논문 2011-48TC-8-6

미래인터넷 테스트베드 가상화 자원의 QoS를 위한 NetFPGA 기반 스케쥴러 구현 및 성능 평가

(NetFPGA-based Scheduler Implementation and its Performance Evaluation for QoS of Virtualized Network Resources on the Future Internet Testbed)

민 석 홍*, 정 회 진*, 김 병 철**, 이 재 용**

(Seokhong Min, Whoijin Jung, Byungchul Kim, and Jaeyong Lee)

요 약

현재, 미래인터넷에 대한 연구가 해외 및 국내에서 활발하게 진행되고 있으며, 국내에서는 미래인터넷 연구를 위하여 한국전자통신연구원과 국내 4개 대학을 중심으로 'FiRST(Future Internet Research for Sustainable Testbed)'라는 미래인터넷 테스트베드 구축 및 관련 핵심 기술 개발 프로젝트가 진행되고 있다. 'FiRST'프로젝트 중 국내 4개 대학이 공동으로 수행하고 있는 'FiRST@PC'의 경우 프로그래머블 플랫폼을 이용하여 오픈플로우 스위치 기반의 테스트베드를 KOREN과 KREONET에 구축하였다. 현재, 테스트베드 구축에 사용 된 프로그래머블 스위치의 가상화를 통하여 테스트베드를 이용하는 실험자들에게 독립적인 네트워크의 구축이 가능 하도록 하는 연구가 진행 중에 있다. 이때, 테스트베드의 가상화는 테스트베드를 이용하는 실험자들에게 슬라이스라는 단위로 독립적인 네트워크의 구성을 가능하도록 하며, 각 실험자에게 제공 된 슬라이스는 신뢰성 있고 안정적인 네트워크의 자원 사용 기회가 보장되어야 한다. 본 논문에서는 미래인터넷 테스트베드를 이용하는 실험자들에게 가상화된 네트워크인 슬라이스에 QoS를 제공하기 위하여 하드웨어 기반의 패킷 처리를 지원하는 프로그래머블 플랫폼인 NetFPGA 플랫폼을 이용하여 슬라이스의 트래픽을 스케쥴링하기 위한 스케쥴러를 구현하였고, 테스트베드를 구축하여 성능 실험을 하였으며, 실험을 통하여 미래인터넷 테스트베드의 가상화된 네트워크에 신뢰성있고 안정적으로 QoS를 제공할 수 있음을 확인하였다.

Abstract

Recently, research activities on the future internet are being actively performed in foreign and domestic. In domestic, ETRI and 4 universities are focused on implementation of a testbed for research on the future internet named as 'FiRST(Future Internet Research for Sustainable Testbed)'. In the 'FiRST' project, 4 universities are performing a project in collaboration named as 'FiRST@PC' project that is for an implementation of the testbed using the programmable platform-based openflow switches. Currently, the research on the virtualization of the testbed is being performed that has a purpose for supporting an isolated network to individual researcher. In this paper, we implemented a traffic scheduler for providing QoS by using the programmable platform that performs a hardware-based packet processing and we are implemented a testbed using that traffic scheduler. We perform a performance evaluation of the traffic scheduler on the testbed. As a result, we show that the hardware-based NetFPGA scheduler can provide reliable and stable QoS to virtualized networks of the Future Internet Testbed.

Keywords : Future Internet, Network Virtualization, Scheduler, QoS, NetFPGA

Ⅰ. 서 론

오늘날 우리가 사용하고 있는 인터넷은 코어 망에서 고속의 포워딩만 지원한다는 단순화 설계 개념을 기반으로 설계되고 구축되었기 때문에, 이로 인하여 확장성

* 학생회원, ** 평생회원, 충남대학교 정보통신공학과
(Chungnam National University)

※ 본 연구는 방송통신위원회의 미래인터넷 인프라를 위한 가상화 지원 프로그래머블 플랫폼 및 핵심원천 기술개발 사업의 연구결과로 수행되었음
(KCA-2011-09913-05006)
접수일자: 2011년4월5일, 수정완료일: 2011년8월17일

(scalability), 보안(security), 가상화(virtualization) 및 QoS(Quality of Service)에 대한 고려가 충분하지 않았다. 이러한 이유로 인하여 인터넷을 이용하여 새롭고 다양한 서비스를 제공함에 있어서 여러 가지 어려움이 나타나고 있다. 이러한 문제를 해소하기 위하여 현존하는 인터넷에 대한 개선 및 향후 도래할 새롭고 다양한 서비스들에 대한 지원을 위한 방안으로 미래인터넷이 화두로 등장하게 되었으며, 미국, 유럽 및 일본에서 GENI(Global Environment for Network Innovation[1], FP7 FIRE (Future Internet Research and Experimentation)[2] 및 NwGN(New Generation Network)[3] 등의 미래인터넷 연구가 "clean-slate"기법을 적용하여 활발하게 진행 중에 있다. 이러한 미래인터넷 연구들은 미래의 새로운 인터넷 구조와 새로운 프로토콜들을 시험할 수 있는 테스트베드에 초점을 맞추고 있으며, 프로그래머블 플랫폼을 비롯한 네트워크의 가상화, 자원관리 및 통합등과 같은 여러 진보된 개념을 담고 있다[4]. 그림 1은 미국의 미래인터넷 테스트베드인 GENI 테스트베드의 네트워크 구조를 나타낸다.

이러한 해외의 미래인터넷 연구에 대한 움직임에 따라 국내에서도 미래인터넷 연구가 활발하게 진행되고 있으며, 미래인터넷 테스트베드를 구축하고 관련 핵심 기술 개발을 위한 프로젝트가 'FiRST'(Future Internet Research for Sustainable Testbed)[5]라는 이름으로 진행 중에 있다. 현재, 'FiRST' 프로젝트는 한국전자통신연구원과 국내 4개 대학이 참여하여 진행 중에 있으며, 한국전자통신연구원에서 상용장비를 이용한 대용량 플랫폼을 개발하여 테스트베드를 구축하는 'FiRST@ETRI'와 국내 4개 대학이 프로그래머블 플랫폼을 이용하여 공동으로 테스트베드를 구축하고 관련 핵심 기술을 개발하는 'FiRST@PC'로 세분화 되어서 진행 중에 있다. 향후, 'FiRST'프로젝트의 미래인터넷 테스트베드의 가상화 및 테스트베드 통합을 통하여 미래인터넷 테스트베드를 이용하는 여러 연구자들에게 각자의 연구를 독립적으로 수행할 수 있도록 테스트베드 자원의 할당이나 처리가 지원되어 다른 연구자들의 연구에 영향을 받지 않도록 연구 환경이 제공되는 국내의 미래인터넷 테스트베드로 활용함과 동시에 국내의 미래인터넷 테스트베드의 선도적인 역할을 기대하고 있다. 또한, 해외 미래인터넷 테스트베드와의 통합을 추진하여 전 세계적인 테스트베드 구축을 진행할 예정이다.

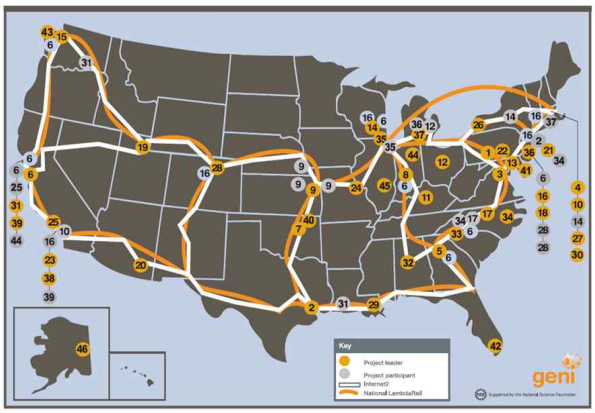

그림 1. 미국의 미래인터넷 테스트베드-GENI[1]
Fig. 1. Future Internet Testbed of U.S.-GENI[1].

본 논문에서는 향후 미래인터넷 테스트베드에 네트워크 가상화 기술을 적용하여 가상화된 미래인터넷 테스트베드 자원을 이용하는 이용자들에게 고속으로 구축된 환경에서 신뢰성 있고 안정적으로 네트워크의 자원 사용 기회에 대한 QoS를 제공할 수 있는 요소기술을 구현하였다. 이를 위하여 하드웨어 기반으로 패킷 처리를 지원하는 프로그래머블 플랫폼인 NetFPGA 플랫폼을 이용하여 스케쥴러를 구현하였으며, 스케쥴러의 경우 DRR(Deficit Round Robin) 및 Priority Queueing 기능을 구현하였다. 또한, 실험을 통하여 향 후 네트워크 가상화 기술이 적용된 미래인터넷 테스트베드를 이용하는 실험자들에게 가상화된 네트워크 자원 사용에 대한 QoS 제공을 할 수 있음을 보였다. 본 논문의 II장에서 국내 4개 대학이 공동으로 수행하고 있는 'FiRST@PC' 프로젝트의 현재 연구 동향 및 향후 연구 방향에 대하여 살펴보고, III장에서는 미래인터넷 테스트베드 구축 및 본 논문의 구현에 이용된 NetFPGA 플랫폼에 대한 간략한 소개와 가상화된 네트워크 자원에 QoS를 제공하기 위하여 구현 된 스케쥴러에 대한 설명을 하였으며, IV장에서 실험을 통한 성능 평가에 대하여 설명하였다. 마지막으로 V장에서는 구현 내용에 대한 결론 및 향후 연구 계획에 대하여 기술함으로 본 논문을 마무리하였다.

II. FiRST@PC 프로젝트

1. FiRST@PC

현재, 해외 여러나라에서 미래인터넷 연구를 위한 활동이 활발하게 이루어지고 있으며, 이러한 분위기에 발맞추어 국내에서도 미래인터넷 연구를 위한 활동이 활

발하게 진행 중이다. 국내의 미래인터넷 연구를 위한 활동 중에서 미래인터넷 구조 와 관련 핵심 기술 개발을 위한 연구의 일환으로 'FiRST'라 불리는 프로젝트가 진행 중에 있으며, 국내 4개 대학(충남대학교, 광주과학기술원, 포스텍, 경희대학교)이 함께하여 'FiRST@ PC'라고 불리는 세부 프로젝트가 진행되고 있다. 'FiRST@PC'는 NetFPGA 플랫폼이라는 하드웨어 기반의 패킷 처리가 지원되는 프로그래머블 플랫폼을 이용하여 오픈플로우 프로토콜을 사용하는 오픈플로우 컨트롤러와 오픈플로우 스위치를 이용하여 미래인터넷 테스트베드를 구축하고, 그 위에 서비스 합성 관련 핵심 기술 개발이 진행 중인 프로젝트이다. 향후, 구축된 미래인터넷 테스트베드의 네트워크의 가상화를 통하여 다수의 실험자에게 독립된 실험 공간을 제공할 수 있는 미래인터넷 테스트베드의 구축에 대한 연구가 진행 중에 있으며, 'FiRST@ ETRI'와의 테스트베드 통합을 통하여 추후 미래인터넷의 연구에 활용할 수 있는 선도적인 역할이 될 수 있도록 함을 목적으로 두고 있다. 'FiRST@PC'의 미래인터넷 테스트베드는 안정적인 결과물을 상시 운용하여 미래인터넷 테스트베드를 이용하는 실험자에게 실험환경을 제공하는 'NetOpen Production'과 'NetOpen Production' 적용 이전에 혁신적인 새로운 기능의 적용 여부를 시험하는 'NetOpen Alpha'로 구분하고 이원화하여 운용할 계획을 갖고 있다. 다음 절에서 'FiRST@PC'의 미래인터넷 테스트베드에 대하여 설명한다.

2. 미래인터넷 테스트베드

'FiRST@PC' 프로젝트에서 구축된 미래인터넷 테스트베드는 향후 미래인터넷 연구의 선도적인 역할뿐만 아니라 연구개발자들에게 여러 가지 새로운 프로토콜 및 어플리케이션을 실험할 수 있는 안정적인 환경을 제공하는 것을 목적으로 구축되었으며, 그림 2와 같이 NetFPGA 플랫폼 및 오픈플로우 기반의 프로그래머블 플랫폼을 이용하였다. 구축된 미래인터넷 테스트베드는 KOREN/KREONET에 오픈플로우 스위치들을 제어하는 오픈플로우 컨트롤러와 캡슐레이터(capsulator)가 추가된 NetFPGA 플랫폼 기반의 오픈플로우 스위치들을 이용하여 지역적으로 떨어져 있는 각 학교의 로컬 테스트베드를 연동하였다. 이때, 테스트베드의 네트워크 구성이 육안으로 확인이 가능하고 임의로 네트워크를 구성할 수 있도록 ENVI[6]를 확장한 GUI가 같이 연동된다. 구축된 미래인터넷 테스트베드에서 사용자가 네트워크

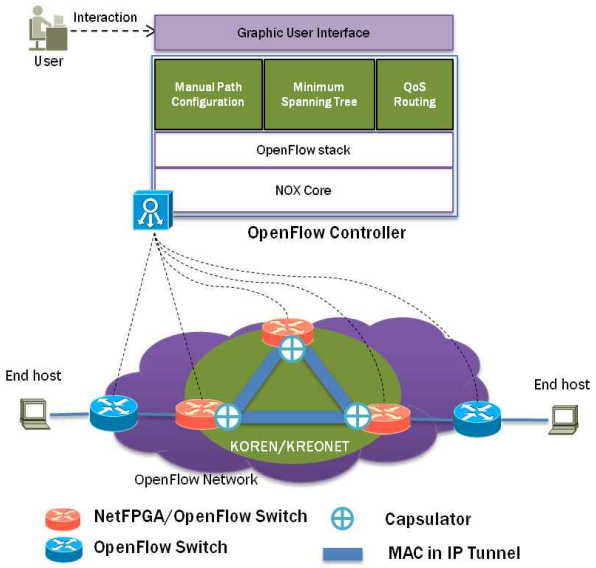

그림 2. 'FiRST@PC' 테스트베드 구조[7]
Fig. 2. 'FiRST@PC' Testbed Architecture[7].

사용을 위한 트래픽을 발생시키면, 발생된 요구에 대하여 오픈플로우 스위치에서 오픈플로우 컨트롤러로 네트워크 사용을 위한 요구가 전달되며, 오픈플로우 컨트롤러는 사용자의 요구 사항을 만족시키기 위하여 미리 설정한 제어 기능을 이용하여 프로그래머블한 네트워크를 형성한다. 이때, 동적으로 사용자 트래픽의 경로를 형성함으로써 프로그래머블한 네트워크가 형성되게 된다. 예를들어, 미래인터넷 테스트베드에서 패킷들은 특정 서비스 또는 우선 순위에 따라 지정되는 경로를 따라 목적지에 전달될 수 있는데, 모든 오픈플로우 스위치 간의 최소거리의 경로, QoS 메트릭을 만족하는 경로를 만든다거나 또는 테스트베드 유저에 의하여 경로가 생성되도록 만드는 방법으로 프로그래머블한 네트워크를 제공할 수 있다[7].

3. 미래인터넷 테스트베드의 네트워크 가상화

'FiRST@PC' 프로젝트의 미래인터넷 테스트베드 구축과 관련된 연구 중 하나인 네트워크 가상화는 하나의 물리적인 미래인터넷 테스트베드를 네트워크 가상화를 통하여 여러 개의 논리적인 네트워크로 나누어 미래인터넷 테스트베드를 이용하는 실험자에게 독립적인 실험 공간을 제공하는 것을 목적으로 하고 있으며, 슬라이스 단위로 가상화된 미래인터넷 테스트베드를 구축할 계획이다. 미래인터넷 테스트베드의 네트워크 가상화를 통하여 각 슬라이스 별로 네트워크의 대역폭을 할당하고, 독립된 네트워크 토폴로지를 갖도록 한다. 또한, 다른

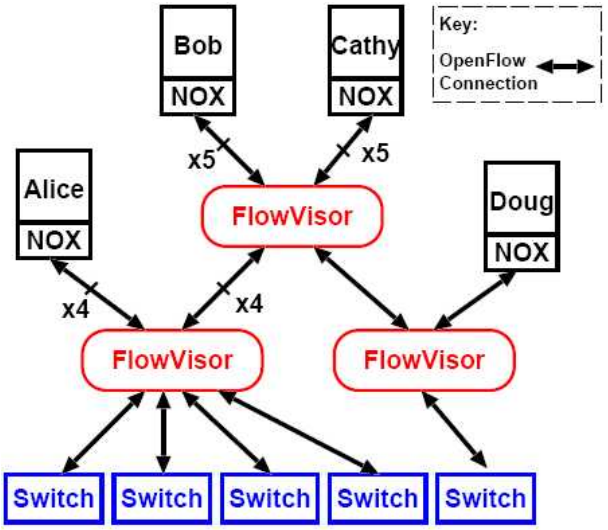

그림 3. FlowVisor를 이용한 네트워크 가상화[8]
Fig. 3. Network Virtualization with FlowVisor[8].

슬라이스의 트래픽과의 분리를 통하여 다른 슬라이스의 트래픽에 대한 영향을 받지 않으며, 오픈플로우 스위치의 CPU 자원과 플로우 테이블을 분리하여 다른 슬라이스에게 영향을 주지 않도록 하는 네트워크의 가상화에 대한 연구를 진행하고 있다. 또한, 가상화된 미래인터넷 테스트베드를 이용하는 실험자에게 네트워크 자원 사용에 대한 보장을 위한 방안으로 슬라이스에 QoS를 제공하는 연구도 진행 중이다. 미래인터넷 테스트베드의 네트워크 가상화에 대한 연구 방향은 다음과 같다.

일반적으로 네트워크 가상화는 전형적으로 구현의 자유도가 높은 소프트웨어의 이점을 이용하거나 또는 하드웨어 컴포넌트의 중복 사용을 통하여 이루어졌다. 하지만, 미래인터넷 테스트베드에서의 네트워크 가상화는 오픈플로우 스위치에서 포트 별로 슬라이스를 나누고 VLAN 태그에 VLAN ID를 마킹하는 방법으로 VLAN을 이용하는 방법과 그림 3과 같이 오픈플로우 컨트롤러와 오픈플로우 스위치 사이에 FlowVisor라는 네트워크 가상화를 위한 특정 어플리케이션을 위치시키는 방법을 통해 이루어진다[8].

먼저 VLAN 태그를 통하여 플로우 별 트래픽 분리를 할 수 있게 되어 플로우 별로 가상화된 자원의 할당이나 처리를 가능하게 한다. FlowVisor의 경우 트래픽 처리를 위한 가상화된 자원의 개별 컨트롤러의 운영을 독립적으로 수행이 가능하도록 하는데 그림 3과 같이 여러 컨트롤러들에 필요한 기능을 구현하고 스위치에 트래픽이 들어올 경우 원하는 컨트롤러와의 제어 연동이 이루어지게 한다. 이 방법의 경우 오픈플로우 컨트롤러와 오픈플로우 스위치는 FlowVisor의 존재를 감지하지 못하며, FlowVisor는 오픈플로우 컨트롤러와 오픈플로우 스위치 간의 오픈플로우 프로토콜을 이용하여 주고받는 메시지를 중간에서 가로채서 수정하는 방법을 이용하여 네트워크를 가상화하게 된다.

Ⅲ. NetFPGA 플랫폼 기반 스케줄러 구현

1. NetFPGA 플랫폼

NetFPGA 플랫폼[9]은 네트워크 관련 학생 및 연구자들에게 필요한 교육 및 연구 개발의 환경을 제공하기 위하여 미국의 스탠포드 대학에서 개발되어 네트워크와 관련된 여러 분야에서 세계적으로 이용률이 증가하고 있는 오픈 프로그래머블 네트워크 개발 플랫폼으로 그림 4와 같이 일반적으로 사용하는 PC에 NetFPGA 하드웨어가 추가된 형상을 하고 있으며, 그림 5와 같은 구조를 갖는다.

현재, 전 세계적으로 여러 기관에서 1G 이더넷 인터페이스를 지원하는 NetFPGA 플랫폼을 연구개발에 이용하고 있으며, 10G 이더넷 인터페이스를 지원하는 NetFPGA 플랫폼이 최근 출시되었다. NetFPGA 플랫폼은 패킷 처리를 전담하는 전용의 하드웨어와 기타 필요한 어플리케이션을 구현할 수 있는 소프트웨어 패키지로 구성되어 있으며, 일반 PC를 이용하여 어렵지 않게 개발환경을 구축하여 사용할 수 있도록 되어 있다. NetFPGA 플랫폼의 가장 큰 매력은 전용의 하드웨어를 이용하여 CPU의 사용률 점유없이 line-rate로 고속의 패킷 처리가 가능하다는 점에 있으며, 패킷 처리를 전담하는 NetFPGA 하드웨어는 4개의 기가 비트 이더넷 인터페이스와 PCI 인터페이스가 제공되며, 제공되는

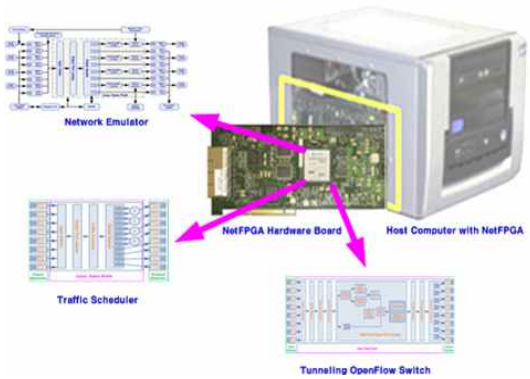

그림 4. NetFPGA 플랫폼
Fig. 4. NetFPGA platform.

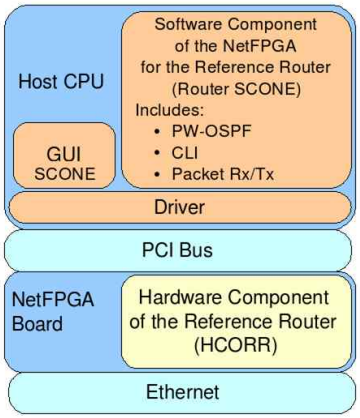

그림 5. NetFPGA 플랫폼의 구조[9]
Fig. 5. Overall Architecture of NetFPGA platform[9].

PCI 인터페이스를 통하여 PC에서 NetFPGA 하드웨어를 제어할 수 있도록 설계되어 있다. NetFPGA 플랫폼을 이용하는 개발자는 PC의 어플리케이션과 NetFPGA 하드웨어의 FPGA 로직 구현에만 전념하면 되도록 설계되어 있어 NetFPGA 플랫폼을 이용한 고속의 패킷 처리 시스템 구축에 용이하다는 점 또한 매력이 아닐 수 없다. NetFPGA 플랫폼을 이용하는 개발자는 NetFPGA 플랫폼 하드웨어의 FPGA(Field Programmable Gate Array)에 HDL(Hardware Description Language)이라는 디지털 하드웨어의 논리 로직을 기술하는 언어를 이용하여 고속의 패킷 처리 로직을 구현하고, C, C++, Java, Perl 또는 Python 과 같은 프로그래밍 언어를 이용하여 필요한 어플리케이션을 구현하여 필요한 시스템을 개발하여 이용할 수 있다. 또한, NetFPGA 플랫폼을 이용하여 구현된 오픈 코드들이 점점 증가 하고 있기 때문에 향후 이를 이용한 교육 및 연구개발에 NetFPGA 플랫폼의 활용도가 높아질 것으로 기대된다.

2. 네트워크 가상화 자원의 QoS를 위한 스케쥴러

미래인터넷 테스트베드에서 가상화된 네트워크 자원에 QoS를 제공하기 위하여 미래인터넷 테스트베드의 슬라이스에 우선 순위를 부여하여 스케쥴링을 할 수 있도록 PQ(Priority Queueing)와 슬라이스에 할당되어 양자로 사용되는 퀀텀(Quantum)을 이용하여 스케쥴링을 할 수 있는 DRR(Deficit Round-Robin)[10] 스케쥴러를 구현하였다.

가. PQ(Priority Queue) 구현

미래인터넷 테스트베드의 각 슬라이스에 우선 순위를 부여하고 우선 순위에 따른 슬라이스 트래픽 처리를 위한 큐(queue)를 구분한다. 각 슬라이스의 트래픽은 오픈플로우 스위치 내부의 우선 순위에 해당하는 큐에 임시 저장되었다가 처리되는데, 우선 순위가 높은 큐의 트래픽이 항상 우선 처리된다. 따라서, 우선 순위가 낮은 슬라이스의 트래픽을 처리하는 큐의 트래픽이 처리되지 않고 계속 쌓이게 되어 큐의 오버플로우(overflow)가 발생되면, 트래픽의 손실이 발생하게 된다. 즉, 우선 순위가 높은 슬라이스의 트래픽이 계속 발생되게 되면 우선 순위가 낮은 슬라이스의 큐에 트래픽이 남아있는 트래픽은 전송이 되지 않을 수 있으며, 그림 6은 PQ의 트래픽 처리 구조를 나타낸다.

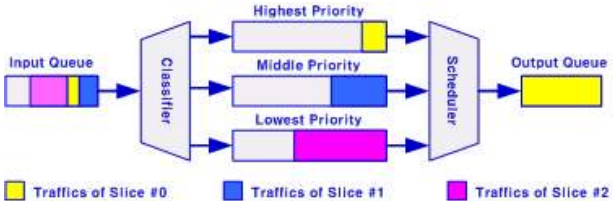

그림 6. PQ의 트래픽 처리 구조
Fig. 6. Traffic Processing Architecture of the PQ.

나. DRR(Deficit Round-Robin)[10] 구현

DRR은 공정하게 다양한 크기의 패킷을 처리하기 위하여 WRR(Weighted Round-Robin)[11] 방식을 수정한 방식으로 패킷의 크기를 이용하여 각각의 슬라이스별로 균일한 트래픽 전송을 제공하게 되는데, 이는 임의로 설정되는 퀀텀(quantum)에 의하여 트래픽이 조절된다. 또한, 각 슬라이스 별로 큐가 할당되며 큐에 있는 패킷의 크기를 DC (Deficit Counter, *bytes*) 와 비교하여 전송 여부를 결정하게 되는데, 이때 DC 는 수식(1) 에 의하여 정해진다[10].

$$DC_{i+1} = DC_i + Quantum - Transmitted\,Packets \quad (1)$$

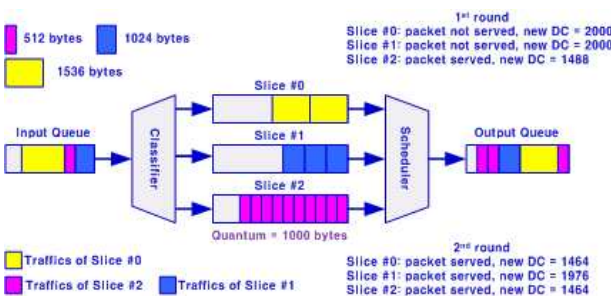

그림 7. DRR의 트래픽 처리 구조
Fig. 7. Traffic Processing Architecture of the DRR.

패킷의 크기가 DC 값 보다 크면 전송은 잠시 보류되고 다음 전송 기회에서 퀀텀 만큼 값이 더해진 DC 값과 큐에서 전송 대기 중인 패킷의 크기와 비교를 하여 패킷의 크기가 DC 값 보다 작으면 전송이 이루어지게 된다. 이때, 적어도 한 개의 패킷을 버퍼링 중인 큐의 정보를 담고 있는 ActiveList를 이용하여 서비스가 필요없는 슬라이스는 서비스를 하지 않는다. 이는 서비스가 필요없는 큐에 대한 불필요한 서비스 시간을 줄임으로써, 스케쥴링 성능의 향상을 제공하며, 그림 7은 퀀텀의 크기가 1000 바이트인 경우의 예를 들은 DRR 모듈의 처리 구조를 나타낸다.

3. NetFPGA 플랫폼을 이용한 스케쥴러 구현

향후 NetFPGA 플랫폼으로 구축된 미래인터넷 테스트베드에 적용하여 VLAN을 통한 가상화된 각 슬라이스에 QoS를 제공하기 위하여 NetFPGA 플랫폼을 이용하여 슬라이스 분류를 위한 트래픽 분류(Traffic Classifier) 모듈과 슬라이스의 트래픽을 처리하는 트래픽 스케쥴러(Traffic Scheduler) 모듈을 구현하였다. 트래픽 분류 모듈은 슬라이스의 VLAN ID를 구분하여 해당 큐에 큐잉하고 트래픽 스케쥴러 모듈에서 트래픽 엔지니어링(Traffic Engineering) 정책(policy)에 따라 스케쥴링되는 구조로 설계 하였으며, NetFPGA 플랫폼의 하드웨어 제어는 PC에서 NetFPGA 플랫폼 하드웨어의 레지스터 액세스를 통하여 이루어진다.

가. Traffic Classifier 모듈

슬라이스를 구분하기 위한 모듈인 트래픽 분류 모듈은 그림 8과 같은 구조를 갖는다. 트래픽 스케쥴러 모듈에서 트래픽을 처리를 하기 위하여, 트래픽에 대한 분류 작업이 선행 되어야 한다. VLAN을 이용한 네트워크 가상화 방법은 VLAN ID로 슬라이스를 구분하기 때문에 슬라이스 분류 모듈에서는 VLAN의 우선 순위 및 VLAN ID를 파악하고, 트래픽의 두 가지 정보를 이용하여 NetFPGA 플랫폼 하드웨어의 FPGA 내부에서 모듈 간 트래픽 전송을 위한 모듈 헤더의 목적지 포트를 수정하여 트래픽 스케쥴러의 해당 큐로 트래픽이 전송되도록 트래픽의 전송경로를 설정한다.

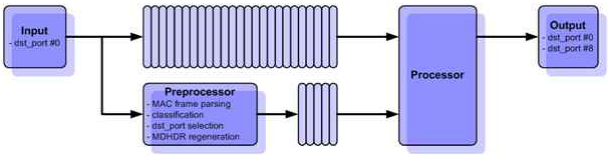

그림 8. Traffic Classifier 모듈의 구조
Fig. 8. Architecture of the Traffic Classifier Module.

나. Traffic Scheduler 모듈

트래픽 분류 모듈을 통하여 슬라이스 별로 분류된 트래픽이 슬라이스 별로 할당된 큐에 입력이 되면, 테스트베드 관리자에 의하여 설정된 트래픽 엔지니어링 정책(우선순위에 따른 스케쥴링을 하여 우선 순위가 높은 슬라이스의 트래픽을 먼저 처리할 것인지 아니면 DRR 스케쥴링을 하여 임의의 정해지는 퀀텀의 비율에 따라 트래픽을 처리할 것인지)에 따라 슬라이스 트래픽의 QoS(Quality of Service)를 위하여 그림 9와 같은 구조를 갖는 트래픽 스케쥴러에서 슬라이스 별 트래픽 제어를 수행하게 되고, 이후 최종 출력 포트로 트래픽을 전송하게 된다.

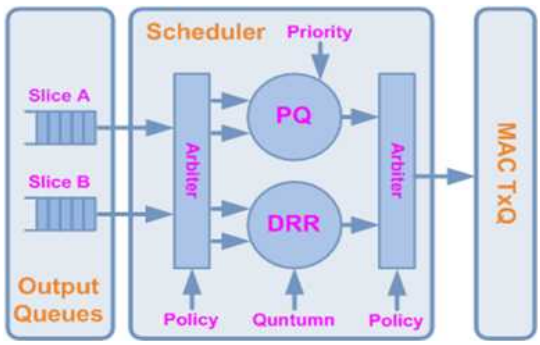

그림 9. Traffic Scheduler 모듈의 구조
Fig. 9. Architecture of the Traffic Scheduler Module.

IV. NetFPGA 스케쥴러 성능 평가

미래인터넷 테스트베드의 네트워크 가상화에 적용하

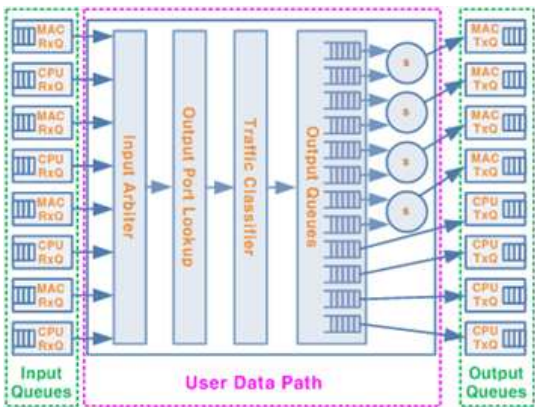

그림 10. 스케쥴러의 NetFPGA 파이프라인 구조
Fig. 10. NetFPGA Pipeline Architecture of the Scheduler.

기 이전에 구현된 모듈의 성능을 확인하기 위하여 슬라이스의 VLAN ID 대신 발생된 트래픽의 전송 목적지의 주소를 이용하여 플로우에 따른 스케쥴링을 하도록 조정된 트래픽 분류 모듈과 트래픽 스케쥴러 모듈을 이용하여 그림 10과 같은 구조를 갖는 스케쥴러를 NetFPGA 플랫폼을 이용하여 구현하였다.

1. Priority Queueing 성능 평가

NetFPGA 플랫폼을 이용하여 구현된 PQ에 대한 성능을 실험하기 위하여 그림 11과 같이 로컬 테스트베드를 구축하였고, iperf[12]를 이용하여 서로 다른 목적지로 전송되는 두 개의 UDP 플로우를 발생시키고 발생된 각 플로우의 우선순위를 다르게 설정하였다. 또한, 두 플로우가 공존하는 전송경로 중간에 NetFPGA 플랫폼을 이용하여 기 구현된 Network Emulator[13]를 이용하여 두 플로우의 가용대역폭을 1Mbps로 제한하여 병목현상(bottleneck)을 유발시키고, 두 가지 경우에 대한 실험을 수행하였다.

첫 번째 실험은 각각의 플로우에 동일하게 600kbps의 트래픽을 발생시켜 목적지로 전송하도록 하였다. 이 경우 우선 순위가 높은 플로우의 트래픽은 목적지까지 생성된 트래픽 모두 전송 되었으나, 우선 순위가 낮은 트래픽은 400kbps 만이 전송되고 200kbps는 목적지 까지 전송되지 못하였으며, 스케쥴러는 안정적으로 스케쥴링 하고 있음을 목적지의 트래픽 전송량을 보고 확인할 수 있었다. 그림 12는 우선 순위가 낮은 플로우의 트래픽을 목적지에서 관찰한 것을 나타내는데 전체 가용대역폭인 1Mbps에서 우선 순위가 높은 플로우가 사용

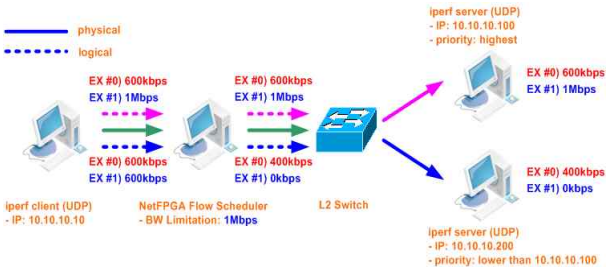

그림 11. Priority Queueing 실험
Fig. 11. Experiment for Priority Queueing.

```
162046.0-162047.0 sec  48.8 KBytes    400 Kbits/sec  4.855 ms   16/   50 (32%)
162047.0-162048.0 sec  48.8 KBytes    400 Kbits/sec  4.021 ms   17/   51 (33%)
162048.0-162049.0 sec  48.8 KBytes    400 Kbits/sec  3.940 ms   17/   51 (33%)
162049.0-162050.0 sec  48.8 KBytes    400 Kbits/sec  3.921 ms   17/   51 (33%)
162050.0-162051.0 sec  48.8 KBytes    400 Kbits/sec  3.930 ms   17/   51 (33%)
162051.0-162052.0 sec  48.8 KBytes    400 Kbits/sec  3.920 ms   17/   51 (33%)
```

그림 12. Priority Queueing 실험 결과
Fig. 12. Experimental Results for Priority Queueing.

하고 남은 가용대역폭인 400kbps만이 전송됨을 확인할 수 있다.

두 번째 실험에서는 우선 순위가 높은 플로우의 트래픽을 링크의 가용대역폭인 1Mbps까지 발생시켜 보았다. 이 경우에는 우선 순위가 높은 트래픽이 전체 가용대역폭인 1Mbps를 전부 사용하여 우선 순위가 낮은 플로우의 트래픽은 목적지까지 전혀 전송되지 못하는 기아(starvation) 현상을 볼수 있었다.

2. DRR 스케쥴러 성능 평가

NetFPGA 플랫폼을 이용하여 구현된 DRR 스케쥴러에 대한 성능을 실험하기 위하여 그림 13과 같이 로컬 실험망을 구축하였다. 이번 실험은 DRR 스케쥴러의 성능뿐만 아니라 고속의 데이터 링크에서 하드웨어의 이점을 활용한 스케쥴링 성능을 확인하기 위하여 iperf[12]를 이용하여 각각 450Mbps의 전송속도를 갖는 2개의 UDP 플로우를 발생시켰다. 또한, 두 플로우가 공존하는 전송경로 중간에 NetFPGA 플랫폼을 이용하여 기 구현된 Network Emulator[13]를 이용하여 400Mbps로 가용대역폭을 제한하여 병목현상(bottleneck)을 유발시키고, 각 플로우의 퀀텀 크기를 변경하며 실험 하였다.

첫 번째 실험은 각 플로우의 퀀텀을 동일하게 설정하였다. 이 경우, 각 플로우는 안정적으로 200Mbps씩 공정하게 가용대역폭을 사용하고 있음을 확인할 수 있었으며, 그 후 각 플로우의 퀀텀 비율을 변경해가며 실험

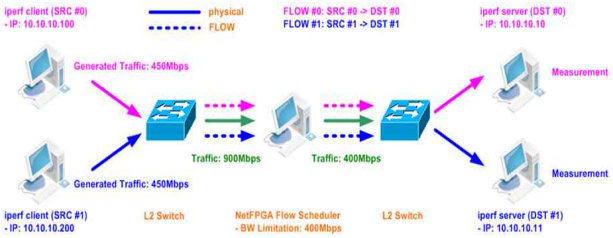

그림 13. DRR 스케쥴러 실험
Fig. 13. Experiment for the DRR Scheduler.

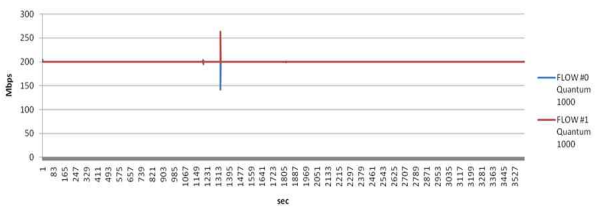

그림 14. DRR 스케쥴러 실험 결과 (동일 퀀텀)
Fig. 14. Experimental Results for DRR Scheduler. (equal Quantum size)

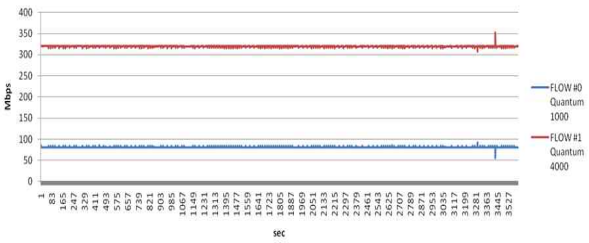

그림 15. DRR 스케쥴러 실험 결과 (4:1 퀀텀)
Fig. 15. Experimental Results for the DRR Scheduler. (4:1 Quantum Size).

표 1. Quantum에 따른 throughput의 변화
Table 1. Throughput for varying Quantum sizes.

	Flow #0	Flow #1	Flow #0	Flow #1
Quantum (bytes)	1000	1000	1000	2000
Throughput (Mbps)	200	200	134	267
Quantum (bytes)	1000	4000	1000	9000
Throughput (Mbps)	80.1	320	40	360

하였고, 1시간 동안 각 플로우를 모니터링하여 실험 결과를 얻어내었다. 각 플로우의 퀀텀이 크기가 1000으로 동일한 비율로 퀀텀이 설정된 경우의 모니터링 결과를 그림 14에 나타내었고, 그림 15에 퀀텀의 크기가 1000과 4000인 경우의 모니터링 결과를 나타내었다.

각 실험의 경우 NetFPGA 플랫폼의 하드웨어에 의한 패킷 처리로 인하여 고속의 환경에서도 안정적으로 동작하고 있음을 보이고 있어 NetFPGA 플랫폼을 이용한 이점이 잘 드러나고 있음을 확인할 수 있었으며, 표 1에 퀀텀의 크기에 따른 모니터링 결과를 1시간 동안의 평균 전송률로 나타내었다.

V. 결 론

본 논문에서는 현재 진행되고 있는 미래인터넷에 대한 국내 연구 및 해외 연구 동향에 대하여 간략하게 살펴보았으며, 국내의 미래인터넷에 대한 연구 중 하나인 'FiRST' 프로젝트에 대하여 살펴보았다. 또한, 'FiRST' 프로젝트 중 국내의 4개 대학이 공동으로 수행하고 있는 연구인 'FiRST@PC'에 대하여 좀 더 살펴봄으로써, 현재 미래인터넷 연구의 동향과 향후 연구방향의 일부로써 프로그래머블 플랫폼인 NetFPGA 플랫폼을 이용하여 구축한 미래인터넷 테스트베드의 네트워크 가상화 방안과 가상화된 네트워크 자원인 슬라이스에 QoS를 제공하기 위한 방안으로 스케쥴러를 구현하여 이용하는 방안을 살펴보았다. 또한, 실험을 통하여 구현된 스케쥴러가 NetFPGA 플랫폼의 가장 큰 장점인 전용의 하드웨어를 이용한 패킷 처리에 의하여 고속의 환경에서도 구현 알고리즘대로 안정적으로 동작하고 있음을 확인하였다. 이를 통하여, 'FiRST@PC' 프로젝트의 미래인터넷 테스트베드에서 네트워크 가상화에 대한 연구 진행 시 구현된 스케쥴러를 이용하여 미래인터넷 테스트베드를 이용하는 실험자들에게 최소한의 네트워크 사용기회 보장이 가능함을 확인하였다.

향후, 미래인터넷 테스트베드의 네트워크 가상화에 대한 연구를 진행 할 것이며, 구현된 스케쥴러를 미래인터넷 테스트베드의 NetFPGA 플랫폼을 이용하는 오픈플로우 스위치에 적용하여, 네트워크 가상화 자원에 QoS를 제공하는 연구를 진행할 것이다.

참 고 문 헌

[1] GENI: Global Environment for Network Innovatio ns, http://www.geni.net
[2] FIRE: Future Internet Research and Experimentation, http://cordis.europa.eu/fp7/ict/fire/
[3] Shuji Esaki, Akira Kurokawa, and Kimihide Matsumoto, "Overview of the Next Generation Network", NTT Technical Review, Vol. 5, No.6, June 2007.
[4] Man Kyu Park, Whoi Jin Jung, Jae Yong Lee, Byung Chul Kim, "A Study of Future Internet Testbed Construction using NetFPGA/OpenFlow Switch on KOREN/KREONET", Journal of the IEEK, Vol. 47-TC, No. 7, July 2010.
[5] Jinho Haham, Bongtae Kim, and Kyungpyo Jeon, "The study of Future Internet platform in ETRI", The Magazine of the IEEK, Vol. 36, No. 3, March 2009.
[6] OpenFlow forum, http://www.openflowswitch.org/ wp/gui/
[7] Seok Hong Min, Yoon Cheol Choi, Namgon Kim, Wan Kim, Oh Chan Kwon, Byung Chul Kim, Jae Yong Lee, Dae Young Kim, Jongwon Kim, Hwhangjun Song, "Implementation of a

Programmable Service Composition Network using NetFPGA-based OpenFlow Switches", 1st ASIA NetFPGA Developer's workshop, Korea, June 2010.
[8] Rob Sherwood, Glen Gibby, Kok-Kiong Yapy, Gu ido Appenzellery, Martin Casado, Nick McKeown y, Guru Parulkary, "Flow Visor: A Network Vir tualization Layer", http://www.openflow.org
[9] NetFPGA forum, http://www.netfpga.org
[10] M. Shreedhar, George Varghese, "Efficient Fair Queuing Using Deficit Round-Robin", IEEE/ ACM transactions on networking, Vol. 4, No. 3, June 1996.
[11] M. Katevenis, S. Sidiropoulos, and C. Courcoubetis, "Weighted round-robin cell multiplexing in a general-purpose ATM switch chip," IEEE J. Select. Areas Commun., vol. 9, pp. 1265 - 1279, Oct. 1991.
[12] iperf, http://sourcerforge.net/projects/iperf/
[13] Seok Hong Min, Jae Yong Lee, Byung Chul Kim, "A Network Emulator on the NetFPGA Platform", 1st ASIA NetFPGA Developer's workshop, Korea, June 2010.

저 자 소 개

민 석 홍(학생회원)
2005년 공주대학교 전기전자정보공학과 석사
2010년~현재 충남대학교 전자전파정보통신공학과 박사과정
2004년 한국전자통신연구원 BcN 시험기술팀 위촉연구원
2005년 디지피아㈜ 방송장비팀 연구원
2006년~2009년 ㈜엠티아이 연구2실 전임연구원
<주관심분야 : 무선 메쉬 네트워크, 데이터 통신, 미래인터넷>

정 회 진(학생회원)
2005년 충남대학교 전기정보통신공학부 학사
2007년 충남대학교 정보통신공학과 석사
2010년~현재 충남대학교 전자전파정보통신공학과 박사과정
<주관심분야 : 인터넷, 미래인터넷, 이동인터넷, 이동통신>

이 재 용(평생회원)
1988년 서울대학교 전자공학과 학사
1990년 한국과학기술원 전기 및 전자공학과 석사
1995년 한국과학기술원 전기 및 전자공학과 박사
1990년~1995년 디지콤 정보통신연구소 선임연구원
1999년~현재 충남대학교 정보통신공학부 교수
<주관심분야 : 초고속통신, 인터넷, 네트워크 성능분석>

김 병 철(평생회원)-교신저자
1988년 서울대학교 전자공학과 학사
1990년 한국과학기술원 전기 및 전자공학과 석사
1996년 한국과학기술원 전기 및 전자공학과 박사
1993년~1999년 삼성전자 CDMA 개발팀
1999년~현재 충남대학교 정보통신공학부 교수
<주관심분야 : 이동인터넷, 이동통신 네트워크, 데이터 통신>