특집

ETRI 미래인터넷 플랫폼 연구

함잔호 · 김봉태 · 전경표 (한국전자통신연구원)

Ⅰ. 미래인터넷 플랫폼 연구동향

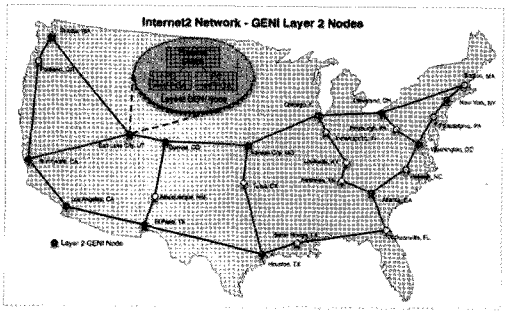
1. 미래인터넷 플랫폼의 필요성

미래인터넷 연구는 기존 인터넷을 대체하는 새로운 인프라를 중장기적으로 구축하고자 하는 연구이다. 지금까지의 대부분의 네트워크 연구는 네트워크 아키텍처를 제안하고, 구현 결과를 소규모 테스트베드를 통해 검증하고, 검증된 결과를 상용 인프라에 적용하는 형태로 추진되었다. 하지만, 이런 방법을 통해서는 연구결과가 대규모 실제 망에 적용되었을 때 나타나는 다양한 문제점들을 사전에 파악할 수 없고, 서비스, 컨텐츠, 단말, 사용자 및 사업자 등이 어우러져 나타나는 생태계적인 상호작용을 누릴 수 없었기 때문에 대부분 실패하였다.

따라서, 미래인터넷 연구에서는 GENI에서 시도하고 있는 바와 같이 아키텍처 연구를 통해 도출되는 다양한 아이디어가 실제 환경과 유사한 대규모 시험 인프라에서 검증되고, 자연스럽게 공중망으로 옮아갈 수 있는 방법론을 적용하고 있다.

미래인터넷 시험 및 서비스를 위한 인프라가 구축되기 위해서는 라우터와 같은 기능을 하는 플랫폼이 필요한데, 기존 라우터에서는 이미 확정된 IP 인터넷 프로토콜만이 탑재 운용되는 데 반하여, 미래인터넷 플랫폼은 향후 도출될 다양한 non-IP 프로토콜을 탑재하여 시험 및 운용해야 한다는 점이 다르다.

아래 <그림 1>은 GENI에서 추진 중인 미래인터넷 시험 인프라의 모습을 보여주고 있다. GENI에서는 Internet2, NLR(National Lambda Rail)과 연계하여 미국 전역을 커버하는 시험 인프라를 구축할 계획이다. 중대형 미래인터넷 플랫폼이 각 노드에 설치 운용된다.



〈그림 1〉 GENI에서의 미래인터넷 시험 인프라의 구축

2. GENI에서의 미래인터넷 플랫폼 연구 현황

미래인터넷이 어떤 구조를 가져야 할지는 아직 구체적으로 정의되지 않고 있다. 이것은 향후 수십 년간 사용될 미래인터넷의 모습이 섣불리 초기 단계에서 어떤 특정 구조로 편향되는 것을 원하지 않기 때문으로, 다양한 아이디어가 서로 비교되고, 경쟁하는 가운데, 자연스럽게 최종적인 미래인터넷 모습이 드러날 수 있을 것이라는 것이 미래인터넷을 이끌고 있는 사람들의 생각이다.

GENI의 미래인터넷 플랫폼 연구 역시 이와 같은 철학을 바탕으로, 아래 <그림 2>에서처럼 5개의 클러스터 연구가 서로 경쟁하는 가운데 추진되고 있다. 각 클러스터는 여러 프로젝트로 구성되어 있고(총 29개의 프로젝트 추진), 각 프로젝트에는 5~6개의 대학 및 연구소, 업체가 참여하고 있다.

이들 클러스터는 각기 나름대로의 컨트롤 프레임워크를 정의하고 있는데, GPO(GENI Project Office)는 클러스터간의 경쟁을 통해, 이 과정에서 얻게 되는 경험과 역동성을 바탕으로 플랫폼 구조 및 컨트롤 프레임워크의 모습이 결정되기를 원하고 있다. 최종적인 플랫폼 구조 예측의 어려움과 다양한 아이디어의 수용을 위해 GENI에서는 Spiral Model을 연구방법론으로 채택하고 있다.

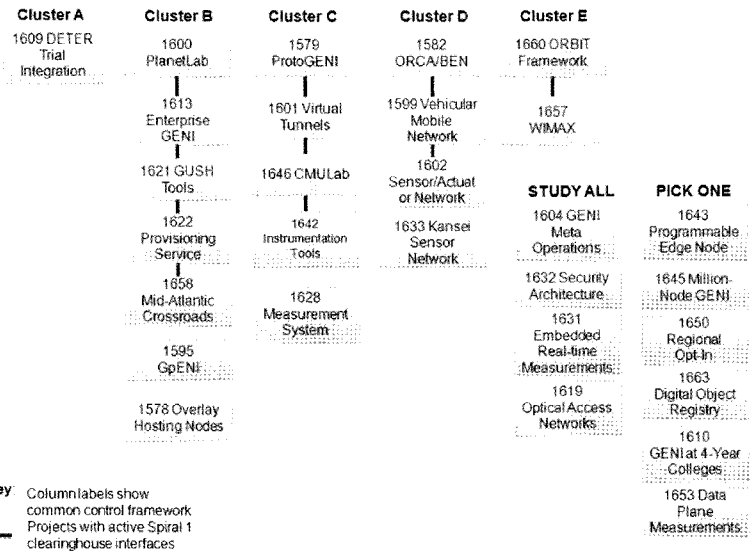
GENI는 연구과정에서 얻게 되는 경험을 서로 공유하기 위하여 4개월마다 GEC(GENI Engineering Conference)를 개최한다. 제4차 GEC가 '09년 3월말 마이애미에서 개최될 예정이며, 플랫폼 개발결과에 대한 데모도 함께 있을 예정이다.

3. GENI 미래인터넷 플랫폼 설계 개념

GENI 플랫폼 설계는 다음과 같은 몇 가지 개념에 기반을 두고 있다.

- Virtualization

GENI 시험 인프라에서는 완전히 다른 다수



〈그림 2〉 GENI Spiral 1에서 추진 중인 5 가지 클러스터 연구의 프로젝트 구성

의 아키텍처 실험이 동시에 이루어져야 한다. 실험을 위해 물리적인 인프라를 다수 운용할 수 없으므로, 물리적인 자원을 가상화하여 사용자 각자에게 시험 인프라를 독점하는 것처럼 보이게 하는 가상화가 필요하다.

- Programmability

인터넷 아키텍처 연구자들에게 자신의 아키텍처 실험을 위하여 플랫폼의 세부적인 기능을 구현하도록 요구할 수는 없다. 따라서, 원하는 아키텍처를 쉽게 구현할 수 있도록 모듈화된 기능 블록을 제공한다.

- End-to-End Slice

슬라이스란 미래인터넷 플랫폼이 제공하는 기능 모듈(프로토콜이 탑재된다)을 연결하여, 오버레이 네트워크 형태로 만들어진 가상 미래인터넷 인프라이다. 이를 위해 필요한 시험 개수만큼 슬라이스가 시험 인프라에서 만들어지게 된다. 플랫폼은 슬라이스의 생성, 변경, 테스트 및 점유한 자원의 해제 등의 과정을 지원한다.

- Federation

미래인터넷은 궁극적으로 전 세계를 커버하는 네트워크로 발전하여야 한다. 이를 위해 각자 자신의 미래인터넷 시험 인프라를 구축하고 이들을 서로 연합(federation)함으로써 글로벌 시험 인프라로 확장하는 방식을 채택한다.

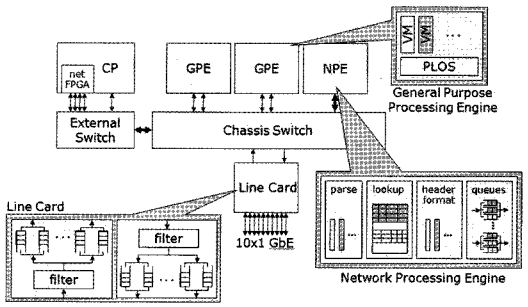
- Network Resource 관리

슬라이스를 구성함에 있어서, 실제 망을 운용하는 것과 유사하도록 자원을 관리할 필요가 있다. 패킷 처리 성능은 실제 이를 담당하는 물리적인 자원인 대역폭, CPU, 메모리, 패킷 처리 모듈 등에 의존적이다. 따라서, 플랫폼에서는 슬라이스 구성 및 운용을 통해 네트워크 자원을 정량적으로 타이트하게 관리할 수 있어야 한다.

4. SPP Platform

아래 <그림 3>은 GENI Cluster B에서 개발되고 있는 SPP 플랫폼으로 Washington University in St. Louis의 조나단 터너 교수가 개발하고 있는 플랫폼이다. SPP는 ATCA 규격을 채택하고 있으며, Intel IXP 28xx 네트워크 프로세서를 사용한다.

SPP 플랫폼은 CPU 기반의 GPE (General Processing Engine)과 Intel NP 기반의 NPE (Network Processing Engine) 및 라인카드로 구성되어 있다. 라인카드를 통해 패킷이 들어오면 처리 모듈이 어디에 존재하는 가에 따라 Slow path와 Fast path로 나뉘어지게 되는데, 라인카드나 NPE에서 기능 모듈이 구현되어 있다면 Fast path를 통해 패킷이 고속으로 처리되고, 기능이 복잡하거나 성능 개선이 절실하지 않은 경우에는 Slow path를 통해 GPE에서 처리된다.



〈그림 3〉 SPP 플랫폼의 상세도

SPP에서 가장 중요한 부분은 NPE로서 Intel 네트워크 프로세서를 기반으로 마이크로 프로그래밍 레벨에서 구현되는데, 마이크로 코어의 프로그램 사이즈의 한계 및 파이프라인 형태의 프로그

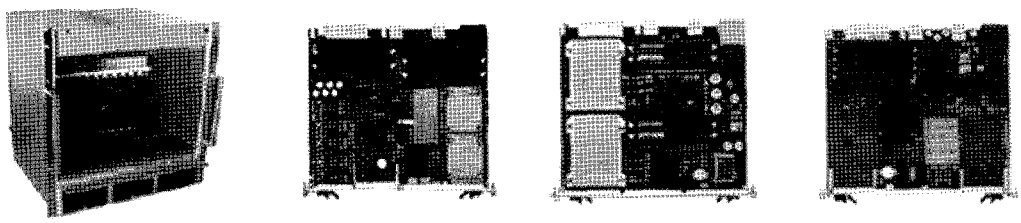
래밍 복잡도로 인해 모듈 개발이 용이하지 않다.

GENI에서는 두세 개의 SPP 노드를 2009년도 중반에 릴리즈 할 예정이며, 여기에는 GENI와 호환성 있는 컨트롤 소프트웨어가 탑재될 것이다. 최종 플랫폼은 2010년 말에 릴리즈 될 예정으로, NetFPGA 기능이 함께 제공된다.

II. 미래인터넷 플랫폼 요구사항

미래인터넷 플랫폼은 향후 정의될 패킷도 처리 가능하도록 확장 가능한 유연한 구조를 가져야 하며, 실험 과정에서 발생하는 다양한 처리 결과를 세밀하게 모니터링함으로써 설계상의 문제점들을 개선해 나갈 수 있어야 한다. GENI가 제시하고 있는 플랫폼 요구사항은 다음과 같다.

1) 동시에 1,000개 이상의 실험을 수행할 수 있어야 한다. 이는 한 플랫폼에서 수백 개의 슬리버 모듈을 생성할 수 있어야 함을 의미한다.
2) 슬라이스 생성을 통해 연결성을 지원할 수 있어야 한다. 플랫폼에서 네트워크 자원을 발견하고, 이를 기반으로 슬리버를 만들고, 슬리버 간의 연결을 통해 토폴로지를 설정하고, 슬리버에 원하는 프로토콜을 탑재할 수 있어야 한다.
3) 서로 다른 실험들이 영향을 미치지 않도록 슬라이스간 강력한 격리를 지원할 수 있어야 한다. 슬라이스는 시험을 위한 가상화된 오버레이 네트워크이므로 물리적인 자원을 공유하고 있다.
4) 기존 인터넷에 대한 접속을 지원할 수 있어야 한다. 미래인터넷은 이미 구축된 인프라와 사용자를 활용할 수 있어야 한다. GENI에서는 이를 위해 Stanford 대학에서 제안된 OpenFlow 방식을 사용하려 하고 있다.
5) 다양한 패킷 포맷을 처리할 수 있어야 한다. 미래인터넷 아키텍처에서는 현재 및 미래의 요구사항을 해결하기 위하여 clean slate 기반의 새로운 패킷구조와 프로토콜을 정의할 것이다.
6) 지형학적인 거리의 두 배 이상의 종단간 지연이 발생해서는 안 된다. 토폴로지의 구성과 패킷 처리가 효율적으로 수행되어야 한다.
7) 가상화 기능을 지원하는 컴포넌트에 대하여 의도적으로 장애를 발생시킬 수 있어야 한다. 시험은 보다 가혹한 환경에서 이루어질 필요가 있다.
8) 자신의 실험을 모니터링하고, 디버깅할 수 있는 풍부한 도구를 지원해야 한다.



〈그림 4〉 ETRI 미래인터넷 플랫폼 하드웨어 구성

III. ETRI 미래인터넷 플랫폼 연구

ETRI는 2009년 3월부터 '가상화 기능을 갖는 프로그래머블 미래인터넷 플랫폼' (이하 ETRI 플랫폼) 개발에 착수하였다. ETRI 플랫폼은 우리나라 미래인터넷 시험 인프라 구축을 위한 기본 장비로 활용될 예정이며, 이를 통해 미래인터넷 아키텍처 연구자들이 제안하는 다양한 아키텍처를 쉽게 구현하고, 성능 및 기능 검증을 통해 보다 우수한 아키텍처로 개선되기를 기대하고 있다.

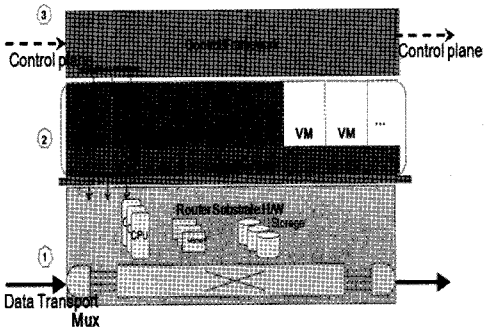
1. ETRI 플랫폼의 하드웨어 구성

ETRI 플랫폼은 크게 하드웨어와 소프트웨어 부분으로 나뉘는데, 하드웨어 플랫폼은 ATCA 기반의 COTS(Commercial Off The Shelf) 제품을 채택하였다. 이렇게 함으로써 하드웨어 개발에 소요되는 시간을 단축할 수 있다. ATCA(Advanced Telecom Computing Architecture)는 PICMIG (PCI Industrial Computer Manufactures Group)에서 제정한 규격으로 컴퓨팅과 통신이 결합된 형태의 장비를 지원하며, 이미 많은 상용 블레이드가 출시되어 있다.

ETRI 플랫폼은 GPE를 담당하는 Xeon 기반의 CPU 보드, NPE와 라인카드 기능을 담당하는 Octeon NP 기반의 패킷처리 블레이드로 구성되며, 이들 블레이드 간의 패킷 교환을 위해서 10GbE 스위치가 사용된다. 하드웨어 플랫폼은 향후 소프트웨어 개발이 완료된 후, 최신 버전의 COTS 블레이드나 특화된 기능을 갖는 자체 개발된 하드웨어 블레이드로 교체될 것이다.

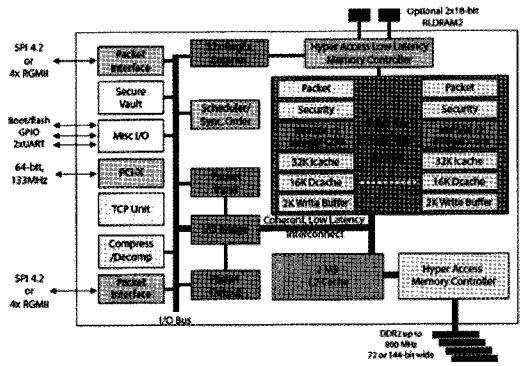
2. ETRI 플랫폼의 소프트웨어 구성

ETRI 플랫폼의 소프트웨어 개념 구조는 <그림 5>와 같다. 가장 하부인 Substrate 계층에서는 CPU, 라인카드, 스위치 보드 차원에서 미래인터넷 플랫폼으로서의 기본적인 기능이 처리된다. 물리적인 네트워크 자원 관리, 어드레스 룩업, 패킷의 분류(classification), 상황 모니터링 등이 여기에 포함된다. 이들 기능은 API 형태로 구현되어 상위 계층에서 이들을 묶어 원하는 보다 포괄적인 기능을 구성할 수 있도록 제공된다.



〈그림 5〉 ETRI 미래인터넷 플랫폼 소프트웨어 구조

중간의 가상화 계층에서는 하부 물리적인 자원들을 논리적인 구조로 매핑하는 기능을 제공한다. 네트워크 연구자는 가상화 계층을 통해 자신이 이용하는 네트워크 자원을 다른 연구자가 이용하는 네트워크 자원과 상충됨이 없이 독립적으로 점유한다.



〈그림 6〉 Octeon 프로세서의 구조

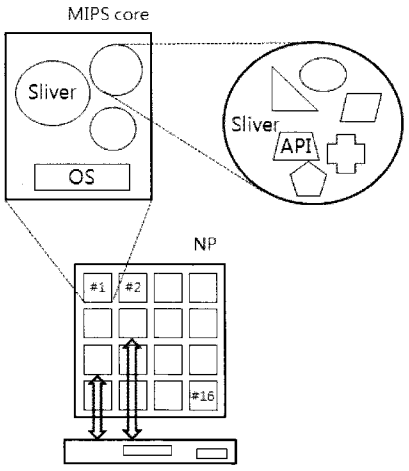
상위 컨트롤 프레임워크는 네트워크 아키텍처 연구자들이 슬라이스를 구성할 자원을 검색하고, 슬라이스를 생성하고, 토폴로지 생성 및 시험을 수행하고, 시험 결과를 분석하는 일련의 처리과정을 지원한다.

3. Octeon NP를 기반으로 한 패킷처리 모듈의 구현

ETRI 플랫폼에 장착되는 NPE와 라인카드는 Octeon 네트워크 프로세서를 기반으로 하고 있다. Octeon 네트워크 프로세서에서는 입력 패킷에 대하여 고속으로 처리되어야 할 기능들이 구현된다. 입력 패킷에 대한 분류, 패킷 헤더에 대한 분석, 포워딩 테이블 룩업, 큐잉, 스케쥴링 등이 NP에서 처리되어야 할 fast path 기능들이다.

Octeon 프로세서는 16개의 MIPS 코어가 있으며, 이들 각각의 코어에 OS를 탑재하거나 SE(Simple Executive) 모드로 실행시킬 수 있다. 앞에서 설명한 패킷처리 기능 모듈인 슬리버를 16개의 코어에 적절하게 할당해야 균등한 부하를 최적의 성능을 이끌어 낼 수 있다.

앞의 <그림 6>과 다음 <그림 7>은 Octeon 프로세서의 내부 구조와 프로세서 내 각각의 코어에 할당된 슬리버를 보이고 있다. 플랫폼에서 다양한 미래인터넷 아키텍처를 지원하도록 하기 위해서는 여러 아키텍처 실현에 공통적으로 적용되는 API를 정의하여 설계할 필요가 있다. 슬리버는 여러 API 처리 모듈을 수용하여 실행 프로그램으로 구성된다. Octeon 프로세서 내에 16개 코어가 있으므로, 기본적으로 프로세서 당 16개의 슬리버를 운용할 수 있으나, 각 슬리버에서 요구되는 처리 용량은 다양할 수 있으므로, 하나의 코어에 여러 개의 슬리버를 할당(스케일 다운)한다든지, 여



〈그림 7〉 Sliver와 API의 구성

러 개의 코어가 하나의 슬리버 기능을 담당(스케일 업)하게 하는 등의 업무 할당이 필요하다.

4. 플랫폼 개발에서의 기술적인 주요 연구 이슈

플랫폼 개발을 위해 해결하여야 할 주요한 기술적인 이슈는 다음과 같다.

- Physical Resource 정량적인 관리

  CPU, 메모리, 전송대역폭, 버퍼, Network processor 등 물리적인 자원의 정량적인 처리에 대하여는 규격 수준에서 정의되어 있을 뿐, 구현 방안이 다각도로 검토되고 있다. 이의 처리를 위해서는 가용한 자원 여부에 대한 상세한 모니터링이 수반되어야 하고, 할당된 자원의 실행을 정량적으로 제어할 수 있어야 한다. 예를 들자면, 슬리버 당 CPU의 10% 만을 사용한다거나, 패킷처리 성능을 100kpps (Packet per Sec) 로 제한하는 등의 제어가 필요하다. 이를 위해서는 컴포넌트 레벨에서 제공하는 low level 모니터링, OS에서 제공하는 다양한 측정 기능을

rspec subtype representing a slice: (name, value)

vm_type : *linux-vserver??*

cpu_share : proportional share CPU scheduler, *currently all slices get equal share*

mem_limit : per node upper bound on memory, *currently no bound is specified*

disk_quota : per node upper bound on disk space used, *current ??*

base_rate : *default 1Kbps*

burst_rate : *default none*, so can burst at the full available rate.

sustained_rate : *default 1.5Mbps*. Limits sustained sending rates over an extended period of time, currently 24 hours (*24h·60m/h·60s/m·1.5Mb/s·1B/8b = 16.2GB/day*). After this the VM is limited to 1.5Mbps

〈그림 8〉 PlanetLab에서의 Rspec의 예

조합하는 것이 필요하다.

<그림 8>에서 PlanetLab에서 정의한 Rspec (Resource Specification)의 예를 보이고 있다.

- 멀티코어 프로그래밍

  Octeon 프로세서는 16개의 코어를 갖고 있으므로, NP에 할당된 슬리버가 최대한의 성능을 발휘하도록 하기 위해서는 슬리버 및 내부 API 처리 모듈에 적절하게 부하를 할당하는 것이 필요하다. 이에 대하여는 모듈 설계 시부터 프로세스 생성, 스레드 스케쥴링과 패킷 처리의 동기화 등이 고려되어야 한다.

- Resource의 평탄화된 사용

  플랫폼에서 새로운 슬리버를 생성할 때 현 상황에서의 물리 자원의 여유도를 감안하게 되므로, 어떤 슬리버의 자원 점유가 예측 불가능하도록 가변적으로 변동하여서는 문제가 있다. 따라서, 슬리버 프로그램에서는 리소스가 점유가 가급적 확정적으로 이루어지도록 설계되어야 한다.

- Programmability를 최적화하기 위한 API의 선정

  일반적으로 API 결정은 어떤 처리에서 공통적으로 채택되는 기능을 추출하여 이루어진다. 미래인터넷 플랫폼에서의 API는 미래인터넷의 전체 아키텍처 모습이 결정되지 않은 상황에서도 미래지향적으로 결정되어야 한다.

- Robustness 확보를 위한 물리적 자원간의 격리

  슬리버에서 발생한 문제는 슬라이스로 연결된 다른 노드의 슬리버에 영향을 미치고, 해당 노드에서의 물리자원을 공유하는 다른 슬리버에 영향을 미치고, 또 해당 슬리버와 연결된 다른 슬라이스에 영향을 미침으로써, 문제가 시험 인프라로 전체로 확산될 수 있다. 따라서, 가상화 처리에서 물리적인 자원의 격리는 매우 중요하다.

IV. ETRI 플랫폼 연구개발 전략

미래인터넷 플랫폼 연구는 개방적으로 추진되어야 하며, 따라서 국제 선도 그룹과의 적극적인 기술교류 및 공조가 무엇보다도 필요하다. 이를 위해 ETRI는 GPO (GENI Project Office)와 GENI 프로젝트에 참여하는 방안을 협의 중에 있으며, 특히 GENI Cluster C 그룹인 ProtoGENI와의 긴밀한 협력을 모색하고 있다. 또, 한국과 미국 시험 인프라간의 연결을 위하여 KISTI와

함께 Internet 2 주관기관인 인디아나 대학과 협력 방안을 논의 중에 있다.

미래인터넷 인프라를 위한 RFC와 같은 표준 규격은 아직 존재하지 않지만, GENI에서는 앞서 설명한 것처럼 클러스터 별로 각각의 컨트롤 프레임워크를 정의한 가운데, 경쟁과 협력을 통해서 향후 단일 컨트롤 프레임워크로 자연스럽게 통일되기를 바라고 있다. ETRI는 미래인터넷 플랫폼을 개발하는 과정에서 일단 GENI 컨트롤 프레임워크를 채택하여 클리어링하우스, 미래인터넷 플랫폼, 사용자 응용 간의 슬라이스 생성 및 운용을 위한 제어 절차를 설계할 예정이며, 설계 및 구현 과정에서 나타나는 이슈 및 문제점에 대하여 GENI 디자인그룹에 적극적으로 제안함으로써 우리의 의견을 반영해 나갈 예정이다.

이와 함께, ETRI 플랫폼 기반의 국내 미래인터넷 시험 인프라와 GENI 시험 인프라간의 federation도 적극적으로 추진해 나갈 예정이며, 이를 위해 아직 별다른 진전이 없는 GENI의 federation 규격 작업에도 참여할 예정이다.

총 5년간 추진될 ETRI의 미래인터넷 플랫폼 연구에서, 1단계 연구가 마무리되는 2011년 말이면 국내 일부 네트워크 연구 베타사용자들은 ETRI 플랫폼을 기반으로 슬라이스 제어에 대한 기본 기능을 사용할 수 있을 것이다. 또한, 프로젝트가 종료되는 2013년 말이면 상용서비스 바로 전 단계에 사용될 수 있는 수준의 플랫폼의 역할을 담당할 수 있도록 기능이 확장될 것이다.

ETRI의 미래인터넷 플랫폼 연구(FIRST: Future Internet Research for Sustainable Testbed)는 내부적으로 앞서 설명한 ATCA 샤시 기반의 중대형 플랫폼을 개발하는 연구(FIRST@ATCA)와 광주과학기술원을 공동연구기관으로 하여 5개 대학이 함께 참여하는 NetFPGA를 활용하는 PC 기반의 미래인터넷 플랫폼 연구(FIRST@PC)로 구성되어 있다. 이 두 종류의 플랫폼은 향후 국내 미래인터넷 시험 인프라를 구축하는 단계에서 코어망과 가입자망의 인프라 장비로 활용될 것이다. 또한, 이 두 망을 연결하는 액세스 망을 위한 플랫폼으로서 Octeon NP 기반의 단일 보드 소형 플랫폼을 개발할 계획도 아울러 가지고 있다. 아직까지 코어망(필요하다면 액세스망까지)의 하부구조로서 람다 스위칭의 채택을 구체적으로 검토하지는 않았지만 시험 인프라의 구축이 구체화되는 단계에서 이에 대한 논의가 필요할 것으로 생각된다.

참고문헌

[1] GENI home page, http://www.geni.net/
[2] PlanetLab, http://www.planet-lab.org/
[3] ProtoGENI, http://www.protogeni.net/trac/protogeni
[4] ATCA specification, http://www.intel.com/technology/atca/
[5] Octeon multi-core Network Processor, http://www.caviumnetworks.com/

저자소개

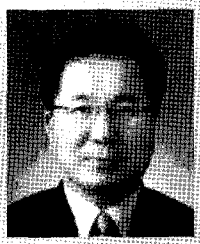
함 진 호

1982년 2월 한양대학교 전자공학과 학사
1984년 2월 한양대학교 전자통신공학과 석사
1998년 2월 한양대학교 전자통신공학과 박사
1984년 3월 한국전자통신연구원 입소
2003년 2월 ~ 2005년 1월 차세대라우터S/W팀장
2008년 3월 ~ 2009년 2월 미래네트워크연구팀장
2009년 3월 ~ 현재 미래인터넷연구팀장
현재 미래인터넷포럼
Architecture Working Group 의장
미래인터넷포럼 FRC 간사

김 봉 태

1983년 2월 서울대학교 전자공학과 학사
1983년 3월 한국전자통신연구원 입소
1991년 12월 미국 NC 주립대학(NCSU) 컴퓨터공학 석사
1995년 12월 미국 NC 주립대학(NCSU) 컴퓨터공학 박사
2004년 3월 ~ 현재 광인터넷포럼 운영위원장
2008년 7월 ~ 현재 한국전자통신연구원
네트워크연구본부장

저자소개



전 경 표

1976년 2월 서울대학교 산업공학과 학사 (B.S)
1979년 2월 한국과학기술원 산업공학과 석사 (M.S.)
1988년 5월 North Carolina State Univ.
Operations Research 전공 박사 (Ph. D.)
2004년 1월 ~ 2008년 2월 한국전자통신연구원
광대역통합망(BcN)연구단장
미래인터넷포럼 FRC (Future Internet
Research Committee) 공동의장
2008년 2월 ~ 현재 한국전자통신연구원 연구위원