

미래 인터넷 연구 동향과 관리기능 정의

김성수, 최미정, 홍원기

포항공과대학교 컴퓨터공학과

{kiss, mjchoi, jwkhong}@postech.ac.kr

요 약

현재의 인터넷은 역사상 그 어떤 기술보다 성공적이었고, 우리의 삶에 있어서 없어서는 안될 필수 불가결한 요소가 되었다. 그러나 현재 인터넷의 전 세계적인 보급과 폭발적인 사용은 동시에 IP주소 부족, 보안, 성능 등 여러 문제를 야기시켰고, 이러한 문제를 점진적으로 해결하기 위한 기술들은 또 다른 문제를 일으켰다. 이제 인터넷 아키텍처 자체를 새롭게 설계하는 방안에 대해 생각해야 할 시기가 되었고, 실제로 세계의 여러 연구 단체들은 현 인터넷의 다음 버전으로 미래 인터넷이라는 용어를 사용하여 아키텍처 설계와 구현에 대해 연구를 진행하고 있다. 본 논문에서는 미래 인터넷이 해결해야 할 과제와 요구사항, 그리고 FIND, GENI, JGN과 같은 미래 인터넷 연구 단체들에 의해 현재 진행되고 있는 미래 인터넷 아키텍처와 기반 기술에 대한 연구들을 바탕으로 미래 인터넷 관리를 위한 요구사항들과 그에 따른 관리기능들을 제안하고자 한다.

1 서론

인터넷은 사람들의 의사 소통, 기업의 업무 그리고 군대의 운용을 비롯한 다양한 분야에 적용되어 널리 사용되고 있다. 인터넷은 또한 사람들 사이의 상호작용과 사람과 컴퓨터 간 상호작용에 대해 새롭게 정의했다. 광범위한 인터넷의 사용은 인터넷의 엄청난 성공을 반영하고 있지만 동시에 단순히 세계에 단지 몇 개의 슈퍼컴퓨터가 존재하고, 그 컴퓨터들을 연결하던 초기 인터넷 시절과는 비교할 수 없을 만큼 복잡해지고 다양해졌다.

여러 연구 단체들은 현재 인터넷의 성공이 현재사회에 어떤 영향을 미쳤는가에 주목해 왔다. 그러나 동시에 이러한 성공은 현재 인터넷이 계속해서 증가하고 있는 이용자, 네트워크 장비, 서비스, 어플리케이션 등에 대한 확장성의 문제에 직면했다는 사실을 명백하게 나타내고 있다 [6], [7]. 또한 전세계의 여러 연구 단체들은 현재 인터넷의 문제점에 주목하고 있다. 현재 인터넷이 직면한 문제를 점진적으로 해결하기 위해 통신 프로토콜의 보완에 관한 연구에서부터 완전히 새로운 인터넷을 설계하고 배치하는 급진적인 연구까지 다양한 연구들이 진행되고 있다. 대부분의 인터넷 관련 문제들은 인터넷의 급격한 이용 증가로부터 발생하며 많은 수의 이용자가 연관되어 있으므로 문제들을 신속히 해결해야만 한다. 필연적으로 인터넷의 문제를 신속히 해결하여 끊임 없는 서비스를 제공하기 위해 근본적인 인터넷 아키텍처에 대해 재고할 여유 없이 임시적인

솔루션을 제공하는 데만 급급했었다. 따라서 네트워크 연구를 하는 많은 사람들은 현재 인터넷 아키텍처의 한계를 인식하고 장기적인 관점에서 인터넷의 기본 아키텍처를 다시 설계하는 연구가 필요함에 동의하고 있다.

새로운 커뮤니케이션과 네트워킹 패러다임이 실제 적용되어 문제 없이 사용되기 위해서는 최소한 그 핵심 개념만이라도 큰 규모의 인터넷에서 테스트를 통해 검증되어야 한다. 테스트의 목적은 단지 기술적인 측면뿐이 아닌 사회, 경제 등에 미치는 영향들을 평가하고 새로운 요구사항들을 추출함으로써 새로운 아이디어의 적용 가능성과 현실에서의 실현 가능성을 평가하는 것이다. 이러한 실험적인 연구 방법은 새로운 인터넷 아키텍처에 대한 연구가 탁상공론에만 그치지 않고 실제 인터넷의 개선을 주도 할 것이다.

현재 인터넷에 대한 새로운 기술적 해결책들은 점진적인 개선 방식일 수도 있고 완전히 새로운 clean-slate[6] 방식일 수도 있다. 전자는 현재까지 우리가 인터넷의 문제를 해결하기 위해 사용해왔던 방식이고 후자는 새로운 개념과 개선된 성능을 제공하기 위한 급진적인 방식으로 인터넷을 아키텍처부터 새로 설계하는 방식이다. Clean-slate 방식은 현재 인터넷의 한계에서 벗어나 완전히 새로운 아이디어를 바탕으로 네트워크 아키텍처를 설계하고 새로운 아이디어를 바탕으로 test facility를 구축과 구축한 test facility에서의 실험을 통한 아이디어의 검증까지 포함한다. Clean-slate 방식에 의해 아키텍처부터 통신 프로토콜까지 새롭게 설계될 인터넷을 ‘미래 인터넷(Future Internet)’이라 칭한다.

본 연구는 두뇌한국 21 과 지식경제부 및 정보통신연구진흥원의 대학 IT 연구센터 지원사업의 연구결과로 수행되었음" (IITA-2008-C1090-0801-0045)

최근 30년 동안 인터넷은 점진적인 개선 방법을 통해 성공적으로 유지될 수 있었다. 하지만 현 인터넷의 단점을 극복하면서 보안과 성능이 보장되는 새로운 서비스를 보장받기 위한 미래 인터넷을 설계하려는 clean-slate 방식의 적용이 필요한 때이다. 새로운 아키텍처를 설계하기 위해서는 미래 인터넷의 요구사항을 분석하고 디자인 초기단계에서부터 관리성에 대한 고려가 이루어져야 한다. 우리는 본 논문은 미래 인터넷의 아키텍처 설계와 관리 연구에 대한 전세계적인 연구 현황 조사를 기반으로 미래 인터넷 관리 요구사항을 추출하고 미래 인터넷의 관리를 위한 기능들을 제안한다.

본 논문의 구성은 다음과 같다. 2장에서는 미래 인터넷이 해결해야 할 과제와 요구사항들을 제시한다. 3장에서는 미국, 유럽, 일본에서 진행되고 있는 미래 인터넷 관련 연구 현황에 대해 소개한다. 4장에서는 현재 연구되고 있는 미래 인터넷 아키텍처들의 개념을 설명하고 5장과 6장에서는 미래 인터넷의 관리를 위한 요구사항과 그에 따른 관리기능들을 제안한다.

2 미래 인터넷의 요구사항

미래 인터넷의 관리성을 고려하여 설계하기 위해서는 먼저 미래 인터넷의 요구사항에 대해서 정리할 필요가 있다. 이 장에서는 미래 인터넷이 해결해야 할 과제와 요구사항에 대해서 정리한다.

2.1 가용성 (availability)

오늘날 가장 신뢰할 만한 네트워크는 전체 가동 시간의 약 99.99%의 시간 동안 이용 가능하다고 한다[16]. 미래 인터넷은 약 99.9999%의 이용 가능성 제공을 목표로 한다. 이는 한 달 동안 이용 불가능한 시간이 1초 이하임을 의미하는 것으로 높은 가용성은 다양한 결함들로부터 신속한 복구를 뜻한다. 즉, 미래 인터넷은 끊임 없는 서비스 제공을 목표로 하며, 결함 발생시 복구가 빨라야 할 뿐 아니라 fault-tolerant한 이중 구조를 지원해야 한다. 미 국방부의 현재 네트워크 중심의 전쟁 정책에 주어진 네트워크의 주도적인 역할을 고려할 때 가용성은 가장 우선해서 고려되어야 할 요구사항이다.

2.2 안전한 정보의 전달

두 번째로 중요한 항목은 정보의 보안으로, 이것은 인증, 권한 부여, 과금, 기밀성 등을 포함한다. 또한 사용자에게 시간과 장소에 따라 어떤 보안 특성들이 이용 가능한지를 알려주어야 한다. 우리는 인터넷 뱅킹이나 온라인 결제와 같은 서비스의 이용시 보안이 충분히 제공되지 못할 때 이것이 통신의 두절 자체보다 더 큰 문제를 야기하게 된다. 왜냐하면, 이것은 사용자나 어플리케이션에 따라 우선적으로 고려하는 사항이

다르므로 보안 정도에 따라 서비스의 이용 여부가 결정될 수 있기 때문이다.

2.3 이동성에 대한 지원

유비쿼터스 컴퓨팅 환경의 등장을 시작으로 미래 인터넷의 이동성(mobility)이 지원되어야 하는 어플리케이션과 그 이용자는 기하급수적으로 늘어나고 있다. 현재 이동성에 대한 문제는 IP 이동성에 관한 솔루션들이 필요 이상으로 복잡하다는 점과 그 성능이 유선 네트워크보다 떨어진다는 점이다. 미래 인터넷은 더 단순한 이동성에 대한 해결책과 일정 수준 이상의 성능을 보장해야 한다.

2.4 유연성과 확장성

미래 인터넷에 대한 하나의 완벽한 정답은 미래 인터넷 이후의 새로운 인터넷이 필요하지 않을 정도로 완벽해야 한다는 것이다. 즉, 미래 인터넷이 인류의 마지막 인터넷 아키텍처가 되어야 한다는 의미이다. 그러나 이것은 현실적으로 불가능하며 미래 인터넷은 반드시 다양한 아키텍처가 공존할 수 있도록 구상되어야만 한다. 왜냐하면 미래 인터넷 아키텍처는 단일하게 홀로 존재하기 힘들기 때문이다. 즉, 아키텍처가 유연하고 확장 가능하도록 구상되어야 한다는 것이다. 이 설계 원칙은 실험과 혁신을 지원하기 위한 test facility의 설계와 미래 인터넷 아키텍처 설계에 모두 해당되는 사항이다. 이 원칙은 미국의 FIND(Future Internet Design) 프로그램의 흐름 안에서 “변화를 위한 설계”라고 불리고 있다.

2.5 통합된 네트워크에 대한 확장성의 지원

현재 인터넷상에서 사용되는 상용 어플리케이션들과 E-science 어플리케이션과 같은 매우 큰 대역폭을 요구하거나 극도로 높은 이동성 요구사항을 가진 어플리케이션들의 이용이 증가하고 있다. 미래의 네트워크는 어플리케이션들의 이런 다양한 요구사항이 하나의 통합된 네트워크 안에서 충족될 수 있는 아키텍처를 제공해야 한다. 그렇지 않고 네트워크마다 지원하는 기능들에 차이가 있다면 이동 단말이 방대한 규모 네트워크의 데이터 센터에 접속할 수 없는 상황이 발생할 수도 있다.

2.6 크로스 레이어 상호작용에 대한 지원

기존의 프로토콜 스택들은 인접 레이어간의 인터페이스만을 제공하고 있다. 그래서 크로스 레이어(레이어간의 상호작용)의 제어를 위한 인터페이스가 부족하며 근래의 프로토콜들은 보안상 취약한 ad-hoc 방식으로 크로스 레이어를 지원한다. 미래의 인터넷 아키텍처는 반드시 내부에 지정된 통신이 요구될 때, 네트워크의 동작을 최적화하기 위한 명시적인 기능 단위(Functional

block)를 갖추고 있어야 한다. 따라서 아키텍처는 모든 네트워크의 기능 단위들이 전통적인 데이터 인터페이스들뿐만 아니라 잘 정의된 제어와 튜닝 인터페이스를 갖추기를 요구한다.

2.7 진화하는 보안 정책의 원활한 통합

네트워크 보안 특징들은 암호화 또는 다른 소프트웨어 위주의 기술에 대한 몰이해, 설정 또는 성능상의 문제를 이유로 종종 무시당한다. 이것은 근본적으로 보안이 아키텍처와 함께 통합되어 설계되지 못하고 부가적인 것으로 인식되어 온 결과이다. 초기 설계 단계부터 원칙적으로 네트워크의 보안 의미(semantic)에 대한 이해와 다양한 통신 서비스와의 통합을 요구하고 있다.

2.8 데이터와 제어의 분산

데이터와 제어가 반드시 아키텍처 전체에 걸쳐서 분리되어 있어야 한다. 제어와 관련된 통신에만 이용될 수 있는 분리된 채널이 존재해야 한다. 많은 양의 데이터 트래픽이 제어 정보의 전송을 방해해서는 안 된다는 사실은 그 동안의 인터넷 네트워크의 운용으로부터 얻은 교훈이다. 제어 기능을 위해 종단뿐만 아니라 중간의 라우터나 스위치도 데이터와 제어를 분리하여 처리하는 프로세서와 버퍼의 할당이 이루어져야 한다.

3 미래 인터넷 관련 연구 현황

미래 인터넷의 연구는 미국 [6], EU [7], 일본 [9] 그리고 한국[15] 등 전 세계적으로 진행되고 있다. 연구 활동은 크게 두 가지로 나누어 이루어지는데, 하나는 미래 인터넷 아키텍처의 설계이고 다른 하나는 제안된 아키텍처를 검증하기 위한 실험 가능한 테스트베드의 조성이다.

미국에서는 NSF (National Science Foundation)의 주도하에 미래 인터넷에 대한 연구가 활발히 진행되고 있다. 네트워크 아키텍처, 프로토콜, 알고리즘의 개발을 포함하는 FIND (Future Internet Design) 프로그램과 미래 인터넷에 대한 연구를 검증하기 위한 test facility를 구축하는 것을 목표로 하는 GENI (Global Environment for Networking Innovations) 프로그램이 있다. FIND는 2006년부터 시작된 장기간의 프로그램으로 인터넷의 설계에 관한 연구를 진행하며, 2014년부터 GENI의 test facility를 이용해 연구 내용을 검증할 계획이다. GENI는 2005년에 시작된 프로그램으로 GENI research 프로그램과 GENI research facility로 구분된다. GENI research 프로그램은 네트워킹에 관한 기초적인 연구나 실험을 중심으로 하며, GENI research facility는 실제와 가까운 조건 하에서 네트워크 아키텍처의 새로운 개념과 검증을 장려한다. 2005년부터 현재까지는 연구단계이고 2008년 후반부터는 실제 테스트 설비의 제작에

돌입할 예정이다.

EU에서의 미래 인터넷 연구는 FP6 (the sixth Framework Program)로부터 시작되었다. 2007년부터는 FP7 (the seventh Framework Program)이 FP6에 이어 미래 인터넷 연구의 중심적 역할을 하고 있다. 현재의 FP7은 2007년부터 2013년까지 7년 예정의 연구 프로그램이며 ICT (Information and Communication Technologies)는 FP7의 주요 연구 테마이다.

FIRE (Future Internet Research and Experimentation)는 네트워킹 테스트베드에 관한 EU의 연구를 통합하기 위한 프로그램이다[4]. FIRE는 2007년 2번째 ICT call에서 FP7-ICT의 ‘새로운 패러다임과 test facility’를 목표로 시작했고, 프로젝트들은 2008년 초부터 시작 예정이며 현재는 준비 중이다.

일본의 미래 인터넷에 대한 연구는 NXGN (Next Generation Network)과 NWGN (New Generation Network)의 2가지 단계로 나누어져 있다. NXGN에서는 기본 IP 구조가 위에 쿼드러플 플레이 서비스가 제공될 것이다. NXGN은 현재 배치 단계에 돌입했으며, NXGN을 이용한 상용 서비스는 2010년경부터 시작될 것으로 보인다. NWGN에서의 네트워크 아키텍처와 서비스는 IP 네트워크와는 다른 새로운 패러다임을 제시한다. 현재, NWGN은 연구 단계이고, 새로운 아키텍처에 대한 연구가 optical 네트워킹 기술과 유비쿼터스 네트워킹과 같은 기술을 바탕으로 진행되고 있다. NWGN의 프로토타입은 2015년에서 2020년 사이에 완성될 것으로 기대된다.

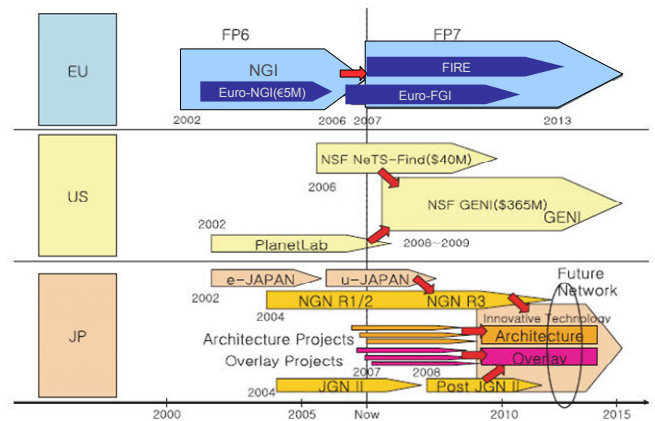


그림 1. EU, 미국, 일본의 미래 인터넷 연구 계획

그림 1은 EU, 미국 그리고 일본의 미래 인터넷 연구의 전체적인 개요를 나타낸다. EU는 FP6을 시작으로 첫 번째로 미래 인터넷 연구를 시작했고 FP7을 통해 2013년까지 계속될 것이다. EU는 네트워크 아키텍처 설계과 테스트 설비를 제작하는 FIRE 프로그램을 진행하고 있다. 미국은 2002년부터 PlanetLab이라는 테스트 설비의 제작을

시작했으며 2006년부터는 FIND 프로그램을 통해 새로운 인터넷 아키텍처에 대해 연구하고 있다. 또 2008년부터는 PlanetLab을 개선한 GENI facility의 구축을 시작할 예정이다. 일본은 2002년과 2006년에 각각 e-JAPAN과 u-JAPAN 프로젝트를 시작했고 인터넷 아키텍처 디자인인 NGN을 수행하고 있다. 또한 2004년부터는 JGN2를 시작했고 2012년까지 Post-JGN2라는 이름으로 연구를 계속 진행할 예정이다.

4 미래 인터넷 아키텍처

이 장에서는 FIND, GENI, FIRE 등의 연구 기관에서 수행중인 인터넷 아키텍처 관련 연구에 대해서 알아본다.

4.1 SILO 아키텍처

SILO (Service Integration, control, and Optimization)[14]는 NC state university와 RTI에서 공동으로 진행하고 있는 FIND 프로젝트 중 하나이다. 이 프로젝트의 목표는 복잡한 통신들이 기존의 계층적 구조의 아키텍처를 기반으로 이루어진 것이 아니라 통신시에 필요한 기능들을 동적으로 조합하여 프레임워크를 완성하고 프로토타입의 구현을 통해 검증하는 것이다. 그림 2는 SILO 아키텍처의 구성 요소들을 나타내고 있다.

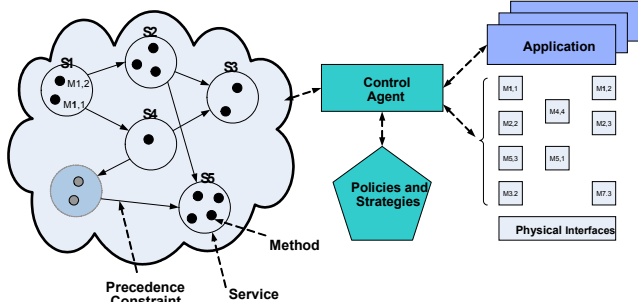


그림 2. SILO 아키텍처의 구성 요소

그림 2의 원은 서비스를 나타내고, 원 안의 점들은 특정한 하나의 서비스를 구현한 다양한 메소드를 나타낸다. 화살표는 서비스들 간의 우선순위 제약을 나타낸다. CA (Control Agent)는 구성 요소들과 상호작용하고 우선순위 제약 하에서 메소드들을 이용해 silo(각 connection마다 생성되는 SILO 인스턴스이며 소문자로 표기)를 구성한다. 각 구성 요소에 대한 설명은 다음과 같다.

- 서비스 (Service): SILO 아키텍처의 기본 요소는 서비스이다. 서비스는 구체적 통신의 수행에 연관된 정의가 명확한 독립적인 기능이다. 그 예로 종단간 플로우 제어, 패킷 단편화, 압축 등이 있다. 각각의 서비스는 분리되고 원자화된 기능 단위에 접근하므로 아키텍처는 대체로 복잡한 기능들을 포함시킨 현재의 프로토콜보다 더 탄력적인 적용이 가능하고 더 세분화 되어야 한다. SILO 아키텍처는 동적으로 특정한 작업을

수행하는데 필요한 서비스들의 모음을 구성할 수 있어야 하며 동시에 각각의 서비스들 사이에는 우선순위가 있어야 한다. 서비스들 간의 우선순위는 현재의 인터넷 같은 계층적 구조가 아닌 명확하게 정의되어야 한다. 예를 들면 ‘압축’과 ‘암호화’라는 두 서비스가 있을 때, ‘압축’은 ‘암호화’ 이전에 수행되어야만 한다. 암호화 이후의 압축은 의미가 없기 때문이다.

- Knob: Knob은 서비스 내의 제어 파라미터를 의미한다. 서비스는 기능, 다른 서비스들에 대한 인터페이스, 다른 서비스들과의 관계에 미치는 특징, 그리고 Knob, 액션과 제약을 완전하게 정의한 것이다. Knob은 서비스에 의해 수행되는 기능에 맞추어 조정 가능한 특정한 범위의 값을 갖는 파라미터이고, 특정한 범위의 값들과 서비스의 성능간에는 명확하게 정의된 관련성이 존재한다. 예를 들면, ‘압축 계수’는 ‘압축’ 서비스를 위한 Knob이다. Knob은 특정한 태스크를 위해 선택된 서비스 집합의 성능을 최적화 하기 위해 CA에 의해 계산된다.
- Method: 우리는 서비스의 정의와 구현을 구분한다. Method는 서비스의 기능을 수행하기 위해 특정한 메커니즘을 사용하는 서비스의 구현이다. 예를 들면, “window-based flow control”은 “end-to-end flow control” 서비스의 한 Method가 될 수 있다. 서비스를 구현한 Method는 서비스 지향적인 Knob 뿐만 아니라 서비스 지향의 인터페이스도 구현해야 한다. 즉 서비스 지향의 인터페이스와 Knob은 서비스를 구현한 Method에 다양한 형태로 존재 할 수 있다. Method는 자신이 구현하는 서비스와 서비스를 구현하기 위해 사용하는 알고리즘 또는 메커니즘, Method 지향적인 제어 파라미터, 액션과 제약들을 기술함으로써 완전하게 정의될 수 있다. SILO 아키텍처는 서비스와 인터페이스를 정의하지만 서비스를 구현하는 Method를 정의하지는 않는다. 이로써 주어진 서비스에 대해 서로 다른 방식으로 구현된 복수의 Method가 네트워크 내에 존재 할 수 있다. 이 때 순서가 정해진 Method들의 부분 집합을 silo라 칭한다. Silo는 어플리케이션에서 네트워크로 나가는 데이터의 변환들을 수행한다. 각각의 데이터 변환은 silo내의 Method에 해당하며 단순한 데이터의 변환 만이 아니라 패킷 안에 포함되어야 할 메타데이터를 생성할 수도 있다.
- CA(Control Agent): CA는 각 노드내에 존재하며 어플리케이션 스트림을 위한 silo를 생성하고 서비스와 Method에 적합하게 Knob을 조정하며, 서비스 간의 상호작용을 용이하게 하는 역할을 수행한다. Silo를 생성한다는 것은 silo가 가지는 서비스들을 선정하고 그 서비스들의 우선순위에 따라 스택에 정렬하며 각 서비스에 해당하는

Method들을 결정하는 것이다. CA는 새로운 종단간의 연결마다 동적으로 그 연결에 적합한 silo를 생성한다. 따라서, CA는 어플리케이션의 QoS 요구사항 및 현재 네트워크 자원의 가용성과 서비스간의 우선순위 제약, 그 시점에 영향을 미치는 정책 등을 고려한다.

4.2 GENI 관리 프레임워크

GENI는 미래 인터넷 실험 환경을 조성하기 위한 연구를 진행하고 있으며 자원의 가상화를 통해 복수의 혁신적인 실험의 동시 수행을 목표로 한다. 이번 장에서는 GENI 관리 프레임워크의 기본적인 구성 요소들의 개념과 관리 핵심(core)에 대해 간략하게 설명한다.

4.2.1 컴포넌트 (Component)

GMC(GENI Management Core)는 컴포넌트를 GENI의 첫번째 기본 요소로 정의한다. 컴포넌트는 물리적 자원 (CPU, 메모리, 디스크, 링크 대역폭)과 논리적 자원(파일 디스크립터, 포트 번호)의 모음으로 구성된다.

4.2.2 슬리버 (Sliver)

슬리버는 컴포넌트의 자원들로 구성된 부분 집합이다. 각각의 컴포넌트는 각 슬리버들을 서로 독립적으로 유지할 수 있는 하드웨어 혹은 소프트웨어 메커니즘을 포함하고 있어야 한다. 슬리버의 예로는 가상 서버, 가상 라우터, 가상 스위치, 가상 AP(access point) 등이 있다.

4.2.3 슬라이스 (Slice)

슬라이스는 GENI 컴포넌트에 존재하는 슬리버들의 집합과 실험을 실행하기 위한 목적으로 슬리버에 접속할 수 있는 이용자를 일컫는 용어이다. 이용자는 슬라이스 내에서 그들의 실험을 수행하고 서비스를 제공할 수 있으며 새로운 네트워크 아키텍처를 테스트할 수 있다. 슬라이스는 내부에 연관된 사용자들의 집합을 가지고 있다.

4.2.4 CM (Component Manager)

GENI의 각각의 구성 요소들은 각각의 컴포넌트 안에 속한 슬라이스들을 할당하고 제어하는 역할을 하는 CM을 실행시킨다. CM은 컴포넌트가 지원하는 가상화된 객체들이 무엇이건 효과적으로 단일한 제어가 가능한 인터페이스를 제공한다. 이 특징은 새로운 컴포넌트가 쉽게 GENI에 추가될 수 있도록 한다. 그림 3은 GENI 노드에 실행되고 있는 CM을 나타낸다. CM은 제어 인터페이스와 센서 인터페이스를 외부에 제공하고, 두 인터페이스는 모두 그 컴포넌트의 슬라이스 위에서 실행되고 있는 서비스들에 의해 접근이 가능하다. 예를 들면, PC 기반의 노드에서는 CM이 가상 머신을 생성하기 위해 OS에서 제공하는 기능을 호출해 슬라이스를 초기화 할 수 있다.

CM은 때때로 오류가 발생한 슬라이스를 제거하거나 슬라이스에 할당된 자원들을 변경할 수 있다.

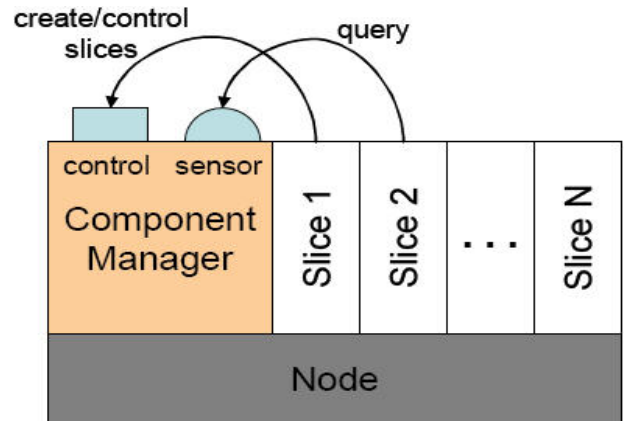


그림 3. GENI 노드의 Component Manager

CM의 제어 인터페이스의 핵심 파라미터는 슬라이스에 할당된 자원들을 식별하는 확장 가능한 슬라이스 명세이다. 예를 들면, 이러한 명세는 특정한 슬라이스에 100M cycles-per-second의 CPU, 128MB의 메모리, 5GB의 디스크 저장소, 그리고 45Mbps의 링크 대역폭이 할당될 것임을 가리킬 수 있다. 또한, 물리적 장비 외에도 포트 번호, 가상 회로 식별자 그리고 네트워크 주소와 같은 논리 자원 또한 할당할 수 있다.

슬라이스를 생성하고 제어하는 방법 외에도 CM은 센서 인터페이스를 통해 CPU와 메모리의 점유율과 같은 컴포넌트의 상태 정보를 보고한다. 그림 3에서는 노드 위의 슬라이스가 센서 인터페이스에게 컴포넌트의 상태에 대해 질의하고 있다.

컴포넌트가 다수의 슬라이스를 할당하는 방법은 가상화의 레벨이 컴포넌트 상에 프로그램 될 수 있는 실험의 종류에 직접적인 영향을 미친다. GENI는 어떤 레벨에서도 여러 종류들이 실험 가능하도록 연구자들로 하여금 모든 레벨의 물리적 장치에 접근을 허락한다. GENI는 가상화 레벨을 다음과 같은 세 가지로 구분한다.

- 오버레이 레벨: 슬라이스가 현재 인터넷의 최상단에 위치한 오버레이 네트워크처럼 어플리케이션과 서비스가 실행하도록 한다. 슬라이스는 가상 머신에 TCP, UDP 또는 IP와 같이 구현된 가상의 모든 링크에 접근할 수 있다.
- 가상 장치 레벨: 슬라이스로 하여금 레이어 2상의 모든 아키텍처나 서비스를 포함 할 수 있도록 한다. 이 가상의 링크들은 검증된 성능 특성을 가지고 물리적인 카드를 충실히 재연해 놓은 것과 같은 가상 장치와 같이 접근된다. 즉, 장치의 큐나 링크 결합 등을 노출한다.
- 회로 제어 레벨: 슬라이스가 회로를 스위칭

디바이스를 통해 제어(설정, 구성 등)하는 네트워크 아키텍처를 실행 할 수 있게 한다. 슬라이스는 가상 제어 인터페이스를 이용해 제어한다. 주의할 것은 모든 GENI 컴포넌트에 모든 레벨의 가상화(프로그램 가능화)가 가능하지는 않다는 점이다. 예를 들면 어떤 컴포넌트는 오버레이 레벨의 슬라이스만을 지원 할 수 있는 반면, 다른 컴포넌트는 첫째와 둘째 레벨의 지원은 허용하나 셋째 레벨의 지원은 오직 백본으로 동작할 때만 허용하는 경우가 있을 수 있다.

4.2.5 GMC (GENI Management Core)

컴포넌트들의 집합은 GMC를 통해 하나의 자율적인 단위로 결합될 수 있다. GMC는 두 가지 기본적인 기능을 한다. 첫째는 물리적인 GENI 컴포넌트들 위에 슬라이스를 초기화 하는 것이고 둘째는 슬라이스화된 컴포넌트들을 원격으로 관리 하는 것이다.

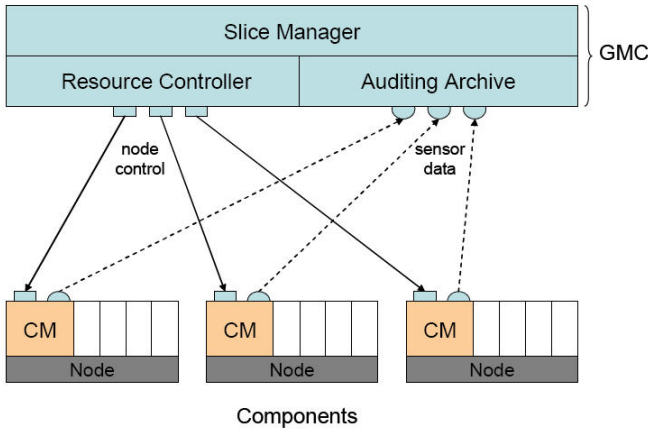


그림 4. GMC와 GENI 노드

- 슬라이스 매니저 (Slice Manager): 각 슬라이스들의 상태를 기록한다. 슬라이스를 생성하고 제어하는 기능을 한다.
- 리소스 컨트롤러 (Resource Controller): 각각을 구성하는 노드, 링크, 서버셋, 각각의 자원들의 상태, 자원들의 성능 (링크 대역폭, CPU 성능, 메모리 크기, 프로세서 타입), 네트워크 상에 자원이 위치하는 장소, 자원에서 실행되고 있는 소프트웨어 그리고 GMC가 원격 자원들을 제어하기 위해 사용하는 메소드 (제어 프로토콜) 등을 포함하는 모든 정보를 기록한다.
- 오디팅 아카이브 (Auditing Archive): 주기적으로 각각의 노드를 감시하며 데이터를 업로드 하고 저장한다. 이 정보들은 보안에 관련된 사고를 감지하고 해결하기 위해 사용된다.

그림 4는 위 세 가지 모듈을 이용해 구성되는 GMC의 구조를 나타낸다. 복수의 GMC가 동시에 초기화 될 수 있다. 각각의 GMC는 GENI에서 이용

가능한 컴포넌트들의 서버셋을 관리한다. 복수의 GMC에 대한 아이디어는 분산화에 대한 지원에서부터 비롯되었다. 한 개의 GMC만 존재할 수도 있지만, 드문 경우이다. 대개는 여러 개의 GMC가 존재하며 우리는 서로 독립적인 GENI 자원들을 제어하는 각각의 개체들을 management authority라고 부른다.

4.3 Japan NWGN 아키텍처

NWGN은 현재 새로운 아키텍처의 연구 단계에 있다. NWGN은 네트워크 아키텍처와 서비스를 기존의 IP 네트워크와 다른 새로운 패러다임으로 제시하고 있다. 새로운 아키텍처에 대한 연구가 optical 네트워킹 기술과 유비쿼터스 네트워킹과 같은 기술을 바탕으로 진행되고 있다.

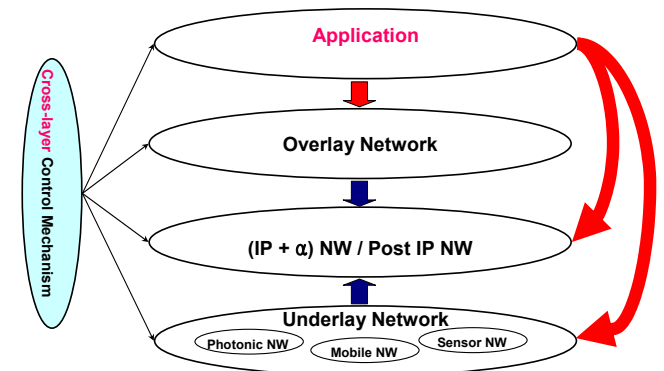


그림 5. NWGN 아키텍처

그림 5는 NWGN의 전체적인 아키텍처를 보여주고 있다 [9]. 이 아키텍처는 NICT (National Institute of Information & Communications Technology) [10]의 AKARI 프로젝트에 의해 제시되었다. NXGN은 현재 배치 (deployment) 및 표준화를 준비 중이며, NWGN은 연구 단계에 있다. 그림에서 보듯이 NWGN은 계층적 구조로 구성되어 있다. 최하위 계층은 언더레이 (underlay) 네트워크이고 그 위 계층은 IP 계층, 그 위로 오버레이 네트워크와 어플리케이션 계층이 존재한다. 현재 언더레이 네트워크에서부터 어플리케이션 계층에 걸쳐 다양한 연구가 진행되고 있다. 언더레이 네트워크에는 광 네트워크, photonic 네트워크, cognitive 무선 네트워크 등의 세 가지 가능한 네트워크가 고려되고 있다. IP 계층에는 현재 IP에 더 많은 기능들을 추가할 수 있으며 차세대 IP 네트워크로 발전할 수 있는 가능성이 있다. 보안과 상황 인지 (context-aware) 서비스를 제공하는 오버레이 네트워크를 위한 연구가 NICT와 대학들 간의 협력 과제로 진행되고 있다. NWGN에 가능한 어플리케이션들은 4K와 8K의 초고화질 어플리케이션과 그리드 컴퓨팅 그리고 e-Science를 위한 어플리케이션 등이 있다. NWGN은 투명성, 개방성, 단순성, 견고성, 신뢰성, ubiquity 그리고

보안성을 필요로 한다.

5 미래 인터넷 관리를 위한 요구사항

미래 인터넷의 관리를 위한 다른 연구들에 기초해서 [8], 우리는 네트워크 자원과 서비스들의 동작과 관리를 위한 요구사항들을 제시한다. 미래 인터넷 관리 요구사항을 관리 모델에 의해 정보 모델 (information model), 통신 모델 (communication model), 그리고 기능적 모델 (functional model), 비기능적 모델 (non-functional model)로 나누어 정리하였다. 미래 인터넷의 정보 모델관점에서의 관리 요구사항들은 다음과 같다

- 네트워크 특성들과 정책에 대한 하이 레벨, 목표 지향적인 명세: 네트워크 관리자는 원하는 자원의 구성 특징을 표현할 수 있어야 하고 네트워크 관리 소프트웨어는 하이 레벨 특성들을 하위 레벨의 구현으로 변환할 수 있어야 한다.
- 다양한 관리 대상들의 정의: 관리 정보 모델은 하드웨어 자원에서부터 사업 목표 그리고 관리 기능까지 상세하게 명시하여야 한다.
- 확장성: 관리 정보 모델은 어떤 종류의 새로운 네트워크 자원이나 서비스에도 적용 가능해야 한다.

미래 인터넷의 통신 모델관점에서의 관리 요구사항은 다음과 같다

- 관리 operation 지원: 통신 모델은 get, set, create, add, delete, action 그리고 notify 등의 오퍼레이션들을 지원해야 한다
- 셀프 부트스트래핑 management plane [13]: management plane은 반드시 data plane과는 독립적으로 운영되어야 하며 미리 지정하지 않아도 부트스트랩이 가능해야 한다.
- 모든 data plane 프로토콜들을 위한 하나의 단순한 관리 인터페이스: 프로토콜 오퍼레이션상의 복잡도는 구현에 국한되어야 하고 간단한 관리 인터페이스를 통해 관리 정보의 표현이 가능해야 한다. 이것은 관리 어플리케이션의 부담을 줄이는 대신 프로토콜을 구현하는 측에 프로토콜 오퍼레이션에 대해 상세히 이해할 것을 요구한다.

미래 인터넷의 기능 관리 요구사항은 다음과 같다

- FCAPS: 기능 모델은 반드시 Fault (결함), Configuration (구성), Accounting (과금), Performance (성능) 그리고 Security (보안)과 같은 기본적인 관리 기능을 지원해야 한다.
- 서비스 지향 아키텍처 (SOA) 지원: 미래 인터넷의 아키텍처는 SOA (Service-Oriented Architecture) 기반이 될 것이다. 관리 기능들도 서비스로 정의하여 제공해야 한다. 서비스는

서비스 저장소에 등록되고 SOA에 기반해서 검색되고 제공되어야 한다.

- 프로그래밍 가능한 지능형 (intelligent) 네트워크 노드: 지능적이고 자율적인 관리를 위해서는 터미널, 중개, 코어 등의 네트워크 노드들이 프로그래밍 가능해야 한다.
- 식별과 어드레싱: 현재의 어플리케이션/서비스 레벨의 어드레싱과 네트워크 어드레싱 간의 강한 연결은 혁신적인 이동성 개념과 동적인 서비스 구성의 일반화를 포함한 새로운 서비스를 개발하는데 장애가 되고 있다. 이용자와 대상 (터미널, 센서, 액츄에이터, 모든 형태의 전자 장비), 어플리케이션 식별자, 플로우 식별자, 그리고 네트워크 주소들은 전세계적으로 통합된 아키텍처 측면의 접근 방법으로 서로 독립적으로 정의될 필요가 있다.
- 데이터 저장소 관리: 관리성(manageability)를 보장하기 위한 미래 인터넷 아키텍처는 네트워크의 현재 상태에 대한 실시간 데이터를 생성하여 제공해야 한다. 관리 정책의 결정 또한 네트워크의 동작이나 진화에 대한 장기적인 관점에 의존한다. 문제는 제안된 수단들에 의해 잠재적으로 방대한 양의 데이터가 수집되리라 예측된다는 점이다. 이것은 직접적으로 측정된 데이터를 효과적으로 조직하고 시간의 범위에서 관리에 대한 정책의 결정을 가능하게 하는 대용량의 데이터 저장소의 필요성을 암시한다.
- 보장된 QoS: 오늘날 QoS를 보장하기 위해 ISP들에 의해 주로 이용되는 방법은 자원의 과잉 공급 (over provisioning)이다. 즉, ISP들은 대개 복잡한 QoS 알고리즘을 구현하는 대신에 그들의 네트워크가 과부하가 걸리지 않도록 그들의 백본 링크에 충분히 큰 대역폭을 제공함으로써 매우 적은 지연시간을 보장하고 전송시에 패킷의 분실을 방지하는 것이다. 미래 인터넷은 현재의 네트워크보다 더 넓은 대역폭과 더 효율적인 채널의 제공이 예상되지만 무선 네트워크의 대역폭 지원 비용은 유선 네트워크보다 더 클 것이다. 그러므로 미래 인터넷에는 과잉 공급 정책이 부적합하며, 반드시 QoS 지원 메커니즘이 필요하게 될 것이다
- 일반화된 이동성에 대한 지원: 현재 인터넷상에서는 터미널 장비가 서비스를 계속 유지하기 위해서 같은 타입의 네트워크 안의 셀을 바꾸는 수평적 핸드오프 (horizontal hand-off)만이 제공되고 있다. 미래에는, 이용자가 유선 액세스 포인트로부터 무선 액세스 포인트로 이동을 가능하게 하는 기술과 같이 더 많은 액세스 기술을 이용할 수 있는 더 넓은 의미의 이동성이 요구된다. 즉, 미래 인터넷 환경에서는 수평적 핸드 오프뿐만 아니라

수직적 핸드오프 (vertical hand-off)도 반드시 지원되어야 한다. 수직적 핸드오프 메커니즘은 터미널 장비가 종단 사용자 어플리케이션에게는 완전히 투명한 방식으로 서로 다른 타입의 네트워크로 이동할 수 있도록 한다. 가장 힘든 문제는 수직적 핸드오프가 특정한 QoS 요구사항들이 만족되는 상태에서 이루어져야 한다는 점이다.

미래 인터넷의 비기능적 관리 요구사항은 다음과 같다.

- 견고성: 네트워크는 하나 이상의 NM을 가질 수 있다. 또한, 위에서 언급했듯이 관리 채널의 기본적인 개념은 서로 분리된 네트워크에 속해있는 NM들 간의 통신의 허용과 같은 다양한 확장성을 요구한다.
- 확장성: 오늘날 가장 큰 네트워크는 수 천개의 라우터와 스위치, 그리고 수만개의 장비를 가지고 있으며 현재 인터넷의 default-free zone은 route를 십만여 개의 destination prefix를 이용해 다룬다. 기존의 서버들이 엄청나게 많은 장비들을 관리하고 네트워크의 목적을 달성하기 위해 이벤트에 즉각적인 응답이 가능한지, 방대한 양의 관리 정보의 전송이 데이터를 전송할 수 있는 한계를 넘어서진 않는가? 등의 질문에 답을 할 수 있어야 한다. 확장성은 미래 인터넷의 요구사항 중 하나이다. 백만개의 서로 다른 네트워크 장비들을 지원하고 서비스를 제공하기 위해 미래 인터넷의 관리 시스템과 오퍼레이션도 확장성을 가져야 한다.
- 상호 운영: 우리는 서로 다른 NM이 각각 보안, 성능 등 서로 다른 역할을 수행하고 있고 서로 다른 도메인 안에 있는 NM들 간의 상호작용 그리고 심지어는 주어진 도메인 내의 다른 벤더들의 NM들이 같은 역할을 수행하기 위해 경쟁할 것으로 가정한다.
- 효율성: 네트워킹 기술과 능력의 확장과 더불어 능률적이고 경제적인 네트워크 오퍼레이션이 요구된다. 미래 인터넷의 네트워크 관리팀은 현재의 네트워크를 관리하는 인력보다 더 적은 인력을 이용한 기반 시설의 조작과 유지가 가능해야 한다.
- 자율성 측면: 최근 자동화된 관리의 진화된 형태인 자율 관리 개념이 등장했다. 관리 기능상의 용어로 시스템은 관리되고 있지 않은 (unmanaged) 상태, 관리 중인 (managed) 상태, 예측 가능한 (predictive) 상태, 적응 가능한 (adaptive) 상태, 그리고 자율적인 (autonomic) 상태 등으로 기술될 수 있다. 다양한 환경에 적응 가능한 closed-loop 자율 관리는 궁극적인 목표인 완전히 분산된 적응적인 자율 관리의 자율적 동작의 첫 형태이다. 다양하고 복잡한 미래의 네트워크 자원과 서비스들은 자동,

자율적으로 관리되어야 한다.

6 미래 인터넷의 관리기능

미래 인터넷 관리는 미래 인터넷 자원과 서비스를 위한 관리 기능들을 제공하고 관리 측면과 네트워크 자원 또는 서비스와 그리고 다른 관리 영역들 간의 통신을 제공한다. FCAPS 관리 영역 또한 미래 인터넷 관리를 위해 지원되어야만 한다. FCAPS 관리는 미래 인터넷 유선 네트워크의 오퍼레이션과 관련된 대부분의 문제들을 충분히 해결할 수 있을 것이다. 무선 네트워크와 크로스 레이어드 네트워크의 등장으로 FCAPS에 의해 해결될 수 없는 몇 가지 항목들이 추가되어야 한다. 추가되어야 하는 항목에는 이동성 관리, 버추얼 리소스 관리, 크로스 레이어 관리, 터미널 관리 등이 있다.

6.1 결함 관리 (Fault Management)

각종 공격으로부터의 보안과 결함으로부터의 신속한 복구를 위해서는 미래 인터넷의 기반 시설에 대한 의존도가 현재보다 더 많이 증가할 것임을 추론할 수 있다. 네트워크의 관점에서 보면 이것은 네트워크에 대한 공격으로 인한 오류의 발생이 불과 발생한 후 몇 초 안에 해결되어야 함을 의미한다. 더 넓은 지역과 많은 이용자를 위해 네트워크 장비의 수가 증가함과 동시에 장비들의 종류도 훨씬 다양해지므로 확장 가능한 결함 관리 방법이 요구된다. 네트워크 장비들은 핸드오프와 로밍, 충전 등과 수반되어 더 많은 이벤트들을 생성할 것으로 보이고, 그에 따라 미래에는 기존의 네트워크보다 처리해야 할 이벤트 리포트가 더 많아질 것이다. 그러므로 결함을 관리하는 전통적인 방법은 미래 인터넷 환경에서 확장성을 가지기 힘들다. 하나의 가능한 해결책은 네트워크 장치들이 스스로를 관리하는 자율 관리 개념을 적용하는 것이다. 이 분야에 관련된 연구가 최근에 시작되었고 그 연구결과가 미래 인터넷 환경에 적용될 수 있다.

6.2 구성 관리

유비쿼터스 모바일 시스템과 가상/오버레이 네트워크는 미래 인터넷의 핵심적인 부분이다. 이 기술들은 이용자로 하여금 전통적으로 네트워크 매니저들의 배타적인 영역이었던 역할을 하게하는, 미래 인터넷 동적 구성 (dynamic configuration)이라 불리는 능력을 내포하고 있다. 여러 종류의 네트워크가 융합된 환경에서 끊임 없는 서비스를 제공하기 위해서는 터미널 장비들이 어떤 네트워크가 이용 가능한가를 인지하고 QoS 요구와 부합하는 네트워크를 선택할 수 있어야 한다. 따라서 네트워크 검색과 선택 메커니즘이 제공되어야 하고, 선택은 자율적이고 지능적으로 이루어져야 한다.

6.3 과금 관리

과금 관리는 인증 (Authentication), 권한 (Authorization) 그리고 과금 (Accounting), 요금 청구, 부과 등을 수반한다. 미래 인터넷에서의 과금 관리는 매우 중요하며 견고하고 정확한 과금 시스템이 필요하다. 서비스 제공자에 의해 제공되는 하나 이상의 서비스 이용을 원하는 이용자는 반드시 인증을 받아야 하고 (이용자의 신원 확인) 권한이 확인되어야 한다 (이용자가 해당 서비스를 이용할 권한이 있는지 여부를 확인). 권한이 확인된 이용자는 서비스를 이용할 수 있고 서비스 제공자는 과금을 위하여 이용자가 서비스 이용을 모니터링해야 한다. 미래 인터넷에서 해결해야 할 가장 어려운 과제 중 하나는 로밍 (roaming)이 필요할 때 이용자가 이용하는 네트워크를 바꾸기 위한 인증과 권한의 확인을 실시간으로 가능케 하는 것이다. 로밍이 수직적 핸드오프와 QoS에 대한 확인과 결합된다면 이것은 더욱 더 힘든 문제가 될 것이다.

또한 미래 인터넷에서 하나의 사용자 세션은 하나 이상의 서비스 제공자를 수반하게 될 것이다. 같은 서비스 제공자 내에서도 세션은 서로 다른 요금 부과 계획을 가진 다양한 네트워크들을 이용할 수도 있다. 요금 청구는 서비스 번들링 (휴대 전화, 인터넷 접속, 위성 TV 등 여러 가지 서비스의 제공)의 여러 가지 항목들에 대해 고려해야 한다. 세션 관리자는 각 세션 동안 이용자의 활동을 기록한다. 각 이용자 개인의 서비스 이용을 기록하는 요금 부과 시스템은 각 서비스 제공자의 네트워크 내에 존재해야 한다. 서비스 제공자들은 다른 서비스 제공자들과 함께 서비스 제공자들 사이의 합의된 정책에 따라 요금을 부과한다. 요금 청구 시스템은 이용자에게 부과된 요금을 기록하고 매달 청구서를 생성한다. 요금 청구 시스템은 또한 이용자의 요금 미납, 신용도 등을 기록한다.

6.4 성능 관리

미래 인터넷은 어플리케이션과 서비스들이 혼재되어 사용됨에도 불구하고 전체 동작 시간의 99.99%이상의 시간 동안 성능 지표를 만족시켜야만 한다. 이 요구는 서비스 이용자들이 서비스를 이용할 때 동의하는 SLA의 핵심적인 부분이 성능에 대한 보장임을 감안하면 꼭 필요한 사항이다. 하위 레벨에서는 네트워크의 성능이 감시되어야 하지만 상위 레벨에서는 서비스 질적인 관리가 요구된다.

SLA는 서비스 제공자와 이용자간에 공식적으로 합의된 동의이다. QoS를 보장하는 서비스 관리는 SLA의 유효기간 내의 모든 서비스 기능에 대한 통합된 관리이다. 이용자가 서비스 제공자에게 서비스를 주문할 때 SLA가 협상되며 협상이 끝나면 계약서가 작성된다. SLA계약서에는

서비스 제공자가 보증할 서비스가 명시된 QoS 파라미터가 포함된다. 서비스 제공자는 제공되는 서비스가 SLA에 명시된 QoS 파라미터들을 만족하는지 여부를 감시해야 한다. QoS 관리 시스템이 명시된 QoS 파라미터를 만족시키는지 확인하기 위해서 시스템은 반드시 네트워크 성능 감시 시스템으로부터 성능 데이터를 얻고 QoS 파라미터의 수집한 데이터들을 매치시켜야 한다.

6.5 보안 관리

미래 인터넷은 반드시 여러 가지 다양한 어플리케이션에서 요구하는 보안 요구사항을 충족시킬 수 있어야 한다. 기본적인 보안 관리는 비 권한자의 정보에 대한 접근을 막는 기밀성 (confidentiality)과 인증되지 않은 개인이나 프로세스에 의해 정보가 생성되거나 수정되지 않는 보전성 (integrity), 그리고 적합한 권한을 가진 개인이나 프로세스에 의해 데이터나 네트워크 자원에 적절한 성능을 보장하며 접근을 가능하게 하는 가용성 (availability)을 만족 시켜야 한다.

6.6 이동성 관리

이동성 관리는 수직적, 수평적 핸드오프 그리고 로밍 등을 수반한다. 수직적 그리고 수평적 핸드오프 모두 제공되어야 하며 이동성 관리 시스템은 양자 모두에 대한 지원이 요구된다. 어떤 QoS 요구를 이유로 세션을 유지하며 네트워크를 바꾸는 것을 가능케 하는 것이 핵심과제이다. 만약 어떤 이동 단말이 네트워크를 변경했고 새로 변경된 네트워크에서 QoS가 만족된다면 문제는 없을 것이다. 문제는 새로운 네트워크가 QoS를 만족시키지 못할 때 발생한다. 세션을 종료시키고 새로운 세션을 생성해야 하는지 아니면 낮은 품질의 세션을 유지해야 하는지 결정해야 한다. 대부분의 이용자가 후자를 선택할 것으로 보이나 이러한 결정들은 이용자들이 개인적인 프로파일에 미리 설정할 수 있다.

서비스의 지속을 위해서는 사전에 서비스 제공자들 사이의 이동성에 대한 동의가 있어야 한다. 이동성의 목표는 이용자로 하여금 그들의 집에서 받는 서비스를 다른 네트워크에서도 똑같이 받을 수 있게 하는 것이다. 이를 위해서 이용자의 홈 네트워크에서의 인증과 세션 매니저 안에서의 등록이 반드시 동시에 이루어져야 한다. 그렇지 않으면 하나의 네트워크에서 다른 네트워크로의 변환이 부드럽지 못해 접속이 끊어지거나 통신이 중지되는 현상을 경험하게 될 것이다.

6.7 가상 자원 관리

자원의 가상화는 중단 네트워크간 다양성의 장려를 목표로 한다. 자원 가상화를 제공하기 위해서는 먼저 새로운 인터넷 아키텍처의 중심적인 요소들을 가상화 한다는 의미를 명확히 해야 한다. [12]. 다양화된 인터넷의 창조는 현재의 억제된

네트워크 혁신에 대한 진입 장벽을 허물 수 있는 잠재력을 가졌고 새로운 네트워크 기술과 개선된 어플리케이션들의 배치를 더 자유롭게 할 수 있는 경쟁적인 환경을 가능하게 한다. 우리는 어떻게 가상화가 인터넷 아키텍처를 통해 통합될 수 있는가와 경제적 타당성을 증명하기 위해 그것들이 어떻게 다양한 협력 단체에 의해 충분한 고성능으로 규모가 큰 환경에 전달될 수 있는지에 대해 고려해야 한다. 가상화 관리는 다양화된 인터넷의 완전한 아키텍처의 개발과 그 아키텍처의 주요 구성 요소에 대한 실험을 통한 검증을 포함한다. 이것은 다양한 단체에 의해 운영되고 소유되는 기반 시설 내에 현재의 아키텍처와 다양한 종단간 네트워크의 공존을 가능하게 하는 기반 프로토콜의 개발도 포함한다. 즉 이것은 두 개의 완전한 종단 네트워크가 다양화된 인터넷에서 얼마나 효과적으로 이용될 수 있는지 여부와 얼마나 가상 자원을 효율적으로 관리할 수 있는가를 탐구하는 것이다.

6.8 레이어 간의 서비스 관리

크로스 레이어드 서비스를 제공하려면 먼저 레이어 간의 상호작용을 지원해야 한다. 크로스 레이어 제어 메카니즘은 서비스와 구체적인 메소드 인터페이스를 알맞게 조정하고 서비스들 간의 상호작용을 용이하게 하는 역할을 한다. 크로스 레이어 서비스는 그 서비스가 포함하고 있는 하위 서비스와 스택에 들어갈 하위 서비스들의 순서, 그리고 각각의 서비스들을 구현할 메소드 등으로 이루어졌다. 크로스 레이어 제어 시스템은 어플리케이션의 현재 네트워크 자원의 가용성과 다른 조건들, 서비스들 간의 우선순위, 그 시점에 영향을 미치는 정책 등과 QoS 요구사항을 고려한다. 현재의 정책은 다양한 범위의 하나의 또는 여러 개의 네트워크 정책뿐만 아니라 각 노드들의 정책들(절전 모드)의 조합으로부터 추출된다.

크로스 레이어드 아키텍처의 의도적인 설계의 특징은 명시적으로 크로스 서비스를 최적화하는 기능을 제공하는 것이다. 서비스들의 특징과 서로의 상호작용을 설명하기 위해서는 서비스의 추상화가 가능해야만 한다. 그리고 우리는 많은 실험과 특별한 목적의 서비스들을 기대한다. 그 중에서 가장 성공적인 실험이나 서비스들이 표준화되기 때문이다.

6.9 터미널 관리

터미널 관리는 터미널의 장소와 터미널의 트레이스 두 가지를 수반한다. 터미널 장소 정보는 이용자가 특정한 서비스를 원하고 어떤 QoS 파라미터들을 만족시키기를 원할 때 관리 시스템과 오퍼레이션의 QoS에 관련된 결정에 이용된다. 또한 이러한 정보는 핸드오프가 필요할 때나 잃어버린 아이나 배우자의 위치를 찾아주는 새로운 서비스들

제공할 때 유용하다.

터미널 트레이스 관리는 현재 서비스 제공자에게 가장 큰 골칫덩어리 중 하나인 도난당한 터미널이나 거짓 터미널을 감지하기 위한 중요한 기능이다. 전세계적으로 잃어버리거나 도난당한 터미널들이 많이 있다. 이러한 터미널들이 불법적으로 소유자가 아닌 다른 이들에 의해 이용될 때, 터미널을 추적하기 위해 터미널 트레이스관리가 이용될 수 있다. 트레이스 정보는 또한 문제를 해결하거나, 기능이 제대로 작동하지 않는 터미널의 근본적인 원인을 찾거나, 자원의 사용과 질을 최적화, RF 커버리지와 능력 개선, 끊어진 통화 분석 등에 핵심적인 역할을 할 수 있다

7 결론

우리 사회의 많은 가치 있는 어플리케이션, 경제 그리고 디지털 라이프 [10]의 지원에 힘입어 현재의 인터넷은 앞으로도 진화할 것이다. 그럼에도 불구하고 현재 인터넷은 이동성, 보안, 관리 또는 확장성에서 비롯될 미래의 문제들을 고려해서 설계되지 않았다. 지금은 미래 인터넷의 기반 시설을 설계 하기 위한 새로운 아이디어에 대한 연구를 시작할 시기이다. 미래 인터넷을 디자인하고 테스트 설비를 만드는 다수의 연구활동이 진행되고 있다. 이러한 연구 및 개발 (R&D) 활동은 국제적인 협력을 필요로 하며 널리 개방되어 있다. 선진국 및 개발 도상국의 모든 지역의 관심 있는 단체들은 미래와 세계의 네트워크화된 경제에 대한 이해를 공유하고 개발에 참여할 기회를 가진다.

우리는 연구단체가 네트워크제어와 관리의 문제에 대한 진화한 클린슬레이트 (clean-slate) 방식의 접근 방법을 추구할 기회가 있음을 믿는다. 만약 이 시도가 성공적이라면, 연이은 연구가 네트워크를 연구하는 학자들로 하여금 그들의 아이디어를 실제 네트워크에 배치하는 것에 대한 가능성을 전망할 수 있다. 궁극적으로 새로운 제어와 관리 프로토콜, 그리고 소프트웨어가 갖추어진 데이터 네트워크는 더 단순하고, 더 견고하고, 더 진화할 수 있고, 보안이 더 강화될 수 있다. 우리는 미래 인터넷의 관리 문제에 집중할 것이고 미래 인터넷의 관리를 위해 가능한 연구 범위를 좁힐 것이다. 앞으로는 미래 인터넷을 위한 각 관리 항목들을 어떻게 관리 할 것인가에 대한 세부적인 연구가 필요하다.

8 참고 문헌

- [1] Michael M.Roberts, "Lessons for the Future Internet: Learning from the Past", EDUCATE Review, vol.41, no.4, July/August 2006, pp.17~24
- [2] C. Jin, D. X. Wei, and S. H. Low, "FAST TCP: Motivation, architecture, algorithms, performance", In

- Proceedings of the 2004 IEEE INFOCOM Conference, pp. 2490~2501, Hong Kong, March 2004
- [3] D. Katabi, M. Handley, and C. Rohrs, "Tussle in cyberspace: Defining tomorrow's internet", In *Proceedings of the 2002 ACM SIGCOMM Conference*, pp. 89~102, Pittsburgh, PA, August 2002.
 - [4] Leslie Ellis. BitTorrent's swarms have a deadly bite on broadband nets. <http://www.multichannel.com/article/CA6332098.html>, 2006.
 - [5] K. Hafner and J. Markoff. *Cyber punk-outlaws and hackers on the computer frontier*. 1995. ISBN: 1-872180-94-9.
 - [6] Darleen Fisher, "US National Science Foundation and the Future Internet Design" *ACM SIGCOMM Computer Communication Review*, Volume 37, Issue 3, July 2007, pp. 85~87.
 - [7] Anastasius Gavras et al., "Future internet research and experimentation: the FIRE initiative", *ACM SIGCOMM Computer Communication Review*, Volume 37, Issue 3, July 2007, pp. 89~92
 - [8] Siekkinen, M. et al., "Beyond the Future Internet--Requirements of Autonomic Networking Architectures to Address Long Term Future Networking Challenges", 11th IEEE International Workshop on Future Trends of Distributed Computing Systems (FTDCS'07), 11th IEEE International Workshop on 21-23 March 2007, pp. 89~98
 - [9] Japan Giga Network 2 (JGN2), <http://www.jgn.nict.go.jp/english/index.html>, Refer Dec. 2007
 - [10] National Institute of Information and Communications Technology (NICT), <http://www.nict.go.jp>, Refer Dec. 2007.
 - [11] Thrasylvoulos Spyropoulos et al., "Future internet: fundamentals and measurement", *ACM SIGCOMM Computer Communication Review*, Volume 37, Issue 2, April 2007, pp. 101~106.
 - [12] Larry Peterson and John Wroclawski, "Overview of the GENI Architecture", GENI document, GDD-06-11, January 2007.
 - [13] Albert Greenberg, Gisli Hjalmytysson, David A. Maltz, Andy Myers, Jennifer Rexford, Geoffrey Xie, Hong Yan, Jibin Zhan, Hui Zhang, "A Clean Slate 4D Approach to Network Control and Management", *ACM SIGCOMM Computer Communication Review*, Volume 35, Issue 5, October, 2005.
 - [14] Dutta. R, Rouskas. G. N, Baldine. I, Bragg. A, and Stevenson. D, "The SILO Architecture for Services Integration, control, and Optimization for the Future Internet", *Communications, ICC '07. IEEE International Conference on June 2007.*
 - [15] Future Internet Forum (FIF), <http://mmlab.snu.ac.kr/fif>, Refer Dec. 2007.
 - [16] Paul Barford, Suman Banerjee, Cristian Eitan, "Design for Manageability in the Next Generation Internet", FIND project, <http://www.nets-find.net/Funded/Manageability.php>, Refer Dec. 2007.