENVI framework, <http://www.openflowswitch.org/wp/gui/>
- [7] The OpenFlow Switch Consortium. <http://www.openflowswitch.org>
- [8] OpenFlow Switch Specification 0.8.9, <http://www.openflowswitch.org>
- [9] NOX: An OpenFlow Controller. <http://noxrepo.org/wp/>
- [10] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner. OpenFlow: enabling innovation in campus networks, ACM SIGCOMM Computer Communication Review, 38(2):69-74, April 2008
- [11] N. Gude, T. Koponen, J. Pettit, B. Pfaff, M. Casado, N. McKeown, and S. Shenker, NOX: Towards an operating system for networks, ACM SIGCOMM Computer Communication Review, July 2008