

# OpenFlow Network에서의 QoS 포워딩 메커니즘

강민협\*, 강진혁\*, 송왕철\*, 석승준\*\*, 최덕재\*\*\*, 허지완\*\*\*\*

\*제주대학교 컴퓨터 공학과

\*\*경남대학교 컴퓨터 공학과

\*\*\*전남대학교 컴퓨터 공학과

\*\*\*\*국가수리과학연구소 미래인터넷팀

e-mail:[kang-min-hyup@hanmail.net](mailto:kang-min-hyup@hanmail.net)

[jinhyeok.kang@gmail.com](mailto:jinhyeok.kang@gmail.com)

[kingiron@gmail.com](mailto:kingiron@gmail.com)

[sjseok@net.kyungnam.ac.kr](mailto:sjseok@net.kyungnam.ac.kr)

[dchoi@jnu.ac.kr](mailto:dchoi@jnu.ac.kr)

[jeewan@nims.re.kr](mailto:jeewan@nims.re.kr)

## A QoS aware forwarding mechanism in OpenFlow Network

Min-Hyup Kang\*, Jin-Hyeok Kang\*, Wang-Cheol Song\*, Seung-Jun Seok\*\*,  
Doek-Jae Choi\*\*\*, Jee-Wan Huh\*\*\*\*

\*Dept of Computer Engineering, Jeju National University

\*\*Dept of Computer Engineering, KyungNam University

\*\*\*Dept of Computer Engineering, CheonNam National University

\*\*\*\*Future Internet team, National Institute for Mathematical Sciences

### 요 약

최근 현재 인터넷이 가지는 한계를 극복하기 위해 전세계적으로 연구가 되고 있는 미래인터넷 테스트베드 분야에서 QoS(Quality of Service)에 대한 요구가 계속 증가되고 있다. 본 논문에서는 시뮬레이션을 통한 검증과 실용성은 한계와 확장성의 문제가 있기 때문에, 기존 네트워크의 정상적인 서비스를 간섭하지 않고 새로운 네트워킹 기술을 실험할 수 있고, 가장 각광 받고 있는 OpenFlow를 이용하여 실제 상용 스위치인 Pronto 3290을 기반으로 한 테스트 베드를 구성하였다. 구성한 테스트 베드를 통하여 패킷 정보에 따라 Path를 사용자 요구에 맞게 변경, 삭제, 생성 할 수 있는 QoS 메커니즘을 설계하고 사용자가 사용하기 편하도록 웹 기반의 인터페이스를 구현을 통하여 검증할 것이다.

### 1. 서론

지금까지 사용되고 있는 인터넷은 초기의 인터넷 설계 시 고려되지 못했던 다양한 환경을 고려하지 않으면 향후 등장할 새로운 응용 및 서비스를 수용하기 어려운 상황에 처해 있다. 이러한 문제점들은 보안, 관리, 이질성, 이동성, 확장성 및 다양한 무선 통신 환경 등에 대한 대처 능력의 부족을 예로 들 수 있다.

이런 이유로 수 년 전부터 전 세계적으로 새로운 형태의 미래인터넷 구조를 설계하고 핵심 요소들을 개발하며 테스트베드 구축을 하려는 노력이 활발히 진행되고 있다. 특히 그동안 네트워크 연구는 수학적 해석 모델링 방법으로 증명하거나 시뮬레이션을 통해 증명하는 방법이 주로 이용되어 왔다. 하지만 수학적 해석 모델링은 문제를 해결하기 위해 과도한 단순 모델링 가능성이 있다는 한계를 가지고 있으며, 시뮬레이션 모델링은 목적에 맞는 적절한 모델링이 중요한데 실험에 필요한 시스템 behavior에 의한 영향을 설계자가 고려하지 못 할 수 있다. 따라서 OpenFlow[1] 테스트베드를 통해 솔루션의 구현 가능성을

확인할 수 있으며 실제 트래픽 실험을 통해 모델링에 의한 오류 가능성을 미연에 방지 할 수 있는 것들이 가능하다. 또한, 기존 네트워크 아키텍처를 완전히 새롭게 설계 하자는 미래인터넷 연구에서 실험자의 아이디어를 실제 환경과 같이 프로그래머블 한 플랫폼을 이용하여 70년도 부터 계속적으로 요구가 되어 왔던 품질 보장에 대한 요구를 현재의 인터넷 구조 보다 전체적인 네트워크 자원을 보다 효율적으로 관리할 수 있는 OpenFlow Network를 통해 QoS(Quality of Service)[2] 메커니즘을 설계하고 웹 기반의 인터페이스를 통해 동작 과정을 소개하고 4장에서 본 논문의 결론을 기술한다.

### 2. 관련 연구

#### 2.1 OpenFlow Protocol

미래인터넷 연구에 있어서 테스트베드는 기존 네트워크 연구의 경우보다 더욱 중요하다고 할 수 있다. 그 이유는 미래 인터넷 연구가 현재 인터넷의 문제점을 구조적으로 근본적으로 해결하려 하기 때문이다. 구조가 변경된

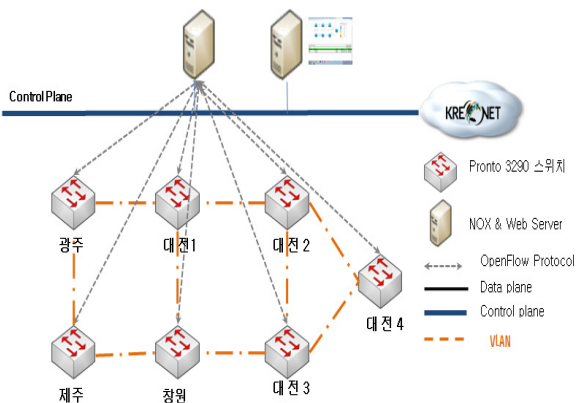
Non-IP[3] 네트워크는 현재로서는 이를 지원하는 시뮬레이션 소프트웨어가 없는 상태이며, 모델링의 증명을 위해서 테스트베드에서의 구현과 실험이 반드시 필요하다. 테스트베드에서의 연구도 시뮬레이션에 의한 연구와 비슷한 문제점이 생길 수 있다. 현재의 네트워크 장비들은 Non-IP를 지원하는 새로운 하드웨어를 제작하여 실제 네트워크와 유사한 네트워크를 만들어야하고, 또한 새로운 하드웨어를 제작하였다 하더라도 표준 인터페이스가 없다면 해당 장비에 따라 서로 다른 사용자 인터페이스를 익혀야 하는 제약이 따른다. 이러한 문제점을 해결하기 위해 연구용 테스트베드는 통일된 표준 인터페이스가 지원되어야 한다. 이를 통하여 연구자들은 자신의 아이디어를 직접 테스트베드에서 구현해 볼 수 있다. 따라서 네트워크 장비를 Control Plane[4] 과 Data Plane[5]으로 분리하여 각 네트워크 벤더들의 장비에 대하여 표준 인터페이스를 제공하고 있다.

### 2.2 Open Programmable Platform

기존의 네트워크는 벤더 프로그래머블 시스템으로써 상용 스위치나 라우터는 고유의 시스템을 가지고 있다. 그러므로 특정 벤더의 시스템을 다룰 수 있는 사람이 다른 벤더의 네트워크 장비를 다루기 위해서는 추가적인 교육이나 학습이 필요하다. 다시 말하면 오픈 플랫폼이 아니며 이러한 시스템은 각 벤더의 중요한 기밀사항이기도 하다. 하지만, 미래 인터넷 연구에서는 연구자들이 직접 네트워크를 구성하고 조작할 수 있는 사용자 프로그래머블 플랫폼이 필요하다. 또한 이 플랫폼은 아이디어의 구현이 용이한 모듈러 구조로 되어 있어야 하며, 적절한 개발 환경과 API 제공이 필요하다. 이를 위하여 오픈 소스가 적합하며 하드웨어 환경 또한 저렴한 것으로 손쉽게 구축 가능하여야 한다. 특히 사용자의 필요에 의해 플랫폼 자체의 수정과 개선이 허용을 통하여 연구자들의 아이디어 환경이 구성되어야 할 것이다.

## 3. 구현

### 3.1 QoS 메커니즘을 위한 OpenFlow 테스트베드



[그림 1] KOREONET 망 테스트베드

[그림 1]은 QoS 메커니즘을 위한 OpenFlow 테스트베드를 나타낸 그림이다. KOREONET[6] 망에 1대의 OpenFlow 컨트롤러, 7대의 Pronto 3290 스위치, 웹서버와 각 스위치에 호스트를 구성하고, 컨트롤러와 스위치에 연결하여 각 포트와 호스트에 IP 주소를 할당하였고, 스위치간은 VLAN[7]으로 연결하였다. 원활한 동작 테스트를 위해 같은 서브네트워크 IP 주소를 같은 PC 인터페이스에 할당하였다. 그리고 사용자가 웹을 통해 OpenFlow 네트워크 토폴로지를 네트워크 웨더맵[8] 형식으로 실시간 트래픽을 모니터링 할 수 있게 제공해주고, Path를 조작 가능하게 할 수 있는 웹서버를 구성하였다.

#### 3.1.1. NOX Controller

NOX 컨트롤러는 기업과 홈 네트워크에 대한 관리 기능을 개발 하기 위한 개방형 플랫폼으로, 상용 하드웨어에서도 동작하며, 기가 비트 속도로 대규모 네트워크를 제어할 수 있는 프로그램을 제공한다. 또한 NOX 컨트롤러는 가장 핵심적인 일종의 서버로써 OpenFlow의 모든 제어 역할을 담당하고, 연구자들이 원하는 환경을 위해 프로그램이 가능한 환경을 제공하고 테스트베드 토폴로지 정보 수집의 기능을 할 수 있게 한다. 또한, OpenFlow 네트워크 구조에서 NOX 컨트롤러는 스위치들의 컨트롤을 담당한다. 즉, 스위치 안의 플로우 테이블[9] 정보들은 Secure Channel[10]을 통해 NOX 컨트롤러로 전달되게 되고 스위치의 목록과 링크 정보를 수집한다. 이러한 정보를 수집하기 위해 컨트롤러에서는 적합한 이벤트를 발생 시켜주기에 따른 정보를 수집하게 해준다.

#### 3.1.2. Pronto 3290 Switch

OpenFlow 네트워크에서는 스위치를 PC 기반이나 상용 스위치에서 OpenFlow 프로토콜 제어가 가능하도록 제공할 것이다. 본 테스트베드에서는 상용스위치인 Pronto 3290 스위치를 국내 최초로 도입 하여 OpenFlow Network를 위해 스탠퍼드 대학에서 개발한 펌웨어를 사용하였다. 이는 최신 스펙 코드를 적용할 수 있고 안정적인 OpenFlow 기술을 사용할 수 있다. 이러한 특징점으로 인해 테스트베드의 코어 스위치로 채택하였고 펌웨어를 최신 버전에 맞게 설치하였다. 이로써 스위치의 Secure Channel[]과 OpenFlow 프로토콜을 NOX 컨트롤러와 통신하여 플로우 테이블 안의 12개의 튜플 정보[11]를 특정 액션을 취할 수 있게 만들 수 있는 환경을 제공한다. 또한, 스위치는 이러한 정보들을 수집해야 하는데 이를 위해 NOX 컨트롤러에 적합한 이벤트를 발생시킨다.

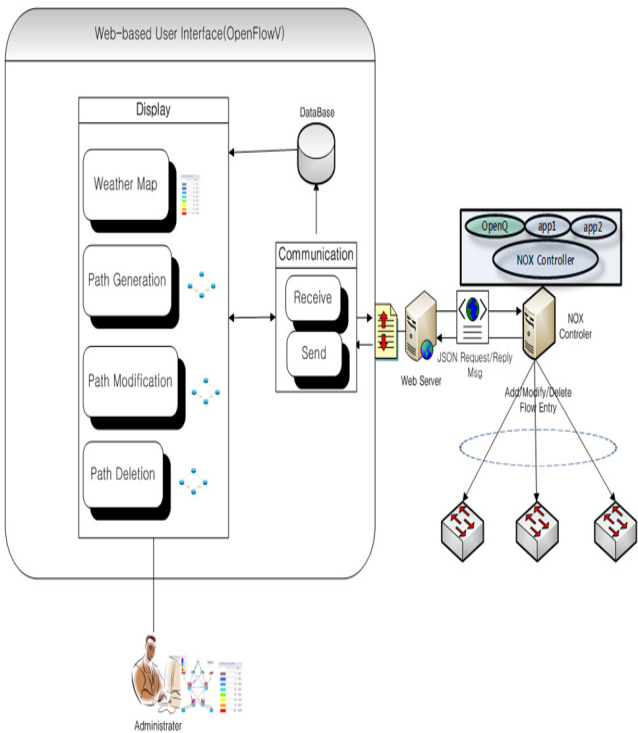
#### 3.1.3. Web-based User Interface

OpenFlow 네트워크의 실시간 트래픽 정보를 웨더맵 형식의 인터페이스를 사용자에게 제공하고 Path에 대한 컨트롤이 가능하도록 하기 위해 웹서버를 두었다. 이를 위해 NOX 컨트롤러에게 주기적으로 트래픽 정보나 Path 컨트롤

률에 대한 리퀘스트나 응답을 주고 받는다. 즉, 웹에 접속하여 테스트베드 토폴로지 정보를 요청하게되면 요청에 대한 정보가 웹 서버를 거쳐 NOX 컨트롤러에게 전송되고 요청을 받은 NOX 컨트롤러는 요청한 컴포넌트를 실행시켜 스위치를 제어하게 된다.

3.2 QoS 메커니즘을 위한 시스템 동작 시나리오

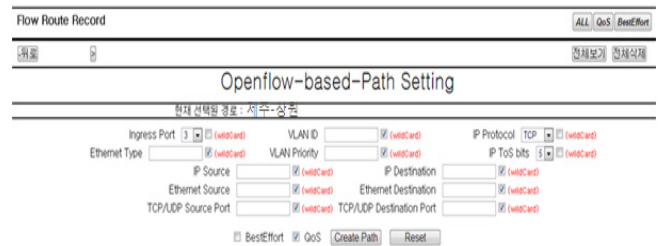
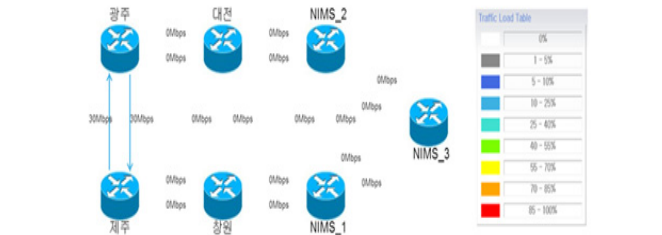
[그림 2]은 OpenFlow를 기반으로 한 QoS 메커니즘을 위한 시스템 동작 시나리오이다. [그림 2]에서 볼 수 있듯이 사용자가 실시간 트래픽을 웨더맵 형식으로 모니터링하거나 Path를 변경, 삭제, 생성을 하기 위해서 웹 인터페이스를 통해 원하는 패킷의 VLAN ID, Ingress, MAC 주소, ToS[12], 프로토콜 타입 등 OpenFlow Spec 1.0에서 지원하는 12가지 튜플 정보들을 입력 하게 되며, 웹 서버는 요구 조건을 NOX 컨트롤러에게 리퀘스트를 보내고 NOX 컨트롤러는 이 해당 요청을 스위치의 플로우 테이블 엔트리에 매핑 시키는 작업을 하고 이에 따른 이벤트를 발생시켜 링크 로드율이나, 각종 정보를 DB에 저장하게 된다. DB에 저장된 정보들은 웹서버가 실시간으로 Reflash 시켜서 사용자는 이러한 정보를 확인할 수 있게 된다. 또한, 패킷 발생 시 ToS 필드에 대한 정보가 없으면 QoS 서비스 요청이 없는 상태로 가정하고 Best Effort[13] 서비스로 가정하여 일반적인 최단 경로 라우팅을 제공한다. 하지만 ToS 필드가 마킹된 패킷을 수신하게 되면 QoS 서비스 요청을 가정하여 Flow Table 엔트리를 생성하여 품질을 보장할 수 있는 경로로 서비스를 제공하는 시나리오를 갖는다.



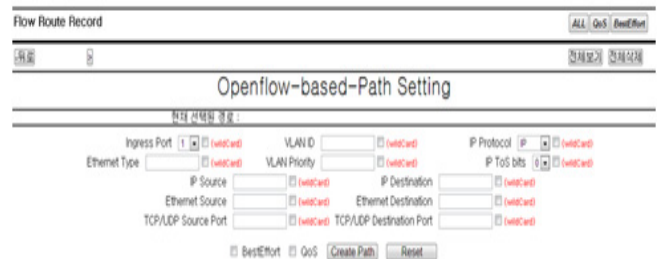
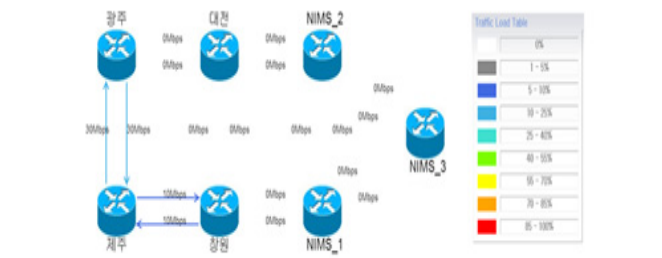
[그림 2] Web Interface의 동작 설계

3.3 웹을 통한 QoS 서비스 요청

[그림 3]는 웹을 통하여 QoS 서비스 요청을 하기 위해 12가지의 튜플의 값을 입력하고 원하는 경로를 클릭한 후 경로를 생성한 그림이다. 생성 후에 [그림 5]과 같이 생성한 플로우 리스트들이 나오고 QoS 서비스를 하는 플로우에 대해서는 Y가 나오고 Best Effort 서비스를 하는 플로우는 N가 나온다. 변경과 삭제를 요청하기 위해선 생성된 플로우 리스트의 체크박스에 체크를 하고 변경, 삭제 혹은 플로우에 대한 자세한 정보를 볼 수 있는 보기 버튼을 클릭하여 사용자가 원하는 데로 경로를 삭제, 변경, 추가, 실시간 트래픽 모니터링이 가능하다.



[그림 3] 웹에서 12가지 튜플 입력하는 모습



[그림 4] 웹에서 Path가 생성된 모습



[http://www.ktword.co.kr/abbr\\_view.php?m\\_temp1=1392](http://www.ktword.co.kr/abbr_view.php?m_temp1=1392)

[14]

<http://harewostelecom.blogspot.com/2009/03/jgn-jgn2-jgn2plus.html>

Flow ID	QoS/Route	생성자	생성 시간	보기	삭제
15	N Route	생성자	2011-09-29 12:20:43	보기	삭제
14	Y Route	생성자	2011-09-29 12:20:25	보기	삭제

[그림 5] Flow 리스트가 생성된 모습

#### 4. 결론

본 논문에서는 상용으로 제공되는 스위치인 Pronto 3290 스위치 기반의 OpenFlow 네트워크에서 QoS 서비스를 제공할 수 있는 NOX에서 돌아가는 컴포넌트와 웹 기반의 인터페이스를 통하여 사용자에게 보다 편한 관리 및 실시간 트래픽 모니터링 가능한 서비스를 구현하였다. 하지만 웹 기반 인터페이스의 디자인적인 측면과 사용자를 위한 회원관리 기능, 실제 플로우의 동적인 가시화를 통해 플로우 정보를 보다 직관적으로 이해 할 수 있는 인터페이스가 요구되고, 여러 가지 아이디어와 실험을 통해 모바일 네트워크 환경이나 네트워크 가상화 환경을 통한 실험 환경과 KOREONET 연구망과 일본의 JGN2Plus[14] 등 국제 연구망 연동을 통해 확장을 해가야 할 것이다.

#### 참고문헌

[1] <http://www.openflow.org/>  
 [2] <http://ko.wikipedia.org/wiki/QoS>  
 [3] <http://www.no-ip.com/>  
 [4] Jeon van der Ham "What is control plane ?", Presentation given at GLIF meeting  
 [5] Cristian Estan "Data Plane Algorithms", University of Wisconsin-Madison  
 [6] <http://www.kreonet.re.kr/>  
 [7] <http://www.terms.co.kr/VLAN.htm>  
 [8] <http://www.network-weathermap.com/gallery>  
 [9] Openflow-Switch 1.0 Specification page 5 ~ 7  
 [10] Openflow-Switch 1.0 Specification page 16 ~ 18  
 [11] Openflow-Switch 1.0 Specification page 8  
 [12] [http://www.ktword.co.kr/abbr\\_view.php?m\\_temp1=2051&mgid=134](http://www.ktword.co.kr/abbr_view.php?m_temp1=2051&mgid=134)  
 [13]