

OpenFlow/SDN Paradigm 에 따른 네트워크 인프라 구축의 혁신이란?

문서 상태: 버전 0.2 (최종수정일: 2011-12-21)

주의: 본 이슈 분석서의 버전 0.2 은 잠정적으로 공개하여 보완 의견을 청취할 목적으로 2011 년 12 월에 준비되었으며, 향후 보완 과정을 거쳐서 정식 버전 1.0 을 만들 예정임.

작성자: 김종원, 김남곤 (광주과학기술원 정보통신공학부), 남기혁 (ETRI)
{jongwon, ngkim}@nm.gist.ac.kr, nam@etri.re.kr

요약: 미래인터넷 테스트베드는 연구자들이 창의적인 서비스 실현을 실증할 수 있도록 연결용 네트워크와 이에 연동된 컴퓨팅 장비들, 그리고 이를 지원하는 각종 실험용 소프트웨어 일체를 지칭하는 일종의 가상적인 실험실이어야 한다. 활용되는 컴퓨팅/네트워킹 자원들이 여러 네트워크 계층에서 프로그램이 가능해야 하며, 다수의 실험자들이 자신들의 실험을 동시에 수행하도록 자원의 가상화도 필요하다. 하지만 네트워킹 자원들은 프로그램화/가상화 부분에서 매우 미진한 상황이며, 네트워킹 자원들이 사용자가 원하는 대로 활용하려면 새로운 패러다임에 따라서 네트워킹 장비를 개발하는 혁신이 필요하다는 주장이 강하게 대두되고 있다. 즉 현존하는 “프로토콜” 중심의 네트워킹 패러다임이 가지는 근본적인 문제점을 -- 지나치게 많은 (또한 대부분 실제로 사용되지 않는) 호환성 검증이라는 장애물들을 -- 제대로 인식하고, 이를 **소프트웨어 정의 네트워킹 (Software Defined Networking: SDN)**으로 불리는 새로운 네트워킹 패러다임으로 해소해야만 한다는 것이다. SDN 을 채택해서 논리적으로 중앙집중된 컨트롤러를 통해서 네트워킹 장치들을 **오픈플로우(OpenFlow)와 같은 표준적인 인터페이스를 활용하여** 직접 프로그래밍하면서 제어하자는 것이다. 따라서 본 이슈분석서에서는 SDN 에 근거한 네트워킹 패러다임에 대한 비전을 이해하는 차원에서 이에 대한 전체적인 비전 리뷰, 기술적인 세부 내용, 그리고 향후 예상되는 변화를 살펴보고자 한다.

주제어: 미래인터넷 테스트베드, 컴퓨팅/네트워킹 자원, 자원의 프로그래밍 및 가상화, 소프트웨어 정의 네트워킹 (Software Defined Networking), 오픈 플로우

목 차

1. 서론.....	1
2. OPENFLOW/SDN 을 통한 네트워킹 분야의 혁신이란?.....	3
3. OPENFLOW 인터페이스 및 이에 기반한 SDN 구현 현황	7
3.1. OPENFLOW 1.0/1.1.....	8
3.2. OPENFLOW 에 기반한 SDN 구현 사례.....	10
3.3. OPENFLOW 전용 언어.....	12
4. ONF 를 중심으로 한 SDN 확산 및 향후 전망	14
4.1. ONF 소개 및 SDN 확산 계획	14
4.2. 향후 전망	ERROR! BOOKMARK NOT DEFINED.
5. 참고문헌	18

1. 서론

미래인터넷 테스트베드는 연구자들이 창의적인 서비스 실현을 실증할 수 있도록 연결용 네트워크와 이에 연동된 컴퓨팅 장비들, 그리고 이를 지원하는 각종 실험용 소프트웨어 일체를 지칭하는 일종의 가상적인 실험실이어야 한다. 활용되는 컴퓨팅/네트워킹 자원들이 여러 네트워크 계층에서 프로그램이 가능해야 하며, 다수의 실험자들이 자신들의 실험을 동시에 수행하도록 자원의 가상화도 필요하다. 하지만 네트워킹 자원들은 프로그램화/가상화 부분에서 매우 미진한 상황이며, 네트워킹 자원들이 사용자가 원하는 대로 활용하려면 새로운 패러다임에 따라서 네트워킹 장비를 개발하는 혁신이 필요하다는 주장이 강하게 대두되고 있다. 이러한 논리를 좀 더 쉽게 설명하자면 아래와 같다.

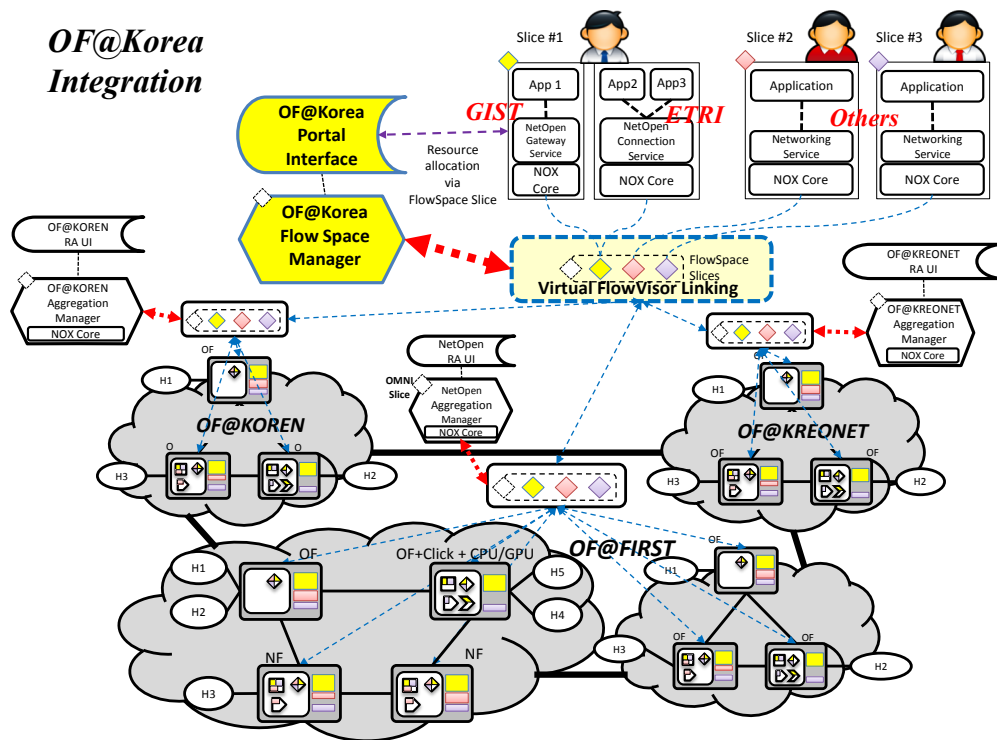


그림 1. OpenFlow 를 이용한 네트워킹 장비들에 대한 플로우 집합별 효과적인 공유.

많은 사람들이 같이 공유하는 다세대 공동주택과 같은 것이 인터넷을 지탱하는 네트워크 장비들의 핵심적인 역할이다. 하지만 같이 사용할 때 지켜야 하는 약속들이 지나치게 많고 복잡해서 문제가 되고 있다. 약속들을 최소화해서, 예를 들면 공동주택에 함께 출입하는 규칙과 같은 기본적인 몇 가지 만 합의해서, 사용하면 어떻게 될까? 즉 공동주택의 자기 방에 출입하기 위한 접근 (즉 공용 복도와 방 출입구에 대한) 방법만을 규칙으로 만들고, 자신의 방 내부를 어떻게 만들어 무엇을 할 것인가는 각자 알아서 하는 것이다. 그림 1 에 제시한 바와 같이 OF@FIRST, OF@KOREN, OF@KREONET 으로 표시된 네트워크 장비들의(전국 각지에 산재한 공동주택들의) 관리자들이 지역에 상관없이 협조하여, 노랑/분홍/보라색상으로 구분된 플로우들의 집합마다 할당된 사용자들을 구별하고 (즉 허가된 사람들만 들여보내고) 원하는 네트워크를 맘대로 하도록 (옆 방 사람들을 방해하지 않는 한에서) 만드는 것이다. 구체적으로 살펴보면, 지금의 네트워크 장비 개발 패러다임은 네트워크 장비 각각이 여러 계층에 걸친 수 많은 프로토콜을 지원해서 동작하도록 만들고, 이의 성능을 개별적으로 최적화한 후에, 이들 각각에 대한 검증 과정을 거치는 개별 장비 중심으로 형성되어 있다. 따라서 이를 벗어나기 위해서 개별 장비들에게는 일반 상용 PC 와 같은 핵심적인 자원 기능들에 대한 검증된 성능만을 요구하고, 기본적인 프로토콜들 몇 개를 제외한 대다수의 프로토콜들 자체가 불필요하게 패러다임을 바꾸는 발상의 전환이 필요한 것이다. 즉 현존하는 “프로토콜” 중심의 네트워크 패러다임이 가지는 근본적인 문제점을 -- 지나치게 많은 (또한 대부분 실제로 사용되지 않는) 호환성 검증이라는 장애물들을 -- 제대로 인식하고, 이를 **소프트웨어 정의 네트워크 (Software Defined Networking: SDN)**으로 불리는 새로운 네트워크 패러다임으로 해소해야만 한다는 것이다. SDN 을 채택해서 논리적으로 중앙집중된 컨트롤러를 통해서 네트워크 장치들을 **오픈플로우(OpenFlow)와 같은 표준적인 인터페이스를 활용하여** 직접 프로그래밍하면서 제어하자는 것이다.

따라서 본 이슈분석서에서는 SDN 에 근거한 네트워크 패러다임에 대한 비전을 이해하는 차원에서 이에 대한 전체적인 현황, 기술적인 세부 내용, 그리고 향후 예상되는 변화를 차례로 살펴보고자 한다. 먼저 2 절에서 OpenFlow/SDN 을 이용한 네트워크 패러다임의 혁신이라는 주제를 논의한다. 이어서 3 절에서는 현재까지 만들어진 OpenFlow 라는 표준적인 제어 인터페이스를 중심으로 관련된 기술 이슈들을 소개하도록 한다. 그리고 4 절에서 ONF (Open Networking Forum)의 활동을 중심으로 SDN 에 대한 전략적인 개발이

어떻게 전개하려고 시도되는 지를 설명하면서 향후의 변화를 전망해 보면서 본 문서를 맺는다.

2. OpenFlow/SDN 을 통한 네트워킹 분야의 혁신이란?

미래에 인터넷을 통해서 사용자가 원하는 서비스를 제공하기 용이한 새로운 차원의 인프라로 나가기 위한 방법들의 하나로 네트워킹 장비 개발과 운용 분야에 혁신 (innovation)이 필요하다고 한다. 전세계 사람들을 항상 연결해주고 있는 인터넷을 떠받치고 있는 근간인 각종 라우터와 스위치 장비들이 계속적으로 고도화 되면서 지속적으로 늘어나는 인터넷 트래픽을 비교적 무난하게 처리하고 있는 실정인데 말이다. 네트워킹 분야의 혁신은 사용자가 원하는 새로운 네트워킹 아이디어가 네트워킹 장비에 빠르게 적용되고 또한 손쉽게 운용될 수 있는 환경이 조성되어야 가능할 것이다. 그런데 지금의 네트워킹 장비 개발과 운용은 “프로토콜” 중심의 고전적인 패러다임이 요구하는 지나치게 많은 (또한 대부분 실제로 사용되지 않는) 연동성과 각종 호환성 검증이라는 장애물을 넘어서야만 사용자의 선택을 받을 수 있는 상황이다.

장비의 사용자와 개발자 간의 다음 대화를 일례로 살펴보자. “MPLS-VPN 기능이 지원되나요? 네. 그런데 지원이 제대로 되는지 검증 자료는 있으신가요? 네, 현재 검증이 진행되고 있습니다. 그러면 IPv6 지원에는 문제가 없으신가요? 네, 지원하고 있고 검증 자료도 가지고 있습니다. 그런가요? 그럼 저희가 보유하고 있는 C사의 장비와의 호환성은 어떠한가요? 네, X사에서 연동해서 사용 중인데 큰 문제가 없습니다. ...” 이와 같이 야심 차게 개발한 새로운 네트워크 장비를 고객에게 팔기 위해서는 위와 같은 질문들 모두에 대한 만족스러운 대답들을 제시해야 비로소 구매를 위한 검토 대상에 포함되는 영광을 누리게 된다. 이와 같이 네트워크 장비 개발과 이의 시장 확보라는 과제를 수행함에 있어서 넘어야 할 산은 너무나 많은 것이 현실이고, 이는 기존에 개발, 검증, 적용/보완의 전반에 걸쳐 생태계를 구축해 놓은 장비 회사들만의 독과점을 조장하게 된다. 새로운 장비의 개발과 시장 적용을 위해서 노력을 경주했던 네트워크 장비 업체들을 제외한 대다수 당사자에게 소위 “검증된” 제품에 대한 구매는 편안한 선택일 수 밖에 없기 때문이다.

앞에 논의한 비합리성이 존재할 수 밖에 없는 이유를 기존의 네트워킹 패러다임이 가지는 근본적인 문제점으로 인식하고, 이를 소프트웨어를 정의한다는 시각으로 네트워킹을 새롭게 바라보면서 **소프트웨어 정의 네트워킹 (Software Defined Networking: SDN)**으로 불리는 새로운 네트워킹 패러다임을 함께 개척하자는 목소리가 나타나고 있다. 즉 복잡성을 즐기는 엔지니어들이 설계하는 네트워킹을 벗어나서 단순함만이 통하는 사용자들이 손쉽게 이해하고 운용할 수 있는 새로운 네트워킹 패러다임으로 혁신해 보자는 것이다.

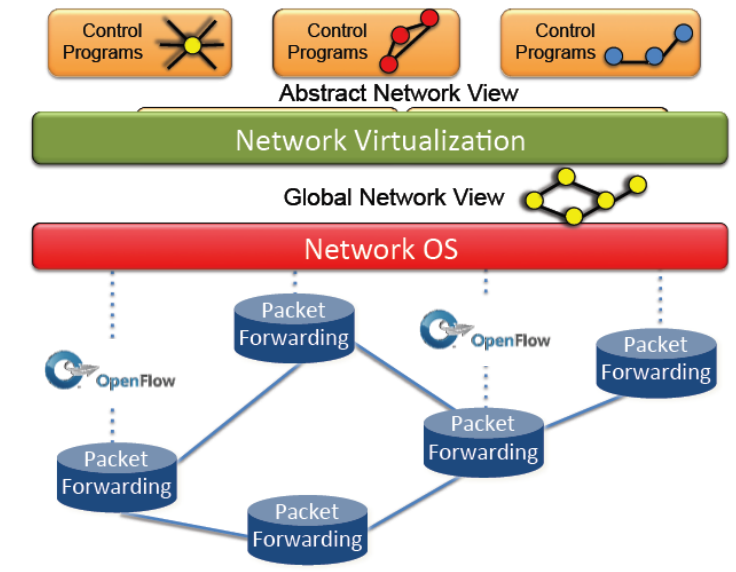


그림 2. Software-Defined Networking and OpenFlow [from Nick McKweon's ONS Presentation Slide].

좀 더 구체적으로 살펴보자면, 지금의 네트워킹 패러다임은 네트워크 장비 하나 하나가 여러 계층에 걸친 수 많은 프로토콜을 지원해서 동작하도록 만들고, 이의 성능을 개별적으로 최적화한 후에, 이들 각각에 대한 검증 과정을 거치는 개별 장비 중심으로 형성되어 있다. 따라서 이를 벗어나기 위해서는 개별 장비들에게는 일반 상용 PC 와 같은 핵심 자원 기능들에 대한 검증된 성능만을 요구하고, 핵심적인 프로토콜들을 제외한 수 많은 프로토콜들 자체가 불필요하도록 상황을 바꾸는 발상의 전환이 요구된다. 이를

위해서 SDN 을 채택하면 그림 2 에 제시한 바와 같이 네트워크 OS (Network Operating System, 즉 네트워크를 위한 Windows 또는 Linux 로 생각하면)를 간단하게 프로그래밍을 제어하기 위한 **오픈플로우(OpenFlow)** 라고 불리는 **인터페이스 부분만 표준적으로 합의하여 만들어서** 이를 통해서 개별 장비들은 네트워크 OS 와의 포팅에 전념하게 한다. 그리고 수 많은 프로토콜에 대한 대응은 논리적으로 중앙집중된 컨트롤러를 통해서 “합의된, 즉 표준화된 프로토콜은 필요가 없는 상태로” 해당 환경에 부합되는 적절한 알고리즘의 직접적인 구현으로 손쉽게 해결하도록 하면 된다. 예를 들면 교과서적인 네트워크 라우팅의 구현 사례인 Dijkstra 의 알고리즘이 장비들의 상태 맵(map)을 파악하고 있는 중앙집중형 컨트롤러에 동작하는 원리는 4 페이지 문서면 모두 설명할 수 있다. 그런데 이를 분산 시스템에서 네트워크 맵을 최신으로 일관되게 분산 방식으로 업데이트하면서 구현하려면 101 페이지에 걸친 문서에 근거해서 구현해야 하면, 실제 장비에 적용되는 표준화된 OSPF 라우팅을 위한 RFC 2328 문서는 245 페이지에 걸쳐서 프로토콜을 정리하고 있다. 즉 4 페이지의 간단한 알고리즘 구현을 50 배나 복잡하게 만들어서 구현하고 또한 이에 부합하는 지, 그리고 유사 장비들과 호환이 되는 지를 일일이 검증하는 난센스가 벌어지고 있는 것이다. 따라서 SDN 을 기반으로 한 새로운 패러다임을 만들어 내어서 아래와 같은 혜택들을 누리는 네트워크 장치 생태계를 만드는 것이 필요하다고 예기되고 있다.

1. 포워딩(forwarding) 추상화에 따라서 오픈플로우 표준에 의해 검증된 하드웨어를 이용하고, 또한 이에 연동하여 소프트웨어 기반으로 제공되는 네트워킹 특성이 요구되는 모든 절차들에 대해서 각각의 절차마다 충분히 증빙되어 있는 견실한 네트워킹을 실현할 수 있는 토대를 구축할 수 있다.
2. 장비 사용자들이 자신의 필요에 따라 네트워킹을 유연하게 구성(customize)하고 불필요한 구성요소들은 과감하게 제거하면서 자신만을 위해서 가상화된 네트워크를 생성하기 쉽도록 지원한다.
3. 하드웨어 추상화(abstraction)에 따라 확장성을 고려한 상태에서 공통화된 (즉 commodity 형식으로) 하드웨어를 구입하고, 또한 소프트웨어도 분리해서 구입하도록 하여 기존의 폐쇄적인 네트워크 장치 공급자 체인을 벗어나서 자체 개발, 외주

개발, 오픈 소스 형식을 모두 포함하는 다변화된 공급자 체인으로 체질 개선을 유도할 수 있다.

4. 소프트웨어를 개발하는 속도로 혁신이 일어나도록 하고, 표준은 구현된 소프트웨어의 확산을 위해 뒤따라가는 방식을 취하고, 소프트웨어 적인 개방성에 근간하여 기술의 공유 협력을 쉽도록 함으로써 혁신의 속도를 가속하도록 지원한다.

이러한 실용주의적인 SDN 을 함께 추진하기 위해 ONF (Open Networking Foundation)이 2011 년 상반기에 Deutsche Telekom, Facebook, Google, Microsoft, Verizon, Yahoo 를 이사회 멤버로 하여 정식으로 출범했다. 즉 ONF 라는 회원사 공동체를 중심으로 SDN 기술을 개발하고 주요 표준도 구현에 기반하여 빠르게 제정해서 확산을 도모해 보자는 것이다. 또한 지난 달 (2011 년 10 월 17-19)에는 OpenFlow 를 최초로 제안했던 미국 Stanford 대학에서 ONF 주관으로 첫 번째 Open Networking Summit 행사가 개최되었다. 주최측이 예상을 두 배가 넘는 폭발적인 관심으로 제한된 400 여명의 인원들만 참가할 수 있었으며, 다양한 영역들에서 온 참가자들이 3 일에 걸쳐서 관련 데모와 발표를 공유하면서 많은 논의를 나눴다. 특히 SDN 이 가지는 융합적인 성격을 반영하여, 추상화개념부터 시작해서 분산 시스템, 데이터베이스, 소프트웨어 공학 등 다양한 컴퓨터사이언스(Computer Science)의 개념을 접목해 보려는 참가자들이 다수 눈에 띄었다. 이번 행사의 keynote 발표에서는 기존의 프로토콜 중심의 네트워크 구축 방법을 버리고 오픈플로우 인터페이스를 통해서 소프트웨어 기반으로 새로운 네트워킹으로 혁신하는 노력을 함께 시작해 보자는 메시지가 전달되었던 행사였다. 또한 현재 시점의 오픈플로우 지원 장비 현황과 이에 기반한 SDN 의 개념을 보여주는 12 개의 데모가 3 차례에 걸쳐서 진행되었으며, 한국에서도 FIF 테스트베드 WG 에서 FIRST/KOREN/KREONET 공동으로 준비한 OF@Korea 데모를 진행하였다 (<http://opennetsummit.org/demonstrations.html>).

또한 실제 네트워크 인프라에 SDN 을 어떻게 적용해서 개선할 수 있을 것인가 대해서는, 지금 막 태동하는 노력이므로 현실을 반영하여 데이터센터, 엔터프라이즈, 네트워크 서비스 제공자의 순서로 점차적으로 적용이 확산되는 것에 대해 공감대를 형성되고 있다. 특히 오픈플로우 기반 SDN 이, 다양한 장치들과 가상 머신들이 병존하면서 네트워크

토폴로지를 묶고 있는 데이터센터, 개인화된 클라우드, 그리고 캠퍼스랜에서 유용하다는 주장이 많았다. SDN 이 클라우드/데이터센터에서 요구되는 로드밸런싱, 플로우제어, 가상화된 네트워크를 지원하는 손쉽게 운용함에 효과적이라는 판단인 것이다. 데이터센터나 엔터프라이즈 네트워크 전반에 걸쳐서는 네트워크 관리 측면의 혜택이 부각되었으며, 앞에서 설명한 4 페이지 대 수백 페이지 비유에서 설명된 바와 같이, 현재 장비가 가진 기능들의 극히 일부만 보수적으로 운영하는 분위기가 SDN 도입으로 보다 풍부한 기능을 활용하면서 동적으로 관리될 것으로 전망되었다. 즉 새로운 SDN 기반 패러다임에서는 다양하고 독창적인 구성을 지닌 라우터와 스위치들로 구성되지만 제공하는 네트워크에 대한 통합된 제어가 일사불란하게 가능하도록 만드는 “네트워크 OS” 같은 역할이 핵심적이라는 것이다.

그러므로 한국에서도 이러한 비전에 대한 대응 전략에 대해서 곰곰히 생각해 볼 시점이 아닌가 생각된다. 한국의 국가적 장점인 초고속 인터넷 보급과 확산에 있어서 세계를 선도했던 경험과 이를 지원했던 네트워크 하드웨어에 대한 신속한 적응력에도 불구하고, 지속적으로 유지하고 관리하면서 구색을 갖추어야만 하는 소프트웨어 적인 부분의 부진으로 인해서 돌파구를 잃고 있는 국내의 네트워크 산업에게도 새로운 기회를 부여할 수 있는 다가오는 큰 물결(Big Wave)과 같은 역할을 할 수도 있기 때문이다. 스마트 폰 개발에 있어서 시기를 놓쳐서 힘겹게 쫓아가면서 깨달은 소프트웨어 마인드를 “스마트 네트워크” 장비의 개발에는 선제적으로 전용할 수 있을 지도 모르기 때문이다.

3. OpenFlow 인터페이스 및 이에 기반한 SDN 구현 현황

OpenFlow 의 개발을 주도하고 있는 스탠포드 대학에서는 OpenFlow 스위치의 요구사항을 담은 스펙(specification) 문서를 공개하여, 누구나 다른 OpenFlow 프로토콜 기반의 플로우 기반 네트워크 능력을 지닌 장치를 개발할 수 있도록 하였다. 그 결과, 소프트웨어 형태의 OpenFlow 스위치뿐만 아니라, Cisco, HP, IBM 과 같은 많은 상용 네트워크 장비 벤더들도 기존 장비의 firmware 업데이트 수준에서 OpenFlow 스위치를 지원하고 있다. OpenFlow 스위치의 스펙 문서는 2010 년 1 월에 1.0 버전의 스펙[8]이 참조구현(reference implementation)과 함께 공개되었고, 그 후 2011 년 3 월에 1.1 버전의 스펙[9]을

공개하였다. 하지만, 1.1 버전에 대응되는 장치들은 매우 제한된 상태이므로 현 시점까지는 1.0 버전이 주로 활용되고 있는 상황이다. 본 문서에서는 OpenFlow 1.0 과 1.1 버전의 주요 기능들을 살펴보고, 그에 기반한 SDN 구현 현황을 소개한다.

3.1. OpenFlow 1.0/1.1

OpenFlow 1.0 스펙 문서는 수 년간의 작업을 통해 안정화된 OpenFlow 스위치의 모습을 담고 있다. OpenFlow 스위치로 동작하기 위한 기본적인 내용과 함께, 플로우 테이블의 형태와 네트워크 컨트롤러와 연결을 위한 인터페이스인 secure channel 을 구성하기 위한 메시지 포맷을 정의하고 있다. 플로우 테이블의 각 엔트리는 플로우와 매칭할 패킷의 헤더 필드와 해당 플로우에 대한 스위치의 행동 (action) 목록, 그리고 일치하는 패킷에 대한 카운터를 포함한다.

Secure channel 을 구성하기 위해 controller-to-switch, asynchronous, symmetric 의 세 가지 메시지 타입을 정의하고 있다. Controller-to-switch 메시지 타입은 네트워크 컨트롤러가 스위치에게 보내는 메시지로, 스위치의 능력에 대한 정보를 요청하는 features, 플로우 테이블 엔트리를 추가/삭제/수정하기 위한 modify-state 메시지, 그리고 스위치의 상태 정보를 수집하기 위한 read-state 등을 포함하고 있다. 또한, 초기적인 형태의 QoS 제공을 위한 슬라이싱에 대한 내용을 담고 있다. Asynchronous 메시지 타입은 스위치가 상태 변화가 생겼을 때 이를 네트워크 컨트롤러에게 알리기 위해 사용하는 것으로, Packet-in, Flow-removed, Port-status, Error 메시지 등이 정의되어 있다. Packet-in 메시지는 플로우 테이블에 매칭되는 엔트리가 없을 시에 이를 네트워크 컨트롤러에게 전달하기 위해 이용되고, flow-remove 메시지는 정해진 시간 동안 패킷이 입력되지 않아 플로우 엔트리가 지워질 경우에 이용된다. Symmetric 메시지 타입은 네트워크 컨트롤러나 스위치 누구든지 요청할 수 있는 것으로, Hello, Echo, Vendor 메시지가 있다. 이 중 Vendor 메시지는 각 스위치 벤더들이 향후에 추가적인 기능을 OpenFlow 메시지 타입 공간 안에서 구현할 수 있도록 표준화된 방법을 준비해 둔 것이다.

OpenFlow 1.1 의 스펙 문서는 2011 년 3 월에 공개되었는데 OpenFlow 1.0 과의 호환성을 유지하면서 파이프라인 프로세싱과 그룹의 개념을 추가해 다중 경로 (multi-path) 지원이나, 개별 플로우의 프로세싱을 보다 다양하게 제어하기 위한 지원이 확대되었다.

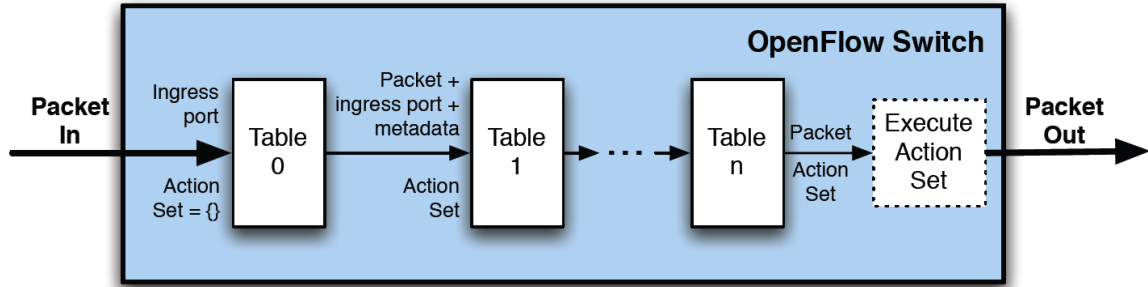


그림 3. 파이프라인을 통한 플로우 처리.

먼저 가장 큰 변화는 플로우에 대한 파이프라인 프로세싱을 추가한 것으로, 1.0 버전에서는 하나의 플로우 테이블만을 이용하였으나, 그림 3 과 같이 다수의 플로우 테이블을 이용하도록 변경되었다. 파이프라인 프로세싱을 위해 먼저 각 패킷별 행동 정보를 Action Set 을 통해 정의한다. Action Set 은 패킷이 각 플로우 테이블에 의해 처리되면서 추가되는 행동의 집합으로 패킷이 프로세싱 파이프라인을 빠져 나왔을 때 수행될 액션들을 나타낸다. 플로우 테이블은 1.0 버전과 같이 여러 플로우 엔트리를 가지고 있지만, 플로우 테이블 엔트리가 행동 목록을 직접 표현하는 대신 instruction 의 목록을 담고 있다는 점이 달라졌다. Instruction 은 각 패킷에 연결된 Action Set 을 수정하거나 파이프라인 프로세싱을 위한 지시사항을 표현하는데, Apply-Actions, Clear-Actions, Write-Actions, Write-Metadata 그리고 Goto-Table 등의 메시지가 있다. 이 중 Write-Actions 는 패킷의 Action Set 에 행동을 추가하고, Goto-Table 은 파이프라인의 다음 테이블을 가리켜 기본적인 파이프라인 프로세싱을 지원한다.

또한 새로운 행동으로 group 이 추가되어 다양한 형태의 포워딩을 지원한다. 그룹은 하나의 고정된 출력 포트를 지정하는 형태가 아닌, 여러 개의 출력 포트를 동시에 이용하거나, 선택적으로 이용하는 것을 지원한다. 이를 통해 멀티캐스트나 브로드캐스트

뿐만 아니라, 다중 경로 중에 선택적으로 하나의 경로를 이용하는 것을 쉽게 지원하고 있다. Group 의 개념을 지원하기 위해 플로우 테이블 이외에 그룹 테이블이 새로 추가되었다.

3.2. OpenFlow 에 기반한 SDN 구현 사례

OpenFlow 가 소개된 이후로 다양한 네트워킹 환경에 OpenFlow 를 적용한 구현 사례들이 소개되고 있다. 초기의 OpenFlow 의 활용은 경로의 제어에 집중되어 있었지만, 차츰 OpenFlow 의 프로그래머블 네트워킹 능력을 응용이 요구하는 네트워킹을 위해 활용하는 결과물들이 많이 소개되고 있다. 현재는 OpenFlow 의 활용성이 높을 것으로 예상되는 데이터센터 내부에서 이용될 수 있는 데이터 센터 전원 관리, 다중 경로 전송 등의 기능들이 SDN 을 기반으로 구현되어 소개되고 있다.

가장 많이 소개되고 있는 구현사례로 웹서버의 로드 밸런싱을 제공하는 Aster*x [10]가 있다. Aster*x 는 스탠포드 대학에서 OpenFlow 를 기반으로 만든 결과물로 네트워크의 혼잡 상황과 더불어 서버의 부하까지 고려한 효율적인 로드 밸런싱을 제공한다. 서버와 네트워크 상태 정보들을 수집하고, 이를 바탕으로 사용자의 웹 서버에 대한 요청을 제어하는 데 이용함으로써 Aster*x 는 클라이언트의 응답 시간을 줄이고, 시스템 비용을 줄일 수 있는 효율적인 로드 밸런싱 시스템을 선보였다.

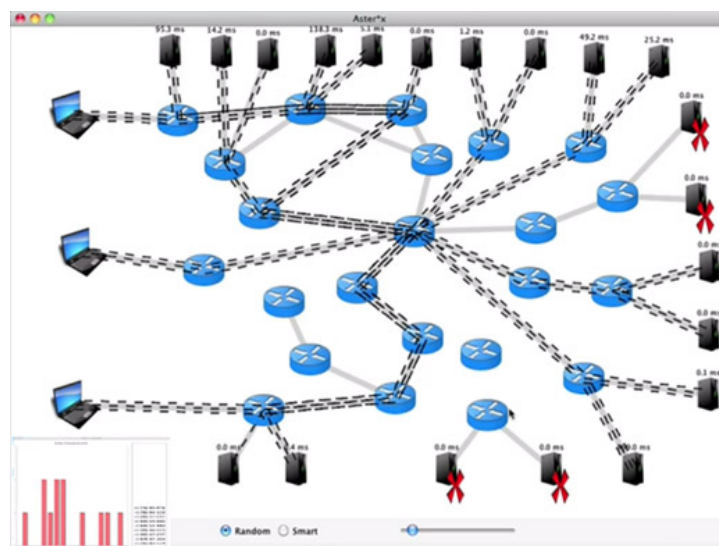


그림 4. Aster*x 의 Front-end GUI.

무선 환경에서 OpenFlow 를 적용한 대표적인 사례로 N-casting [11]을 들 수 있다. N-casting 은 무선 단말이 지닌 다수의 Wi-Fi, WiMAX 인터페이스를 동시에 활용하여 전송의 신뢰성을 높일 수 있음을 보였다. OpenFlow 를 지원하는 WiFi AP 와 스위치, 라우터 들을 활용하여 패킷 복제를 동적으로 제어함으로써 단말이 지닌 다수의 무선 인터페이스를 통해 패킷 전송의 신뢰성을 높인다.

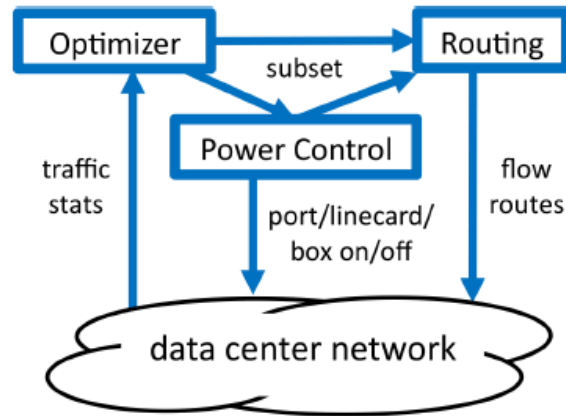


그림 5. Elastic Tree 의 시스템 다이어그램.

데이터 센터 내의 네트워크 관리에 대해서는 OpenFlow 의 논리적으로 중앙집중된 구조와 프로그래머블한 네트워킹을 활용하여 기존의 분산처리로 인해 불필요하게 복잡했던 부분을 단순화 시키면서 효율성을 높이는 일이 진행되고 있다. 대표적인 구현으로 데이터 센터 내의 트래픽 상황에 따라 전원 관리 기능을 제공하는 ElasticTree [12]가 있다. 그림 5 는 Elastic Tree 의 시스템 다이어그램으로 데이터 센터 내부의 트래픽 상황에 대한 모니터링을 기반으로 현재 상황을 처리하기에 적합한 최적의 토폴로지를 구하고, 이에 따라 네트워크 링크와 스위치와 같은 네트워크 구성 요소의 활성화 여부를 동적으로 제어한다. 또한, 이를 바탕으로 패킷들의 경로를 결정함으로써 네트워크 구성 요소의 동적인 비활성화가 네트워크 연결에 미치는 영향을 최소화하면서 에너지를 효율을 높인다.

다른 한편으로는 기존에 릴레이 노드와 같은 부가적인 미들박스가 처리했던 부분들을 네트워크 컨트롤러를 활용하여 제공함으로써 쉽게 네트워크 서비스들을 관리하고 보급할 수 있는 방법에 대해서도 연구가 진행되고 있다. 그 중 Phonenet [13]은 그룹 통신을 위해 필요한 기능들인 위한 세션관리 기능과 패킷 릴레이 기능을 모두 네트워크 컨트롤러를

통해 제공함으로써 추가적인 노드 없이 네트워크 컨트롤러와의 통신을 통해 그룹 통신이 가능함을 보였다.

3.3. OpenFlow 전용 언어

OpenFlow 가 네트워크 장비에 대해 Open API 를 통한 소프트웨어적 접근을 추구하는 만큼, 전통적으로 복잡하고 하위 레벨로 접근하여 에러도 많이 발생하는 네트워크 설정 문제를 보다 상위 레벨의 기능을 통해 편리하고 안정적으로 풀기 위한 Domain-Specific 언어도 OpenFlow 기술에 발맞추어 개발되고 있다. 현재까지 ONS 행사와 논문을 통해서 소개된 언어로는 Frenetic, NetCore, Nettle 등이 있다.

3.3.1. Frenetic

제 1 회 ONS 의 주요 세션에 소개된 언어로서, 프린스턴 대학과 코넬 대학이 공동으로 개발된 언어로서, 기존 NOX 나 Beacon 과 같은 OpenFlow 컨트롤러를 제어하는 코드를 보다 편리하고 모듈화 된 형태로 작성할 수 있다[5]. NOX 와 같은 OF 컨트롤러를 직접 제어하는 것을 어셈블리 프로그래밍에 비유한다면, Frenetic 은 C 나 Java 와 같은 고차원 기능을 제공함으로써 OF 프로그래밍의 유지보수 측면과 프로그래밍 편의성을 향상시키는데 기여한다. 주요 특징으로는 OpenFlow 네트워크를 구성하기 위한 절차를 기술하기 보다는, 원하는 네트워크 설정을 표현하는 방식으로서, 마치 데이터베이스 스키마를 정의하고 테이블을 다루듯, OF 스위치의 테이블이나 컨트롤러의 핸들러를 정의할 수 있다. 또한 이렇게 작성한 코드를 추후 재사용할 수 있도록 모듈로 구성할 수 있다. 이 외에도 비트 연산과 같은 하위 레벨 표현을 손쉽게 해주는 부가적인 기능도 제공한다. 언어 분류상 C 보다는 Datalog 와 같은 DB 용 쿼리 언어 계통으로 정의했으며, 현재는 NOX 의 확장 라이브러리 형태로 구현되어 있지만, 독립적인 컨트롤러와 Frenetic 런타임 형태로 개발 중이다. 주요 기능은 다음과 같다.

3.3.2. NetCore

Frenetic 이 가진 장점뿐만 아니라, 언어의 의미를 정형적으로 정의한 언어로서, Frenetic 의 주요 저자가 개발했다[6]. Frenetic 에는 없던 formal semantics 가 제공된다는 점에서 추후 두 언어가 하나로 통합될 것으로 예상되지만, 현재는 별도 언어로 유지되고 있다. 의미론을 엄격히 정의했기 때문에, 단순히 OpenFlow 스위치를 좀 더 편리하게 다루는 차원을 넘어, 다양한 정책을 표현하고 wildcard 룰과 exact-match 룰을 적절히 구성하여 코드를 생성하는 메커니즘을 개발했다. 저자에 의하면 NetCore 컴파일러와 런타임은 올해 말 또는 내년 초 오픈 소스로 공개될 예정이다.

3.3.3. Nettle

예일 대학에서 개발한 하스켈 Haskell 기반 언어로서[7], Frenetic 과 마찬가지로 FRP(Functional Reactive Programming) 연산자를 이용하여 설계됐지만, Nettle 이 NOX 와 같은 OF 컨트롤러 레벨의 기능을 포함한 반면, Frenetic 은 NOX 와 같은 컨트롤러 위에서 동작하도록 설계됐다. 또한 Nettle 은 Frenetic 과 같은 쿼리 언어의 기능은 제공하지 않고 있다.

4. ONF 를 중심으로 한 SDN 확산 및 향후 전망

4.1. ONF 소개 및 SDN 확산 계획

ONF(Open Networking Foundation)는 2011 년 도이치 텔레콤(Deutsche Telecom), 페이스북(Facebook), 구글(Google), 버라이즌(Verizon), 야후(Yahoo!)를 창립 멤버로, SDN 을 통한 네트워킹의 혁신을 추구하면서 OpenFlow 를 비롯한 관련 기술과 표준을 개발하고 공유하기 위해 설립된 비영리 기관 목적이다[3]. 따라서 지금까지 진행되던 OpenFlow 규격은 이제 ONF 에서 관리하게 되며 스탠포드 대학에서 운영하던 OpenFlow 사이트[4]와 통합 및 조정 작업이 진행중이다.

4.1.1. 회원 구성

현재 회원사는 설립 멤버(Foundings and Board Members)와 일반 회원으로 구분되어 있다. 설립 멤버는 앞서 언급한 초기 멤버와 Microsoft, 그리고 최근 가입한 NTT Communications 로 구성되고, 일반 회원사로는 Cisco 와 Juniper Networks 를 비롯한 기존 장비 업체와 HP, IBM, Intel, NEC, Nokia Siemens, Ericsson, Huawei 을 비롯한 주요 IT 업체와 Big Switch Networks, Nicira 와 같은 신생 벤처(이 두 회사는 OpenFlow 핵심 멤버가 창업한 회사임) 등과 같이 다양하게 구성되어 있으며, 우리나라에서는 ETRI 와 Samsung, KT 가 현재 가입한 상태다. 이처럼 네트워크 장비 사업자와 망 사업자 뿐만 아니라, 인터넷 서비스 업체와 기존 주요 IT 업체가 모두 포진한 상태로서, 커뮤니티 자체의 구성과 잠재력은 상당히 강력한 편이다. 특이한 점은 설립 멤버의 구성을 통해 알 수 있듯이, Cisco 와 같은 기존 장비 업체보다는 구글이나 페이스북과 같은 인터넷 서비스 업체나 기존 IT 솔루션 회사가 주도하고 있으며, 기존 장비 업체는 SDN/OpenFlow 가 점차 호응을 얻는 추세에 동참하는 차원의 성격이 없지 않다.

4.1.2. 성격 및 활동

ONF 는 표준화 기구로 설립됐지만, OpenFlow 와 같은 표준 규격 제정 활동에만 머무르지 않고, SDN 이라는 새로운 패러다임을 실현하기 위한 전반적인 아키텍처와 요소 기술을 개발하는 목적도 중요시하고 있다. 이러한 관점에서 OpenFlow 는, 지금까지 SDN 문맥에서 제시된 유일하면서 구현 가능한 표준 기술이며, SDN 을 구성하는 여러 가지 하부 기술 중

하나로서 강조되고 있지만, ONF의 탄생이 OpenFlow 프로젝트의 성공에 기반을 둔 만큼, 현재까지는 OpenFlow 표준 개발이 ONF 활동의 중심축에 있다. 올해 12월 5일자로 OpenFlow 1.2 버전이 완료된 상황이며, 1.4 버전까지 2012년도 중반에 완료할 계획을 갖고 있다. 회원사의 의지와 실행력을 바탕으로 올해부터 상당히 빠른 속도로 버전 업이 이루어지고 있으므로, 2.0까지 일정대로 무난히 진행될 것으로 예상된다.

4.1.3. 오픈 소스가 아닌 오픈 커뮤니티

ONF는 집단 지성과 생태계 중심의 성격이 강하여 종종 오픈 소스 단체로 오해하기 쉽지만, 어느 한 회사가 기술과 시장을 장악하지 않고 여러 회사가 서로 공동으로 협력하는 오픈 커뮤니티를 추구한다는 의미일 뿐, ONF의 활동으로 파생되는 IPR은 멤버십을 가진 회사간의 공유의 성격이 강하다. 이러한 멤버십은 연 3만달러로 유지되며, OpenFlow 프로토콜을 비롯한 SDN 관련 기술 및 라이선스에 비용을 포함한다고 볼 수 있다.

4.1.4. IETF의 SDN

ONF는 네트워킹 기술 및 표준 개발을 목적으로 한다는 점에서 IETF와 유사한 부분이 있다. 최근 IETF에서도 SDN BoF가 결성됐는데, 여기서 말하는 SDN은 Software Defined가 아닌 Driven으로서 소프트웨어 및 표준 인터페이스 중심의 접근 방법이란 점에서만 유사할 뿐, 실제 ONF와는 공식적인 연계가 없는 독립적인 활동으로 볼 수 있고, 오히려 현재는 약간의 긴장 관계가 엿보이기는 하나, 이는 향후 두 단체가 어떻게 합의하는가에 따라 달라질 수 있다. 현재 ONF는 SDN 및 OpenFlow의 초창기 멤버를 중심으로 결성되어 있으며, 개인적인 차원에서 ONF와 IETF 모두 활동하는 사람은 다수 존재한다. ONF의 주요 표준 문서는 IETF와 달리 저자 명시없이 공동체의 결과로 귀속된다.

4.1.5. 구성

ONF는 최상위 결정 기구인 Board of Directors와 전반적인 기술 자문을 담당하는 Technical Advisory Group(TAG), 세부 기술별 Working Group(WG)과 각 WG의장의 모임인 Council of Chairs, 그리고 ONF를 총괄하는 Executive Director(Dan Pitt)과 홍보 및 운영 조직으로 구성되어 있다. 다른 표준 단체와 마찬가지로 주요 기술적인 활동은 WG를 통해 이루어지고 있으며, 산업체 소속이 아닌 SDN/OpenFlow 초기 멤버는 TAG를 통해 ONF 활동의 전반적인 방향에 조언하고 있다. WG의장을 비롯한 주요 직책에 대한 결정은

OpenFlow 초기 멤버에 의해 결정됐지만, 이제 독립적인 ONF 가 설립된 이상 TAG 를 통한 조연 외에는 철저히 커뮤니티 중심으로 이루어질 예정이다.

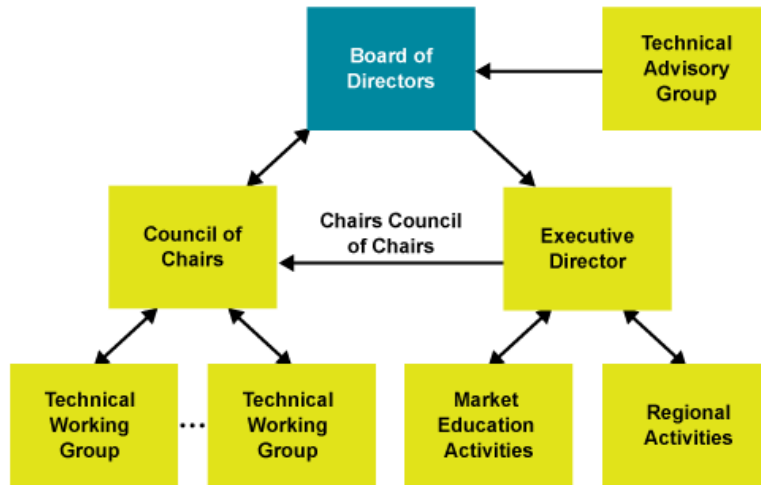


그림 6.ONF 구성[3].

현재 다음과 같은 WG 가 결성되어 있다. (각 WG 의장은 괄호 안에 명시) 현재 각 WG 은 오프라인 미팅보다는 메일링리스트와 간헐적인 컨퍼런스 콜 중심으로 이루어지고 있으며, 아시아측 참여자가 아직 적은 관계로 주로 미국 및 유럽 시간대에 진행되며, 주요 결과물이 나온 시점에 이뤄지는 오프라인 미팅도 샌프란시스코 및 산호세 지역에서 개최되고 있다.

- Extensibility (WG Chari: Jean Tourrihes, HP)
 - 모듈화, 인코딩, 프로토콜 독립성을 위한 핵심 요소를 개발하며, OpenFlow 규격 문서를 개발함
 - 가장 많은 관심을 받고 있고, 다른 WG 에 비해 활동이 적극적인 편임
- Configuration and Management (WG Chari: Deepak Bansal, Microsoft)
 - OpenFlow 스위치의 설정 및 관리 이슈를 다룸
- Testing and Interoperability (WG Chari: Michael Haugh, Ixia)
 - 호환성 검증 및 테스트 이슈를 다룸
- Hybrid (WG Chari: Bruce Davie, Cisco Systems)
 - 기존 장비와 OpenFlow 스위치의 혼용 환경에 대한 이슈를 토의

또한 공식 WG 형태는 아니지만 메일링 리스트 등으로 다음과 같은 여러 가지 주제에 대한 토의도 활발히 진행되고 있다. 메일링 리스트 및 기타 자료에 대한 접근은 회원사 로 제한되어 있다.

- OpenFlow Future (OF 2.0)
 - 현재 가장 활발히 활동하고 있으며, OpenFlow 2.0 의 핵심 구조와 이를 표현하기 위한 정형 언어를 논의하고 있음
- Match-Action Table
 - IPv4 나 IPv6 와 같은 특정 프로토콜을 처리하기 위한 방법을 논의
- Northbound API
 - 네트워크 제어 및 관리 애플리케이션 측면의 이슈를 다룸
- Use-case
 - OpenFlow 를 비롯한 여러 기술의 활용 사례를 논의

4.2. 향후 전망

ONF 는 SDN 의 확산 및 OpenFlow 규격을 개발하는 것을 주요 목표로 두고 있는 만큼, 정기적인 ONS 및 ONF 미팅을 통한 커뮤니티 교류와 주요 기술 및 표준 공동 개발을 좀 더 적극적으로 개발할 것이다. 실제로 제 1 회 ONS 및 ONF 미팅을 가진 이후로 OpenFlow 규격 개발 속도가 급격히 빨라졌으며, 최근 신설된 OpenFlow-Future 그룹을 통해 OpenFlow 2.0 에 대한 논의도 가속화되고 있다. Scott Shenker, Nick McKweon 을 비롯한 주요 SDN 창시자가 강조한 학제간 연결을 위해 ONF 의 TAG 에도 현재 공석인 분산시스템 전문가가 충원될 예정이며, 기존 CS 기술과의 시너지는 보다 극대화될 것으로 전망한다.

표준 측면에서는 현재로서는 IETF 의 SDN 활동과의 연계는 예측할 수 없으며, 당분간 ONF 회원사 중심으로 주요 기술과 표준을 빠른 속도로 개발해나갈 것으로 예상된다. 이러한 표준 기술 규격과 더불어 NEC, HP, Cisco, Juniper, Big Switch Networks 등을 비롯한 회사를 통해 OF 스위치 장비가 빠르게 개발될 것으로 예상되며, OpenFlow 1.0 이후 상당히 복잡해진 구조로 인해 규격의 효율적인 하드웨어 구현 문제를 해결하고, 범용적인 추상 모델과 현실적인 구현의 간격을 채우기 위한 노력도 계속 진행될 것이다. 또한 기존 ONS 데모 및 기타 연구 활동을 통해 제시된 여러 가지 라우팅 기술과 오버레이 네트워크 기술, 그리고 IPv6 지원을 비롯한 기존 기술과의 연계가 구체적으로 진행될 것이다.

OpenFlow 가 미래 인터넷을 위한 Clean slate 방식을 추구한 반면, ONF 에서는 좀 더 현실적인 접근을 취하여, 기존 인터넷 기술보다 융통성있는 기술 혁신이 가능한 데이터센터나 클라우드 분야에 적용되는 사례가 늘어날 것으로 예상되며, 실제로 구글이나 페이스북을 비롯한 자체 데이터 센터 및 클라우드 구축 기술을 보유한 업체에서는 이미 상당한 실험을 진행하고 있다. 또한 GENI 프로젝트에서도 OpenFlow 기술을 주요 요소 기술로 삼고 있는 만큼, 테스트베드 차원의 OpenFlow 망 구축도 보다 고도화될 예정이며, 국내 테스트베드 또한 SDN/OpenFlow 연계에 대해 적극적인 행보를 보이고 있는 만큼, 내년에는 미래인터넷 테스트베드의 SDN/OpenFlow 지원이 보다 구체화될 것이다.

현재의 추세대로 SDN 에 대한 커뮤니티의 기대가 큰 만큼, 조만간 다양한 SDN 관련 기술과 솔루션, 그리고 구축 및 활용 사례도 점차 늘어날 것이며, 국내에도 ETRI, Samsung, KT 등이 최근 ONF 에 참여한 만큼 향후 국내 전문가의 ONF 활동도 기대해볼 수 있다.

5. 참고문헌

- [1] 이동훈, 박주원, 김종원, "실감미디어, 서비스 합성, 프로그래머블 인프라에 근간한 미래인터넷 서비스 프레임워크," 한국정보과학회 논문지, 제 28 권, 제 1 호, pp. 31-40, 2010.
- [2] J. Roberts, "The clean-slate approach to future Internet design: A survey of research initiatives," *Annals of Telecommunications*, vol. 64, no. 5-6, pp. 271-276, 2009.
- [3] Open Networking Foundation, <http://www.opennetworking.org/fa>
- [4] Open Networking Summit, <http://www.opennetsummit.org/>
- [5] Frenetic, <http://www.frenetic-lang.org/>
- [6] Christopher Monsanto, et al., A Compiler and Run-time System for Network Programming Languages, POPL'12, January 25-27, 2012, Philadelphia, PA, USA.
- [7] The Yale Haskell Group, <http://haskell.cs.yale.edu/>
- [8] OpenFlow Switch Specification Version 1.0.0., OpenFlow, Dec. 2009.
- [9] OpenFlow Switch Specification Version 1.1.0., OpenFlow, Feb. 2011.
- [10] N. Handigol, M. Flajslik, S. Seetharaman, R. Johari, and N. McKeown, "Aster*x: Load-Balancing as a Network Primitive," in *Proc. Architectural Concerns in Large Datacenters (ACLD '10)*, Jun. 2010.
- [11] K.-K. Yap, T. Huang, M. Kobayashi, M. Chan, R. Sherwood, G. Parulkar, N. McKeown, "Lossless handover with n-casting between WiFi-WiMAX on OpenRoads." in *Proc. Annual Int. Conf. on Mobile Computing and Networking (MobiCom '09)*, Sept. 2009
- [12] B. Heller, S. Seetharaman, P. Mahadevan., "ElasticTree: Saving energy in data center networks," in *Proc. USENIX Symp. on Networked Systems Design and Implementation (NSDI'10)*, Apr. 2010.

- [13] T. Huang, K.-K. Yap, B. Dodson, M. S. Lam, N. McKeown, "PhoneNet: A phone-to-phone network for group communication within an administrative domain," in *Proc. ACM SIGCOMM Workshop on Networking, Systems, and Applications on Mobile Handhelds (MobiHeld '10)*, Aug. 2010.