

SDN 의 use-case, 문제점, 그리고 보완책

정재용
아토리서치

요약

Software Defined Network(SDN)은 네트워크의 새로운 기술 트렌드로서 Cloud Computing, LTE 백본망등 다양한 분야에서 네트워크 리소스의 가상화 및 효율적 관제에 적용되고 있다. 본 논문에서는 SDN의 다양한 사용 예와 문제점, 그리고 이를 해결하기 위한 방안을 살펴본다.

I. 서론

Software Defined Network (SDN) 은 네트워크의 새로운 기술 트렌드이다. 기존의 blackbox 형태의 네트워크 장비들은 한 벤더가 하드웨어, 소프트웨어, 관리 툴을 모두 제공하며, 각 장비가 distribute 된 형태의 라우팅 알고리즘을 구동하여 네트워크 망을 구성하고 관제하였다. SDN 은 이러한 방식과는 확

연히 다른 방식으로, distribute 된 형태의 관제 방식이 아닌 중앙 집중된 형태로 망을 관리한다. 중앙에서 집중하여 정보를 모으고 관리를 함으로서, 다양한 최적화를 네트워크 상황에 대한 global view 를 가지고 할 수 있다. 이러한 중앙 관제의 핵심은 SDN Controller 로, 다양한 네트워크 서비스 소프트웨어를 구동하여, 최적화된 네트워크 운영이 가능하도록 한다. 또한 각 네트워크 장비는 더 이상 분산된 형태의 컨트롤 기능이 필요하지 않으므로, 그 구조가 훨씬 간단해지고, 가격도 낮아질 수 있어 전체 망 구성 비용의 감축도 가능하다. <그림 1>은 이와 같은 SDN 의 일반적인 구조를 보여주고 있다.

II. 본론

SDN 은 Cloud 상의 네트워크 자원 관리의 관점에서 혁신적인 변화를 불러오고 있다. Cloud 환경에서는 물리적인 장비들이 가상 머신 기술을 기반으로 CPU, 메모리, 스토리지, 네트워크 카드 등의 물리적인 자원들을 리소스 풀 화하여, 다수의 사용자의 요구사항을 보다 효율적으로 수용하기 좋도록 하고 있다. 문제는 기존의 일반적인 네트워크 망 구성에 비하여 Cloud 는 좁은 지역 내에 많은 장비가 복잡하게 구성되어 있으며, 통신의 주체가 되는 가상 머신들이 자유롭게 Cloud 환경에서 물리적인 위치를 바꿀 수 있고, 하나의 고정된 물리적인 망 구성 위에 서로 다른 사용자들의 가상 머신들이 서로 다른 가상 네트워크 토폴로지를 구성하며 가동된다는 것이다. 기존의 방식으로 이 문제를 해결하기 위해서는 Cloud 운영자가 새로운 사용자들이 들어올 때 마다 이들의 네트워크 구성 요구사항을 분석하고, 기존의 사용자와의 요구 사항 충돌여부를 판단한 후, heuristic 하게 새로운 네트워크 구성 방안을 찾아, 수동으로 (또는 약간의 관리 툴의 도움을 받아) 네트워크 구성을 변경한다. 이러한 방식은 많은 문제점을 가지는데, 1) 새로운 사용자가 끊임없이 들어오고 나가는 Cloud 환경에서는 사용자가 Cloud 서비스를 처음 사용할 수 있게 되기까지 오랜 시간이 걸

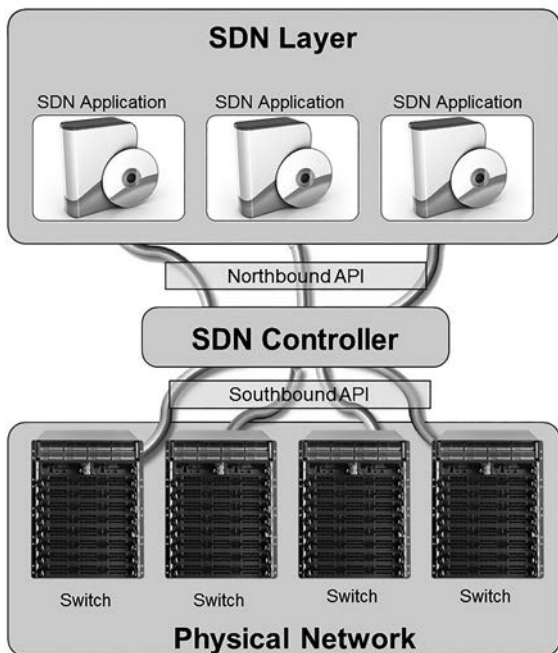


그림 1. SDN 의 일반적인 구조 (출처: chucksblog.emc.com)

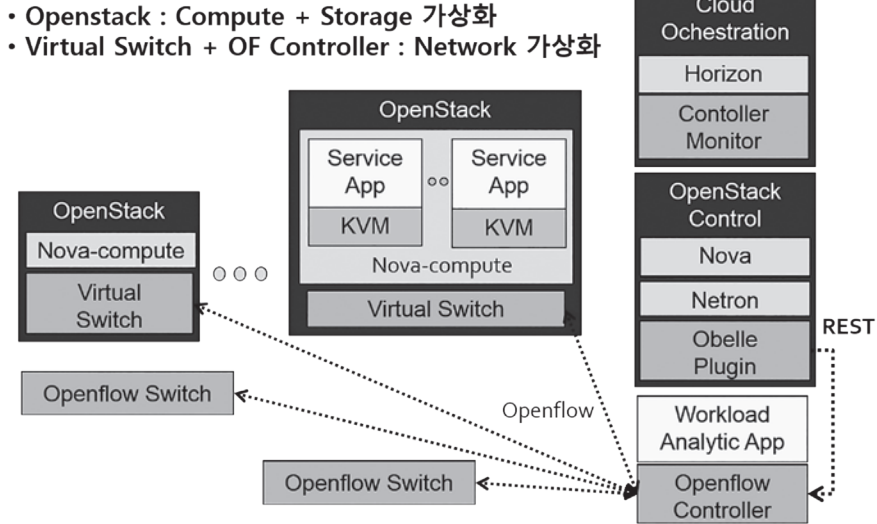


그림 2. OpenStack 기반의 Cloud 환경과 SDN 의 통합

리며, 2) 구성 후에도 설정된 네트워크 환경이 효율적으로 구동되고 있는지 확인하기도 어려울 뿐더러, 3) 초기 구성 이후 cloud 환경의 효율적인 사용을 위해 가상 머신이 그 물리적인 위치를 변동하고자 할 때 이에 맞추어 매번 네트워크 구성을 수동으로 바꾸어주는 것 거의 불가능에 가깝고, 4) 사용자가 특정한 네트워크 서비스 (예를 들어, 방화벽이나 로드 밸런서) 를 사용할 필요가 있을 경우 이것을 동적으로 추가해 주는 작업도 매우 어렵다.

SDN 은 물리적인 서버 박스들 내부의 네트워크와 Cloud 환경을 구성하는 스위치/라우터들을 중앙집중적으로 관리하여, 위와 같은 문제들을 해결하고 Cloud 네트워크 자원 사용의 효

율을 높인다. Cloud 환경에 SDN 이 적용되었을 때의 기본적인 구성 예는 <그림 3>와 같다. 그림에서는 open source 가상화 솔루션인 OpenStack 이 사용되어 물리적인 서버의 리소스 들을 가상화 하고 있다. Nova 를 통해 CPU 자원의 가상화하고, Swift 를 통해 storage 자원을 가상화하며, virtual switch 를 통하여 서버 박스 내의 가상 머신들이 서로간, 또는 외부와의 통신이 가능하도록 한다. 이 virtual switch 들은 SDN 기술을 지원하는 프로토콜 (예를 들어, Openflow 프로토콜)을 지원하여, 중앙집중식 SDN Controller 로부터의 관제를 받게 된다. 각 물리적인 서버를 연결하는 스위치와 라우터들도 Openflow 프로토콜과 같은 SDN 기능을 지원하여 SDN

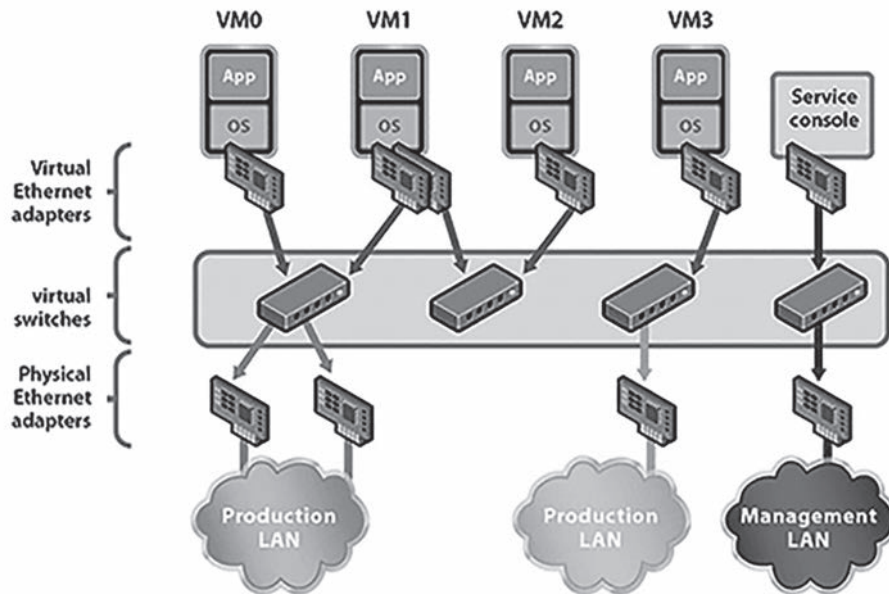


그림 3. 물리적인 서버 내에서의 가상 네트워크 구성 (출처: blogs.cisco.com)

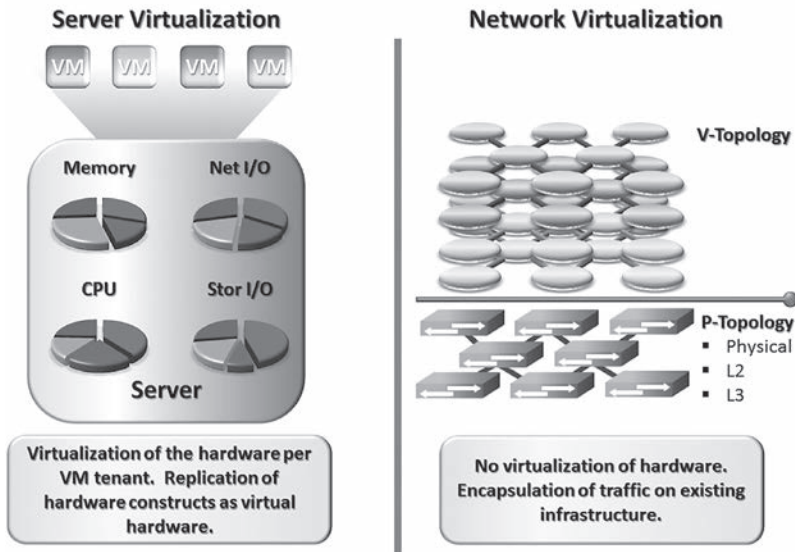


그림 4. Network slicing 을 통한 네트워크의 가상화 (출처: www.definethecloud.net)

Controller 가 서버 내부/외부 모두의 네트워크 상황을 모니터링하고 관제할 수 있도록 한다. Cloud 환경 전체의 관점에서 볼 때, SDN Controller 는 네트워크 자원을 관리하기 위한 backend 관제 시스템으로 동작하며, Cloud 환경 전체를 관리하는 orchestration 시스템 (OpenStack 에서는 Horizon) 에 통합되어 Cloud 관리자가 모든 Cloud 환경을 통합하여 관리할 수 있도록 한다.

〈그림 4〉는 이렇게 SDN 기반의 가상화가 이루어진 이후의 물리적인 서버 내부에서의 가상 머신들의 통신 형태를 보여준다. SDN Controller의 관점에서는 virtual switch 가 실제 스위치 또는 라우터로 보이므로, 가상 머신이 원하는 형태의 다양한 가상 토폴로지의 구성을 라우팅 테이블의 정보를 원격에서 세팅 함으로서 빠르게 지원하고, 동적인 변경도 용이하다. 또 가상 머신이 물리적인 머신 외부의 장비들과 통신하여야 할 때에는 이를 위한 low overhead 통신 채널도 virtual switch가 제공한다. 무엇보다 가상 머신들 간에 어떤 형태의 통신이 일어나고 있는지 모니터링이 가능하여, 일부 가상 머신이 다른 물리적인 서버로 이동하여야 할 때, 그러한 이동이 발생시킬 네트워크 상의 오버헤드 및 네트워크 자원 용량 요구량, 다른 가상 머신들과의 자원 충돌 여부들을 미리 판단하고 최적의 Cloud 환경 구성을 동적으로 지원할 수 있게 된다.

〈그림 5〉는 스위치와 라우터에 SDN 기술이 적용된 형태를 보여주는 예로, 기존의 OpenStack 과 같은 가상화 솔루션이 물리적인 서버내의 자원들만 가상화 시킨 반면, 스위치와 라우터에서도 하나의 물리적인 망을 여러 개의 가상 망들로 분리하여 관리할 수 있도록 해준다. 이를 위해 SDN Controller 는 마치 VLAN 이 동작하는 것과 유사한 형태로 각 스위치/라우터 장비

들의 라우팅 테이블을 원격에서 조절하여, 특정 가상 머신 그룹을 연결하는 가상의 별도 네트워크를 구성하고, 해당 네트워크 자원이 이 그룹에 의해서만 사용될 수 있도록 하며, 더 나아가 같은 물리적인 네트워크 라인을 공유하는 복수개의 가상 네트워크가 동작할 수 있도록 네트워크 사용량을 분배하는 기능도 해주게 된다. 또, 가상 머신이 물리적인 위치를 바꿀 경우에도, 이 가상 머신이 사용하는 가상 네트워크 토폴로지가 유지되도록 동적으로 스위치/라우터의 설정 변경을 해주는 작업도 한다.

지금까지는 주로 Cloud 환경에서의 네트워크 자원 가상화에 초점을 맞추어 설명하였으나, SDN 기술은 이보다 더 다양한 방법으로 Cloud 환경을 지원할 수 있다. 앞에서 언급되었던 것처럼 Cloud 환경에서는 방화벽과 같은 별도의 네트워크 서비스를 필요로 하는 사용자들도 있다. 이 경우 이러한 사용자의 가상 머신 및 네트워크 환경을 관제하는 일은 더 복잡해지게 되는데, 특정 가상 머신으로 유입되는 외부 입력은 반드시 방화벽을 거쳐야 한다는 트래픽 라우팅상의 추가 요건이 발생하기 때문이다. 만약 가상 머신이 자원 효율성 측면에서 여러 물리적 서버들 사이를 움직여 다녀야 한다면, 이에 맞추어 네트워크 설정 또한 방화벽을 거쳐 해당 가상 서버로 가도록 지속적으로 변경되어야 한다. 이러한 작업은 위에서 설명한 것처럼 수동으로 하기

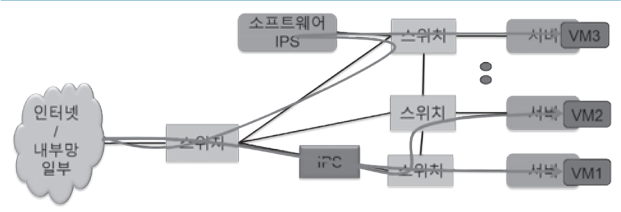


그림 5. CloudWatcher 동작 설명

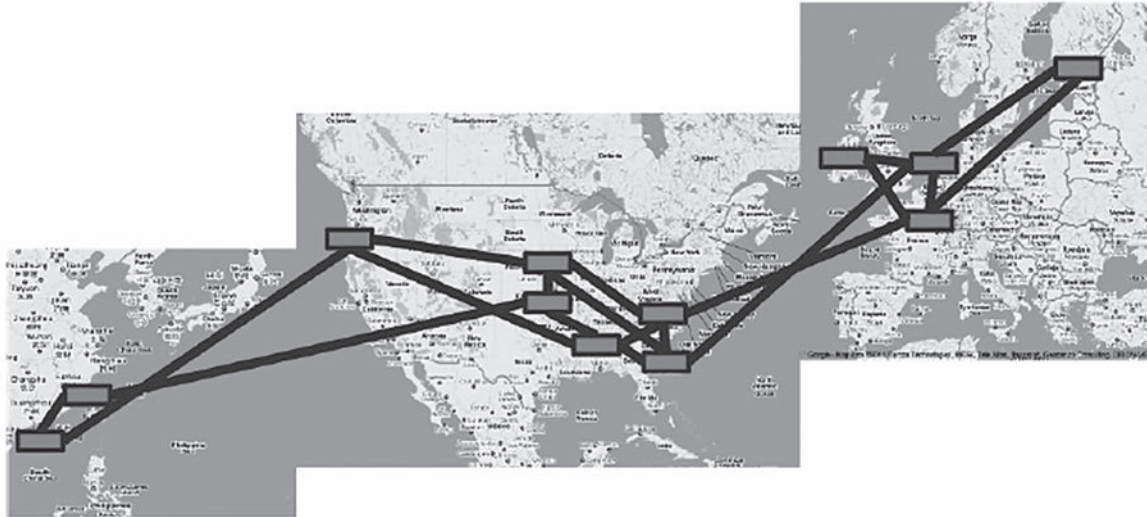


그림 6. 구글 SDN 적용 예(출처: B4: Experience with a Globally-Deployed Software Defined WAN, SIGCOMM'13)

에는 매우 어렵다. 또한 현재까지의 라우팅 알고리즘들은 통신 주체들 사이의 최단 경로를 찾기 위한 계산만을 할 뿐 라우팅상의 보안 요건 같은 것들은 고려하지 못하므로 문제 해결에 도움이 되지 못한다. 이 같은 문제를 해결하기 위한 기술의 한 예인 CloudWatcher 는 보안 요건을 고려한 라우팅 패스를 계산하는 새로운 라우팅 프로토콜로, SDN Controller 상에서 응용소프트웨어의 형태로 구동된다. <그림 6>는 CloudWatcher 의 기본 동작을 보여준다. 기본적으로 VM1 가상 머신은 하드웨어 보안 장비인 IPS (Intrusion Prevention System) 을 통하여 외부 트래픽을 받아 처리한다. VM2 가상 머신의 경우는 기존 라우팅 알고리즘이 계산한 최단 경로를 따라 트래픽을 받으면 IPS 장비를 거치지 않고 바로 외부 입력을 받아들여지게 돼 사용자의 보안 요건을 만족시킬 수 없게 된다. 이 경우 CloudWatcher 는 이러한 보안요건을 고려하여 IPS 장비를 거치는 경로를 산출하여 트래픽이 VM2 가상 머신으로 전달되도록 한다. 즉, VM2 가상 머신을 위한 최종 라우팅 패스는 IPS 를 거쳐 보안요건을 만족시키

는 라우팅 패스들 중 최단경로가 된다.

SDN 의 적용 범위는 하나의 데이터 센터 내부에만 국한되지 않는다. 구글이나 아마존 같은 세계적인 데이터 센터 기업들의 경우, 전세계에 다수의 데이터 센터를 구축하여 두고, 서비스를 제공하고 있어, 이 데이터 센터들 간의 대량의 통신이 필수적이다. <그림 6> 문제는 이러한 전세계를 아우르는 데이터 센터간 통신이 매우 비싼 통신비용을 발생시킬 뿐만 아니라 물리적인 거리상의 이유로 매우 비효율적으로 통신 채널이 사용되고 있다는 것이다. 이 문제를 해결하기 위해 구글에서는 SDN 을 데이터센터간 망에 적용하여, 그 효율을 크게 올렸다. 기본 아이디어는 각 데이터 센터가 전송해야 할 데이터가 있을 때, 먼저 중앙의 네트워크 트래픽을 관리하는 시스템에 사용 신청을 한다. 이 관리 시스템이 전세계 데이터 센터로부터의 네트워크 사용 요청들을 받아들여, 최적의 글로벌 네트워크 자원 사용 스케줄링을 한 후 그에 맞도록 SDN 컨트롤러를 통해 라우팅을 설정한다. 이를 통해 통상 30% 미만이던 네트워크 자원 사용효율을

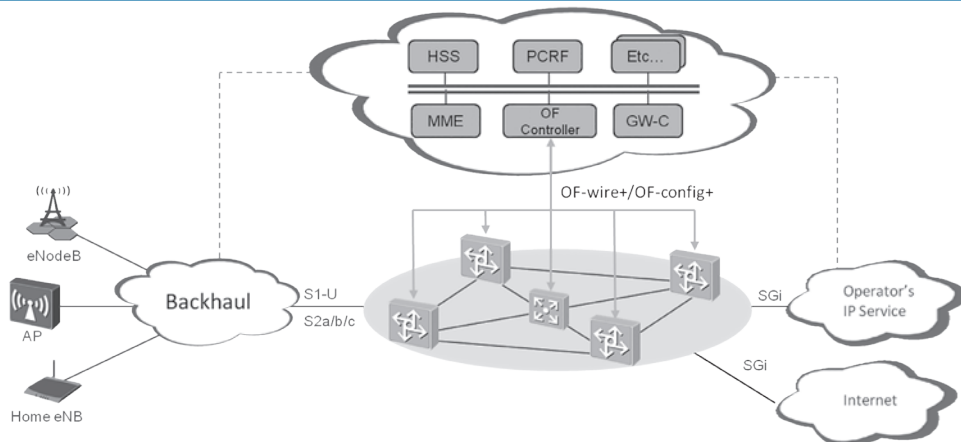


그림 7. SDN의 EPC적용예(출처: ONF Mobile WG)

거의 100% 가깝도록 올리는 성과를 거두었다

또 다른 SDN 적용 예로 private cloud 와 public cloud 를 연동하여, 평소의 트래픽은 private cloud 로 처리하고 사용량이 증가할 때만 public cloud 를 이용하여 처리하고자 할 경우 (예를 들어, 대학에서의 수강 신청 기간), private cloud 와 public cloud 로 물리적으로 분리된 네트워크라도 이를 하나의 가상 네트워크로 묶어주어 public cloud 가 private cloud 의 자연스러운 확장이 되도록 할 필요가 있는데, SDN 기술은 이러한 가상 네트워크 망 구성으로 매우 쉽게, 빠르게, 그리고 동적으로 해결할 수 있다.

SDN 는 데이터 센터 이외에도 다양한 응용분야를 가진다. 그 대표적인 예가 무선 통신망이다. (Figure 7) 스마트폰을 필두로 한 무선 통신 환경은 네트워크 산업을 이끄는 핵심 분야이다. 현재의 무선 네트워크 망은 주로 특화된 하드웨어 장비를 기반으로 고가로 구성되어 있어 날로 늘어나는 다양한 통신 요구사항을 만족시키기에는 구조적/비용적인 문제를 가지고 있으며, 다양한 서비스 구성을 동적으로 하기에도 어렵도록 되어 있다. 이에 Open Networking Foundation 같은 그룹에서는 무선 통신망을 보다 효율적으로 구축하고 관리하기 위한 기술을 개발 중에 있으며 이에 핵심이 되는 요소는 SDN 이다. 핵심 아이디어는 무선 이동 통신망을 구성하는 다양한 요소를 가상화된 환경에서 구축한 후 이들 요소에 대한 다양한 연결, 동적 확장 및 트래픽 스티어링을 SDN 을 적용 함으로서 비용/관리 효율성을 동시에 추구한다.

지금까지 SDN 기술의 다양한 use case 를 살펴보았다. SDN 은 분명 미래 네트워크를 선도할 기술이기는 하나 문제점 또한 가지고 있다. 첫째, 네트워크 라우팅 정보를 SDN 스위치가 컨트롤러로부터 받아와야 하므로, 라우팅 정보를 동적으로 세팅해 주어야 할 필요가 있는 패킷에 대해서는 성능 저하가 발생할 수 있다. 둘째, 중앙 집중 형태로 구동되는 컨트롤러가 다수의 스위치를 관리해야 하므로 라우팅 정보 요청량이 급격히 몰릴 때 성능상의 문제가 발생할 수 있다. 셋째, 컨트롤러 상에서 구동되는 네트워크 소프트웨어가 기존의 하드웨어 방식의 구현에 비해 성능상의 문제가 있을 수 있다. 넷째, SDN 을 통해 다양한 라우팅 룰을 스위치에 설정할 수 있으나, 라우팅 룰을 저장할 스위치상의 특수 메모리 (TCAM) 용량의 한계로 인하여, 일정 숫자 이상의 라우팅룰을 저장해야 할 경우, TCAM 을 소프트웨어 적으로 구현한 모듈에 의해 라우팅 룰이 처리되어야 하고, 이 경우 하드웨어 TCAM 기반의 처리에 비해 성능 저하가 발생할 수 있다. 이와 같은 문제점들을 보면 전반적으로 SDN 의 장점이 바로 단점이 되는 것을 볼 수 있다. 즉, SDN 이 제공하는 다양한 flexibility 가 성능상의 이슈를 발생시킬 여지를 만

들고 있는 것이다.

이러한 성능 상의 문제를 해결하기 위한 다양한 방법들 또한 도출되고 있다. 컨트롤러 상의 네트워크 처리 가속을 위해 DCA(Direct Cache Access), AVX (vector instruction) 등의 특수 CPU 기능들이 사용되며, Intel DPDK 과 같은 네트워크 소프트웨어 스택의 성능 최적화 또한 이루어지고 있다. 또한, NIC 상의 TCP/IP offload engine 하드웨어의 적용 및 RDMA (Remote DMA) 기능 적용을 통한 데이터 카피 횟수를 줄이는 등의 네트워크 카드 상의 성능 향상 기술 또한 적용되고 있다. 컨트롤러의 분산화 및 다 계층화를 통한 처리 능력 향상이 연구되고 있으며, 특수한 소프트웨어의 경우는 네트워크 프로세서가 포함된 PCIe 폼팩터 카드를 추가하여 성능 가속을 하기도 한다. 또한, SDN 에 최적화된 대용량의 TCAM 제품 또한 시장에 소개되고 있다. 이러한 다양한 해결책의 SDN 솔루션으로의 도입은 여전히 초기 단계이나, 그 연구 및 초기 상용화가 빠르게 이루어지고 있다.

III. 결론

지금까지 SDN 기술 적용 예 및 문제점 해결책들을 간단히 살펴본다. 다양한 기술적용사례 및 장점들로 인해 관련 장비 산업 또한 빠르게 발전하고 있으며, 정부의 지원도 크게 이루어지고 있고, 기술 수요 또한 국내외에서 지속적으로 증가하고 있다. 현재 전세계적으로 네트워크 관련 주요 기업인 CISCO, HP, Arista, Juniper 등이 모두 관련 기술 개발 및 사업 기회 확보에 적극 나서고 있으며, 국내에서도 파이오링크, 유비쿼스, 다산, 아토리서치, 쿨클라우드 등 다양한 네트워크 관련 중견/신생 기업들이 제품 개발을 진행하고 있다. 우리나라 네트워크 시장 보호 및 새로운 사업 기회 확보를 위해 국내 네트워크 기업들이 더욱더 적극적으로 나서 SDN 기술의 개발 및 확산에 나서야 할 것이다.

약 력



정재웅

1997년 한국과학기술원 공학사
1999년 한국과학기술원 공학석사
2008년 스탠포드 대학교 공학박사
2008년~2010년 AMD research center senior design engineer
2010년~2012년 Intel lab senior scientist
2012년~현재 아토리서치 대표이사
관심분야: software defined network, finance IT