

논문 2010-47TC-7-16

KOREN/KREONET 기반 NetFPGA/OpenFlow 스위치를 이용한 미래인터넷 테스트 베드 구축 방안 연구

(A Study of Future Internet Testbed Construction using
NetFPGA/OpenFlow Switch on KOREN/KREONET)

박 만 규, 정 회 진*, 이 재 용**, 김 병 철**

(Man Kyu Park, Whoi Jin Jung, Jae Yong Lee, and Byung Chul Kim)

요 약

미래인터넷 연구를 위해 대규모의 테스트 베드를 구축하는 일은 새로운 프로토콜을 테스트하고 clean-slate 기법을 적용한 새로운 네트워크 아키텍처를 테스트하기 위해 꼭 필요하다. 우리나라의 경우 현재 'FIRST'(Future Internet Research for Sustainable Testbed)라는 프로젝트 이름으로 미래인터넷 테스트 베드를 구축하는 연구가 금년 3월부터 추진되어 ETRI와 5개의 대학들이 공동으로 연구를 진행하고 있다. 이 중 5개 대학교를 중심으로 진행되는 FIRST@PC의 경우 NetFPGA의 하드웨어 가속 기능을 이용하여 오픈플로우 스위치기반의 미래인터넷 테스트베드를 KOREN과 KREONET에 구현하는 것을 목표로 연구가 진행 중이다. 본 논문에서는 FIRST@PC 프로젝트에 대한 간략한 소개와 지역별로 구축된 오픈플로우 스위치 테스트 베드 간 연동을 위해 구현된 소프트웨어 기반의 MAC in IP 캡슐레이터를 설명하고 이를 이용한 연구실 내의 로컬 테스트와 충남대와 광주과학기술원 간 연동 테스트한 결과를 제시한다. 실험 결과 소프트웨어 기반의 캡슐레이터를 사용하면 현재 많은 어플리케이션이 요구하는 대역폭에 대해서는 지원 가능할 것으로 예상된다.

Abstract

Building a large-scale testbed for Future Internet is very important to evaluate a new protocol and new network architecture designed by clean-slate approach. In Korea, new Future Internet testbed project, called FIRST (Future Internet Research for Sustainable Testbed), has been started since Mar. 2009 to design and test new protocols. This project is working together with ETRI and 5 universities. The FIRST@PC is to implement a virtualized hardware-accelerated PC-node by extending the functions of NetFPGA card and build a Future Internet testbed on the KOREN and KREONET for evaluating newly designed protocols and interesting applications. In this paper, we first briefly introduce FIRST@PC project and explain a 'MAC in IP Capsulator' user-space program using raw-socket in Linux to interconnect OpenFlow enabled switch sites on the KOREN and KREONET. After that, we address test results for TCP throughput performance for varying packet size. The test results show that the software based capsulator can support a reasonable bandwidth performance for most of applications.

Keywords: Future Internet, testbed, NetFPGA, OpenFlow, Capsulator, KOREN/KREONET

I. 서 론

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

※ 본 논문은 “미래인터넷 인프라를 위한 가상화 지원 프로그래머블 플랫폼 및 핵심원천 기술개발” 과제 (2009-F-050 -01)에 대한 결과물 중 일부분으로 연구개발 업무에 도움을 주신 분께 감사드립니다.

접수일자: 2009년11월30일, 수정완료일: 2010년7월14일

현재 사용하는 인터넷은 처음 설계 시 코어 망에서는 고속 포워딩만 지원하겠다는 식의 단순화 설계 개념을 기반으로 하였기에 확장성, 보안, QoS (Quality of Service), 가상화(Virtualization) 기능과 같은 고급 서비스를 지원하는데 한계에 직면하였으며, 새로운 서비스

의 출현은 담보 상태를 보이고 있다. 이에 최근 세계적으로 “clean-slate” 기법을 적용한 새로운 인터넷 구조를 설계하자는 연구와 함께 새로운 프로토콜의 테스트 및 성능 평가 등을 위해 대규모의 실제 테스트베드를 구축하는 대형 프로젝트가 활발히 진행되고 있다.

잘 알려져 있듯이 미국의 경우는 GENI(Global Environment for Network Innovation)^[1], 유럽 연합을 중심으로 한 FIRE (Future Internet Research and Experimentation)^[2], 그리고 일본의 경우 NwGN (New Generation Network)^[3]라는 프로젝트로 각각 미래 인터넷에 대한 연구가 선진국들을 중심으로 활발히 진행되고 있으며, 우리나라에서도 역시 새로운 미래 인터넷 테스트베드를 설계하고 구축하기 위한 새로운 프로젝트가 ‘FIRST’ (Future Internet Research for Sustainable Testbed)^[4]라는 이름으로 올해 3월부터 시작되었다.

FIRST 프로젝트는 한국전자통신연구원과 5개의 대학교가 공동으로 수행하는 프로젝트로서, 수행주체에 따라 2개의 세부 프로젝트로 나뉘지게 된다. 먼저 한국전자통신원의 경우 ‘FIRST@ATCA’라는 세부 과제 이름으로 미래인터넷 테스트베드용 대용량플랫폼을 개발하고 있으며, 대학들의 경우는 ‘FIRST@PC’라는 세부 과제 이름을 가지고 PC기반의 미래인터넷 테스트베드를 설계하고 구축하기 위해 노력하고 있다.

특히 FIRST@PC에서 충남대학교 팀은 오픈플로우(OpenFlow)^[5] 프로토콜을 지원하는 NetFPGA 기반 오픈플로우 스위치를 이용한 테스트베드 구축을 담당하며, 각 대학 간 구축된 소규모 테스트베드를 KOREN/KREONET 연구망을 이용하여 모두 연결하여 대규모 테스트베드가 되도록 하는 방법에 대해서 연구를 진행하고 있다.

FIRST@PC가 최종 목표로 하는 테스트베드의 형상은 모든 소규모 테스트베드가 서로 연결성을 갖게 되는 대규모 테스트베드를 구축하고 그 위에서 새롭게 디자인된 프로토콜 또는 시스템이 실험자의 의도에 따라 동작 및 테스트되고 테스트 결과가 실험자에게 보고 되도록 하는 것이다.

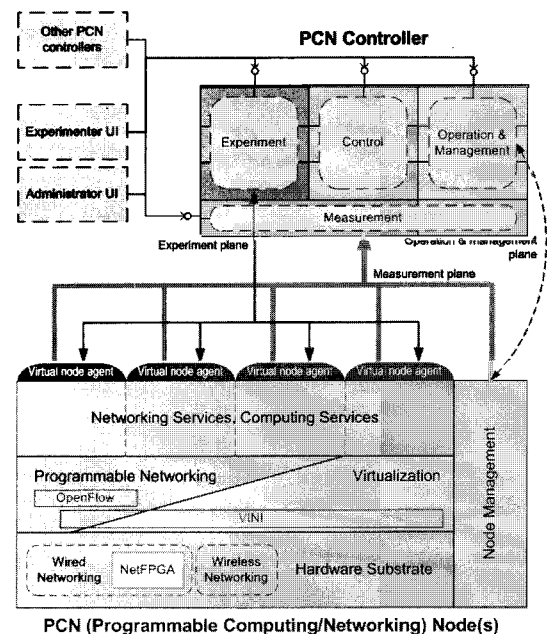
본 논문에서는 먼저 II장에서 미래인터넷 테스트 베드 구축 프로젝트인 FIRST 프로젝트에 대해 소개한다. 그리고 III장에서는 NetFPGA를 하드웨어 가속 장치로 활용하여 KOREN/KREONET 연구망에서 테스트베드를 구축하는 방법과 리눅스 기반의 raw socket과 libpcap 를 이용하여 MAC In IP 캡슐레이션을 지원하

는 소프트웨어 기반 캡슐레이터 구현 방법을 설명한다. 그리고 IV장에서는 소프트웨어 기반의 캡슐레이터를 이용하여 각 대학을 연동시키는 실험 및 수행한 TCP 실험 결과를 설명하고 마지막 V장에서 논문의 결론과 향후 연구 과제에 대해서 기술하고 본 논문을 끝맺도록 한다.

II. FIRST 프로젝트

1. FIRST 프로젝트

미국을 비롯한 여러 선진국에서 수행되는 미래인터넷 테스트베드 구축 연구가 우리나라에서도 현재 ‘FIRST’라는 프로젝트 이름으로 2009년 3월부터 한국전자통신연구원(이하 ETRI)과 국내 5개의 대학들(충남대학교, 광주과학기술원, 한국과학기술원, 포스텍, 경희대학교)이 공동으로 수행하고 있다. ETRI의 경우 ‘FIRST@ATCA’라는 세부 프로젝트 이름을 가지고 ATCA (Advanced Telecommunications Computing Architecture)기반의 하드웨어 플랫폼과 네트워크 프로세서(Network Processor: NP)를 이용하여 대용량을 지원하는 가상화된 프로그래머블 미래 인터넷 테스트베드용 플랫폼을 개발하고 있으며, 5개의 대학의 경우 “FIRST@PC”라는 세부 프로젝트 이름을 가지고 PC 기반의 소용량 미래인터넷 테스트베드용 플랫폼을 개발



PCN (Programmable Computing/Networking) Node(s)

그림 1. FIRST@PC에서 PCN 플랫폼 구조^[6]

Fig. 1. PCN platform architecture for FIRST@PC^[6].

하고 있다.

FIRST@PC의 하부 구조에 대해서 그림 1과 같이 VINI(Virtual Network Infrastructure)^[7] 형태와 하드웨어 가속 기능을 제공하는 NetFPGA 장비를 이용한 오픈플로우 스위치 형태의 2가지 하부 구조를 개발하고 있으며, 상위 구조에서는 PCN 컨트롤러라고 하는 서비스 합성 모듈이 사용자 및 운용자와 인터페이스를 갖고 능동적 (Dynamic)으로 사용자가 원하는 서비스를 지원하기 위한 하부 구조 리소스(Programmable Computing Networking Node)를 합성하게 된다.

추후 FIRST@PC와 FIRST@ATCA 모두 개발이 완료되면 두 플랫폼은 모두 KOREN/KREONET 연구망을 통해서 우리나라 미래인터넷 테스트베드라는 이름으로 테스트베드 역할을 수행하게 될 것이며, 액세스(Access) 네트워크와 코어(Core) 네트워크의 핵심 장비로 사용 될 것이다.

그림 2는 일단 FIRST@PC만 고려하였을 때, 각 대학에 구성된 소규모 테스트베드가 KOREN/KREONET과 같은 연구망을 통해서 서로 연결되어 전국 규모의 대규모 테스트베드로 구축되었을 때 전체 형상이다.

FIRST@PC 테스트베드는 실험자 인터페이스를 통해 실험자의 요청을 받게 되면 PCN(Programmable Computing/Networking Node) 제어노드를 통해 각 PCN노드 간의 능동적인 상호 연결성을 지원하도록 네트워크가 연결되는데, 만약 멀티미디어를 지원하는 서비스를 실험자가 요청하게 되면 PCN 제어 노드가 요청 사항에 맞는 리소스를 효율적으로 할당 및 예약하여 실험자가 요청한 서비스 실험이 네트워크상에서 잘 동작 되도록 한다. 이때에 네트워크 자원은 가상화된 리소스

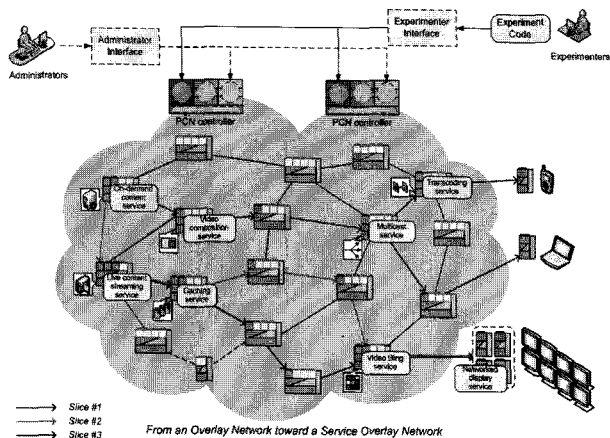


그림 2. FIRST@PC 테스트베드 모델^[6]
Fig. 2. The overall FIRST testbed model^[6].

장비, NetFPGA를 장착한 가속장비, 오픈플로우 스위치, 기존의 레거시 스위치/라우터 등이 될 수 있다.

III. KOREN/KREONET 기반 NetFPGA/OpenFlow 스위치 테스트베드 구현

1. KOREN/KREONET 소개

본 논문에서는 지역적으로 멀리 떨어진 테스트베드 연동을 위해 연구망인 KOREN^[8] (Korea Advanced Research Network)과 KRONET^[9] (Korea Research Environment Open NETwork : 국가 과학기술연구망)을 이용한다. 각 망 모두 비영리 목적의 순수 연구망으로서 광대역, 고품질의 국내외 연구시험망을 산·학·연에 제공하여 미래 네트워크 관련 기술의 시험검증과 첨단 응용분야 연구개발을 지원함으로써 연구개발촉진 및 국제공동연구 협력기반을 조성하기 위함을 목적으로 하고 있다.

KOREN 망의 특징을 간략히 살펴보면 그림3과 같이 전국 6개 대도시 지역(서울, 수원, 대전, 광주, 대구, 부산)을 10Gbps~20Gbps로 연결하는 백본망을 구축 운영하고 있으며, 국제연구망(APII, TEIN, JGNI, Internet2), BcN시험망 등과 연동되어서 향후 우리가 미래인터넷 연구의 핵심 기술 국가의 테스트베드들(GENI, FIRE, NwGN)과도 연동이 가능하다^[8].

KREONET의 경우 그림 4와 같이 대덕 연구단지 특화지역의 첨단연구개발을 지원하기 위하여 대덕 첨단 과학기술연구망(SuperSiReN)을 10Gbps 백본으로 구축하여 7개 기관(KISTI, KAIST, KBSI, KARI, KRIBB, KIGAM, CNU)을 연동하여 고성능 네트워크 인프라를

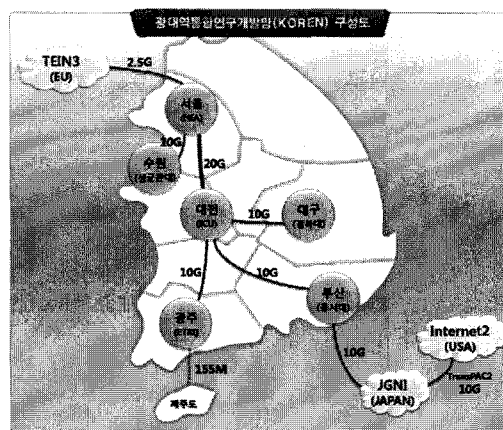


그림 3. 광대역 통합연구 개발망(KOREN) 구성도^[8]
Fig. 3. KOREN network topology^[8].

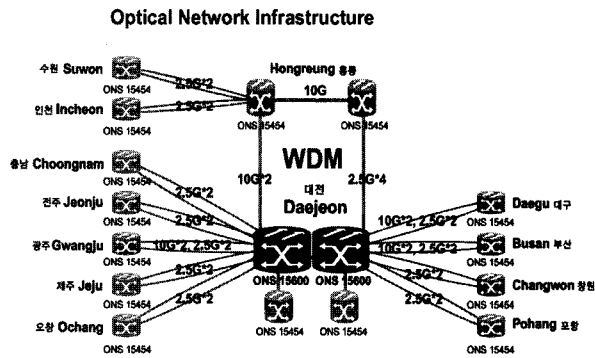


그림 4. KREONET 네트워크 구성도^[9]
Fig. 4. KREONET network topology^[9].

통한 첨단응용연구개발 및 시험 등을 지원한다. 또한 KREONET의 국제 연구망 서비스는 국제수준의 공동 협업연구 활성화를 위해 전세계 과학선진국(미국, 중국, 러시아, 캐나다, 네덜란드 등) 들과 긴밀히 협력하여 10Gbps급의 고성능 네트워크 인프라인 글로벌 과학기술협업연구망(GLORIAD)을 구축하고 이를 통해 첨단과학기술 응용연구 등의 연구개발을 지원하고 있다^[9].

2. NetFPGA 기반 오픈플로우 스위치

현재의 네트워크 장비는 내부와의 인터페이스를 지극히 제한하는 폐쇄적인 구조를 갖고 있어서 외부에서 장비 내부로의 인터페이스가 불가능한 상황이다. 비록 SNMP(simple network management protocol)가 존재 하기는 하지만, 이는 범용적인 용도의 네트워크 장비 관리 프로토콜이며, 라우터나 스위치 장비의 핵심인 포 워딩 테이블을 제어하여 실험자가 원하는 새로운 라우팅 프로토콜이라든지, 새로운 네트워크 구조를 실험할 수는 없다. 이에 NetFPGA^[10] 기반 오픈플로우 스위치가 프로그래머를 네트워킹을 바라는 최근 연구자들의 요구 사항에 맞추어 등장하였다. 오픈플로우 스위치는 그림 5와 같은 구성요소를 가지고 동작하게 된다.

● 플로우 테이블 (Flow Table)

각 플로우가 어떤 action(drop, forwarding, redirection)에 관계있는지, 어떻게 플로우를 처리할 지를 지정한다.

● 시큐어 채널 (Secure Channel)

원격 제어 프로세스인 오픈 플로우 컨트롤러와의 연결통로이다. 컨트롤러와 스위치사이에서 명령과 데이터 패킷들이 송수신된다.

● 오픈 플로우 프로토콜 (OpenFlow Protocol)

컨트롤러가 스위치와 통신할 때 사용되는 개방(open)되고 약속된 프로토콜이다.

오픈플로우 프로토콜이 동작하는 네트워크에서 모든 패킷의 경로는 컨트롤러의 제어를 받게 되는데, 예를 들어 A노드에서 B노드로 패킷을 송신하는 경우, 오픈 플로우 스위치에 패킷이 인입되며 오픈 플로우 스위치는 자신의 플로우 테이블을 룩업(lookup)하여 해당 패킷에 대한 action에 따라 drop, forwarding, redirection을 결정하여 처리하게 된다. 만약 처음 패킷으로써 플로우 테이블에 인입된 패킷과 관련하여 아무런 정보가 없는 경우에는 컨트롤러로 보내서 처리 방법을 결정 받아 처리하게 된다. 이와 같이 오픈플로우 프로토콜을 이용하게 되면 연구자는 패킷 하나하나에 대해서 원하는 대로 처리할 수 있는 방법을 얻게 된다.

본 논문에서 사용한 오픈플로우 스위치는 스탠퍼드

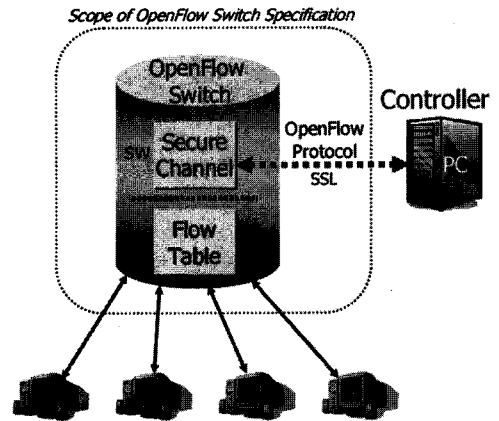


그림 5. 오픈플로우 스위치 구성요소^[11]
Fig. 5. OpenFlow switch components^[11].

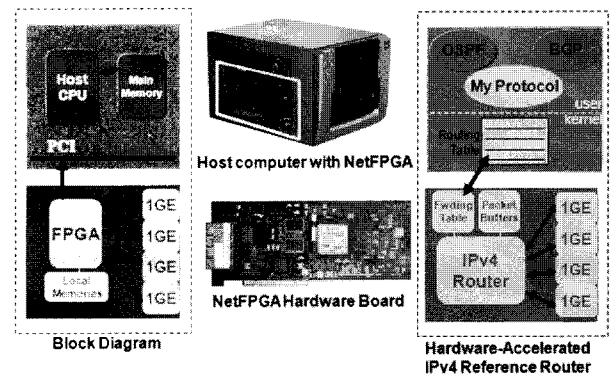


그림 6. NetFPGA 플랫폼 구조^[10]
Fig. 6. NetFPGA platform^[10].

대학에서 공개한 NetFPAG 기반 오픈플로우 스위치로서 그림 6과 같이 NetFPGA 카드와 Host PC로 구성된 NetFPGA 플랫폼을 이용하여 구현된다. NetFPGA 카드는 PC 에 사용할 수 있도록 PCI 인터페이스를 지원하며 FPGA (Field Programmable Gate Array)로 설계된 프로그래머블 하드웨어 가속 장비이기 때문에 새로운 네트워크 시스템 프로토타입을 설계하거나, 기존의 인터넷 장비의 성능을 측정하는 계측 장비 등으로 구현 가능하다. NetFPGA를 오픈플로우 스위치로 동작시키기 위해서는 “www.netfpga.org” 사이트에서 오픈플로우 스위치 비트파일을 다운받아 NetFPGA에 다운로드 시키면 되며, 이렇게 설정된 NetFPAG 기반 오픈플로우 스위치는 오픈플로우 컨트롤러의 제어에 받아 패킷들을 플로우 테이블에 적힌 패킷 action에 따라 처리하게 된다.

3. 어플리케이션 기반 소프트웨어 캡슐레이터

전국 단위의 미래인터넷 테스트 베드는 KOREN과 KRONET을 백본 망으로 이용하여 구축할 수 있다. 그림 4는 현재 소규모로 구축된 오픈플로우 테스트베드들을 연동한 네트워크 토폴로지로서, 충남대학교에 오픈플로우 컨트롤러인 NOX^[12]가 설치되어 있고, 포스텍과 광주 과학기술원에 각각 구축한 테스트 베드가 충남

대학교의 NOX에 의해 제어 받는 테스트 베드 형상이다. 구축된 테스트베드에서는 그림 7에서와 같은 3가지의 컴포넌트들이 구현 및 포팅 되어 이용되었다.

- 오픈플로우 컨트롤러 (OpenFlow Controller)

오픈플로우 컨트롤러는 오픈플로우 프로토콜을 이용하여 각 오픈플로우 스위치에게 포워딩 테이블을 내려주게 된다. 현재 테스트 베드에 사용한 오픈플로우 컨트롤러는 NOX를 이용하였다.

- 오픈플로우 스위치 (OpenFlow switch)

오픈플로우 프로토콜을 지원하는 스위치로써, 본 논문에서는 NetFPGA기반의 오픈플로우 스위치를 이용하였다.

- 캡슐레이터 (Capsulator)^[13]

연구망에 연결된 다른 오픈플로우 테스트베드를 연결할 때 사용되는 L2 프레임 포워딩 장비로써, 한 테스트베드의 MAC 프레임을 다른 테스트베드로 고속 포워딩 하는 역할을 하며 소프트웨어 기반의 캡슐레이터를 구현하여 이용하였다.

3.1 소프트웨어 캡슐레이터 동작

그림 8과 같이 프로토콜 스택을 갖는 소프트웨어 캡슐레이터는 하드웨어적으로 2개의 1Gbps 이더넷 카드

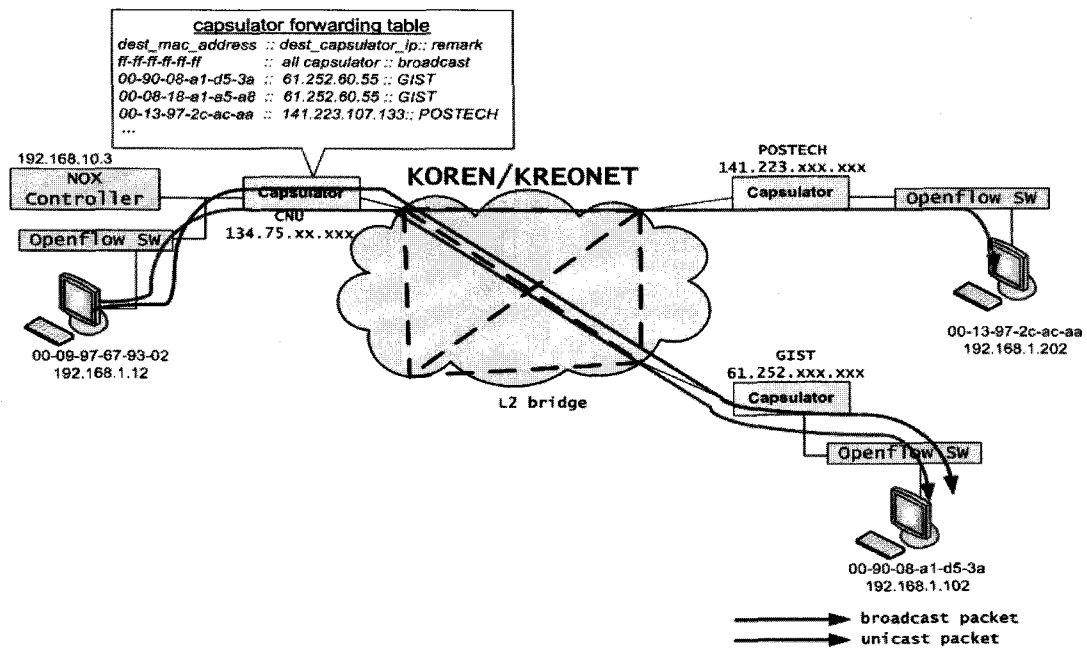


그림 7. KOREN/KREONET 망 기반 테스트베드 구성도
Fig. 7. Testbed network topology on KOREN/KREONET.

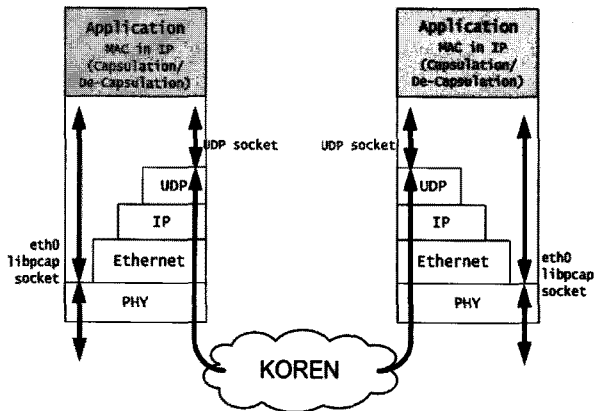


그림 8. 소프트웨어 캡슐레이터 동작
Fig. 8. Software capsulator Operation.

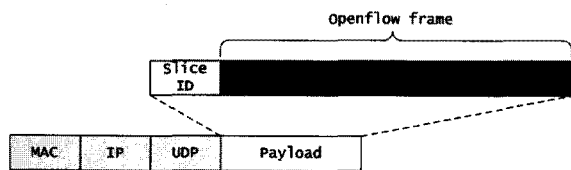


그림 9. 슬라이스 구분자
Fig. 9. slice ID.

가 장착된 하드웨어를 이용하여, 하나의 포트 (eth0)는 오픈플로우 테스트 베드에 연결되며 또 다른 포트 (eth1)는 연구망에 연결된다.

먼저 캡슐레이터의 eth0는 libpcap 라이브러리를 이용하여 구현되었으며, promiscuous 모드로 동작하면서 오픈플로우 테스트베드 네트워크의 모든 MAC 프레임에 대해서 캡처하게 된다. 캡처된 MAC 프레임은 어플리케이션 영역으로 전달되고, 전달된 MAC 프레임에는 그림 9와 같은 slice ID^[14]가 첨가된 후 UDP의 페이로드(Payload)영역에 실린 후 목적지 사이트의 캡슐레이터 IP를 획득하여 전달된다. 슬라이스 구분자는 추후 종단 간 트래픽 및 자원 분리 (Isolation)를 위해 사용될 구분자이다.

3.2 L2 switch emulation 기능 구현

소프트웨어 캡슐레이터는 KOREN/KREONET을 L2 스위치로 간주하게 하는 L2 스위치 에뮬레이션 (Emulation)을 수행 하는데, point-to-point 연결 방법과 bridge 연결 방법 2가지 모드를 지원한다.

point-to-point 연결 모드의 경우 지원하는 캡슐레이터는 하나의 테스트베드에서 발생한 MAC 프레임을 모두 설정 파일을 통해 획득된 목적지 테스트 베드로 직접 전달한다. 반면 bridge 모드로 동작하는 경우 캡슐레

이터는 기존의 L2 스위치 장비처럼 MAC learning 기능을 수행하여 자동으로 목적지를 학습하고 패킷을 스위칭하게 된다.

그림 10과 11은 bridge 모드로 동작하는 경우 캡슐레이터의 동작 절차이다. 먼저 그림 10의 전송 절차를 살펴보면 테스트베드로부터 MAC 프레임이 libpcap 소켓을 통해 eth0 포트로부터 획득되면 해당 프레임에서 목적지 MAC 주소가 ARP request와 같은 브로드캐스트 패킷이거나, 캡슐레이터에서 관리하는 포워딩 테이블에 관련된 목적지 정보가 없는 경우 캡슐레이터가 설정 파일로부터 획득한 KOREN/KREONET 상에 설치된 모든 캡슐레이터의 주소로 프레임을 브로드캐스트 전송한다.

공중망인 KOREN/KREONET에 연결된 eth1로부터 UDP 패킷 헤더로 캡슐레이션 된 패킷을 캡슐레이터가 수신하면, 캡슐레이터는 만약 자신의 포워딩 테이블에 전달되어온 패킷안의 소스 호스트의 정보가 없으면, 해당 사이트에 있는 호스트의 MAC 주소와 캡슐레이터 주소를 포워딩 매핑 테이블에 적어서, 다시 응답 패킷을 보내고자 할 때 기록해둔 정보를 이용하여 유니캐스트 (Unicast) 전송을 수행한다.

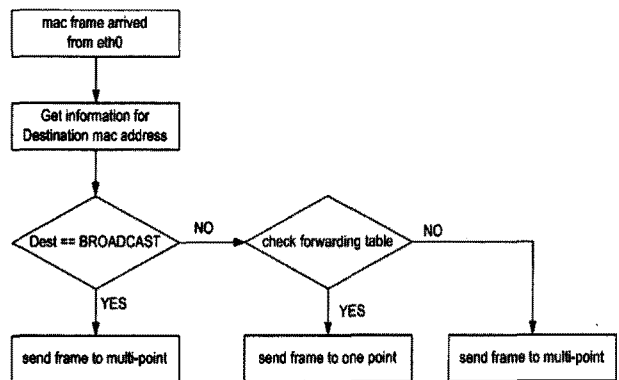


그림 10. 프레임 전송 절차
Fig. 10. Frame transmission procedure.

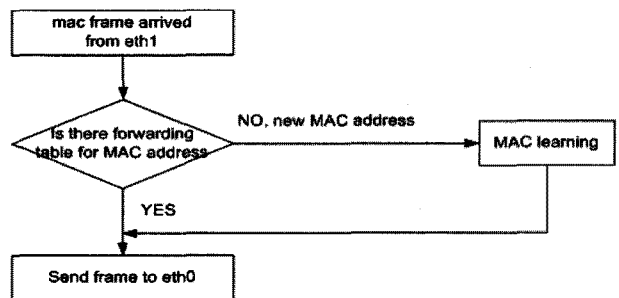


그림 11. MAC learning 절차
Fig. 11. MAC learning procedure.

예를 들어 그림 7과 같은 연동 시나리오를 갖는 상황에서 POSTECH 캡슐레이터에 CNU 호스트로 부터의 MAC-In-IP 패킷이 도착하면 캡슐레이터 포워딩 테이블에 도착한 MAC주소에 대해서 검사를 수행하고, 관련된 MAC 주소 정보가 없으면 해당 호스트의 MAC주소와 CNU 캡슐레이터 IP주소의 매핑 정보를 기록하여 현재 MAC 주소를 갖는 호스트는 CNU 캡슐레이터 밑에 존재한다는 정보를 기록하여 호스트와 캡슐레이터의 위치 관계를 학습하게 된다.

4. 연동테스트 결과

본 장에서는 구현한 소프트웨어 캡슐레이터의 포워딩 성능에 대해 로컬테스트와 실제 망에 연결해서 테스트한 결과를 설명한다.

4.1 로컬테스트 베드 구성 및 실험 결과

구현한 소프트웨어 캡슐레이터의 최대 전송성능을 테스트하기 위해 그림 12와 같은 실험 환경을 구축하였다. 캡슐레이터는 공중망 주소와 사설망 주소 각각 2개가 할당되었으며, 오픈플로우 프로토콜을 이용하기 위해 NetFPGA기반의 오픈플로우 스위치와 컨트롤러를 설치하고 오픈플로우 스위치 포트에 1Gbps 인터페이스로 사용자 호스트를 연결하여 각각 iperf 서버와 클라이언트를 동작 시켰다. 이때 TCP 테스트는 그림 13의 패킷 사이즈와 같이 64, 128, 256, 512, 1024, 1400, 1500 bytes로 페이로드 크기를 변경하며 각각 3600sec 씩 동작시켜 전달 성능을 측정하였다.

브릿지 모드를 동작시키며 로컬 테스트 베드에서 성능 테스트를 수행한 경우에는 3대의 캡슐레이터와 3대

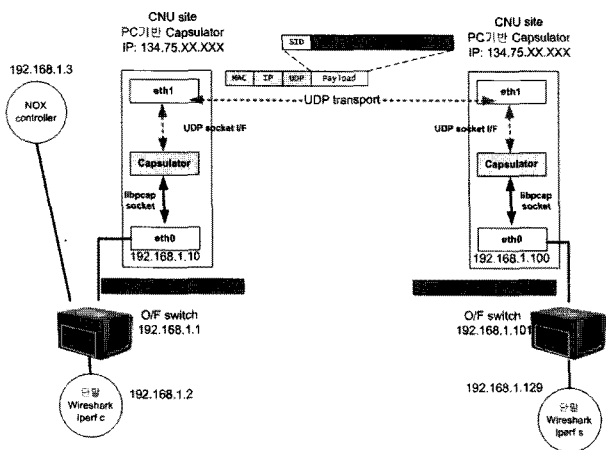


그림 12. 로컬 테스트 베드 구성도
Fig. 12. Local testbed topology.

의 오픈플로우 스위치를 이용하였으며, 그림 14와 같은 형상으로 구성하였다. 다만 로컬 테스트 환경을 구성하기 위해서 공중망 주소를 갖는 네트워크 인터페이스는 모두 1개의 같은 스위치에 연결되도록 구성하여 외부로부터의 패킷 유입과 유출을 차단하였다. 패킷 사이즈 별로 측정된 패킷 처리율 (Throughput)은 그림 13과 같다. 패킷이 작을수록 단위시간당 들어오는 패킷의 수가 많은 것을 의미하고 MAC-In-IP 패킷 터널링에 따른 오버헤드 비율로 커지며 더불어 처리해야할 패킷의 개수도 많기 때문에 페이로드 사이즈가 클수록 패킷 처리율이 높다. 다만 UDP 패킷이 캡슐레이션이 되므로 1500byte의 경우 MTU (Maximum Transmission Unit) 을 넘게 되어 패킷 단편화(fragmentation)의 영향으로 오히려 처리율이 떨어지게 된다.

point-to-point 모드와 bridge 모드 사이의 성능 저하는 현재 테스트 결과에서는 적지만, 만약 관리해야하는 호스트의 수가 늘어나서 포워딩 테이블이 커지게 되면

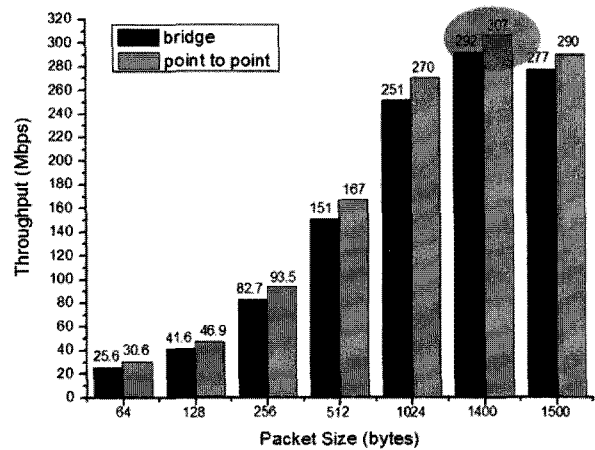


그림 13. 로컬테스트 결과

Fig. 13. local test results.

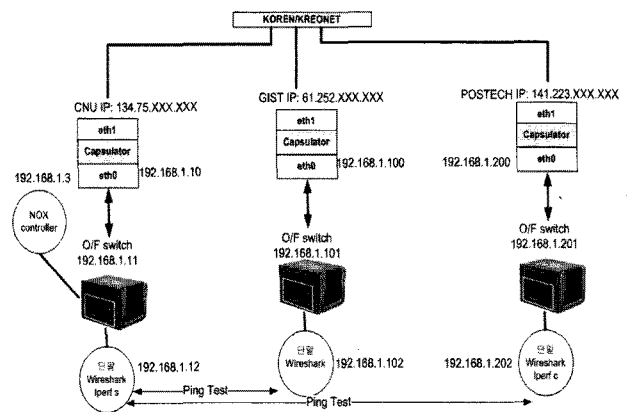


그림 14. 3개의 테스트베드 연동 토폴로지

Fig. 14. Interworking topology for three testbed sites.

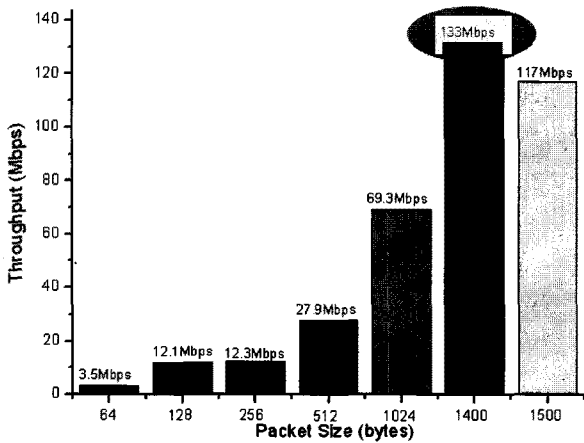


그림 15. 충남대와 광주과학기술원 연동테스트 결과
Fig. 15. Throughput result between CNU and GIST.

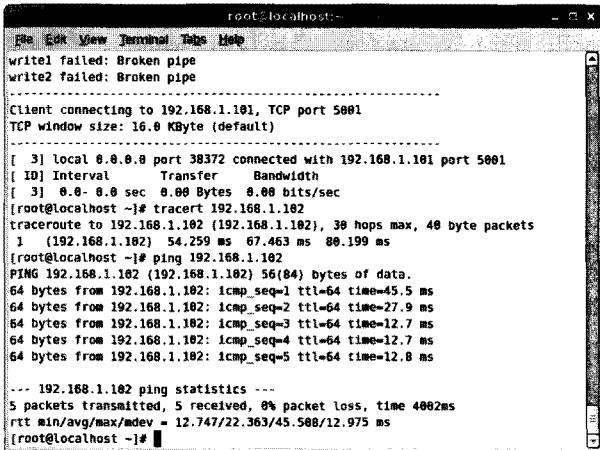


그림 16. 충남대와 광주과학기술원 Ping 테스트 결과
Fig. 16. Ping test result between CNU and GIST.

bridge 모드의 성능이 지금보다 저하될 것으로 예상된다. 그림 7과 그림 14는 충남대(CNU)와 포항공대(POSTECH)는 KREONET망을 이용하고, 광주과학기술원(GIST)는 KOREN망을 이용하여 서로 다른 테스트 베드가 연동될 때 오픈플로우 스위치와 캡슐레이터, 그리고 NOX 컨트롤러의 네트워크 토폴로지의 형상이다.

그림 15는 실제 연구망을 통해서 수행된 테스트 결과로써, 충남대 사이트의 단말(IP: 192.168.1.12)와 광주과학기술원 사이트의 단말(IP: 192.168.1.102) 사이에서 iperf를 이용한 TCP성능 테스트 결과이다.

앞의 그림 13에서와 같이 캡슐레이터의 최대 포워딩 능력은 약 300Mbps이지만 실제 망 테스트에서는 TCP 성능이 133Mbps에 밖에 이르지 않았다.

그 이유는 그림 16의 Ping 테스트결과에서 나타나듯 충남대학교는 KREONET에 연결되어 있고, 광주과학기술원의 경우 KOREN망에 연결되어 있어서 각 노드

의 종단 간 RTT (Round Trip Time)가 22.36ms로써 비교적 긴 값을 갖게 되어 TCP의 성능이 저하되었기 때문이다

IV. 결론 및 향후 연구

본 논문에서는 미래 인터넷 테스트베드 구축 프로젝트인 FIRST 프로젝트 중에서 5개의 대학을 중심으로 수행되고 있는 FIRST@PC에 대해 설명하였다. 그리고 연구 망 즉 KOREN/KREONET을 기반으로 MAC In IP 캡슐레이션 기능을 지원하는 캡슐레이터를 이용하여 따로 떨어져 있는 테스트베드들 간을 연동하여 대규모 미래인터넷 테스트베드를 구축하는 방법에 대해 설명하였다.

특히 본 논문에서는 리눅스 소켓 라이브러리를 이용한 소프트웨어 캡슐레이터를 이용하는 구현방법과 구현된 캡슐레이터를 이용한 로컬 연동 테스트를 수행하여 약 300Mbps정도를 지원함과 실제 망에 적용하여 테스트 한 결과 네트워크 지연에 의한 TCP 성능 저하로 인해 133Mbps의 결과를 나타냄을 보였다.

이와 같은 성능은 현재 존재하는 어플리케이션이 요구하는 서비스 대역폭을 어느 정도 충족할 수 있을 것으로 판단되지만 미래 네트워크에서 등장할 것으로 예상되는 대용량 대역폭을 요구하는 어플리케이션을 지원하기 위해서는 NetFPGA와 같은 하드웨어를 이용한 망 연동 장치가 필요할 것으로 예상하며, 이에 현재 연구를 NetFPGA 기반의 캡슐레이터를 구현하는 것으로 연구를 진행하고 있다.

참고 문헌

- [1] GENI: Global Environment for Network Innovations, <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] Jinho Hahm, Bongtae Kim, and Kyungpyo Jeon, "The study of Future Internet platform in ETRI", The Magazine of the IEEK, Vol.36, No.3, March, 2009.

[5] OpenFlow, <http://www.openflowswitch.org/>
 [6] SangWoo Han, Namgon Kim, JongWon Kim
 "Designing a virtualized testbed for dynamic multimedia service composition", Conference on the Future of the Internet, Proceedings of the 4th International Conference on Future Internet Technologies
 [7] Sapan Bhatia, Murtaza Motiwala, Wolfgang Muhlbauer, Vytautas Valancius, Andy Bavier, Nick Feamster, Larry Peterson, and Jennifer Rexford, "Hosting virtual networks on commodity hardware," Georgia Tech Computer Science Technical Report GT-CS-07-10, January 2008.
 [8] KOREN 소개
<http://www.koren.kr/koren/kor/abt/introduce.html>
 [9] KREONET 개요
http://www.kreonet.re.kr/01_info/01_info.php
 [10] NetFPGA, <http://www.netfpga.org/>
 [11] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner. OpenFlow: Enabling Innovation in Campus Networks. ACM Computer Communications Review, 38(2):69~74, 2008.
 [12] Natasha Gude, Teemu Koponen, Justin Pettit, Ben Pfaff, Martin Casado, Nick McKeown, and Scott Shenker, "NOX: towards an operating system for networks", ACM SIGCOMM Computer Communication Review, Vol.38, No.3, 2008.
 [13] Capsulator, <http://www.openflowswitch.org/wk/index.php/Capsulator>
 [14] flow visor, http://www.openflowswitch.org/wk/index.php/FlowVisor_config

저 자 소 개



박 만 규(학생회원)
 1999년 공주대학교 물리학과 학사
 2001년 공주대학교 전기전자 정보통신공학과 석사
 2001년~2004년 시스월 기업 부설 통신기술 연구소 전임연구원
 2004년~2006년 케이디넷 통신사업본부 기술팀 전임연구원/대리
 2007년~현재 충남대학교 전자전파정보통신 공학과 박사과정
 <주관심분야 : 이동통신 네트워크, 데이터 통신, 센서네트워크>



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



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



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 부교수
 <주관심분야 : 이동인터넷, 이동통신 네트워크, 데이터 통신>