

SDN의 개요

네트워크의 이해와 SDN의 정의
SDN 이해와 실습(Demo)

네트워크의 이해

• 네트워크 디바이스

전통적인 네트워크 디바이스

- OSI(Open System Interconnection Reference Model) - https://ko.wikipedia.org/wiki/OSI_%EB%AA%A8%ED%98%95
- 확인할 장비 : 허브 / 스위치 / 라우터
- 데이터 : 패킷 – 컴퓨터 네트워크가 전달하는 데이터의 형식화된 블록, 데이터의 단위

패킷의 전달

- 하드웨어 계층과 소프트웨어 계층
- L1 = 허브 : 받은 패킷을 연결되어 있는 모든 패킷에 전달
- L2 = 스위치 : 허브 + @ 연결된 디바이스에 대한 MAC 주소 목록
- L3 = 라우터 : 스위치 + @ 패킷에 포함된 IP 주소를 활용
- L4 / L7 스위치

네트워크의 이해

• 가상화에서 이야기하는 가상 네트워크

LAN 카드 또는 NIC

- Local Area Network 또는 Network Interface Card
- PC를 Network 에 연결하는 장치
- OSI 의 1계층 : 물리계층

가상화의 발전과 네트워크 속도 그리고 연결된 장치들

- 네트워크 속도의 변화



네트워크의 이해

• 네트워크 장비 – PC?

네트워크 장치(물리적 장치)

- 메인보드
- CPU / 메모리
- 랜 카드 / 연결 포트(시리얼 또는 콘솔 포트 등)

장치의 결론 – 쉽게 하나의 컴퓨터?

- CPU/DRAM/NVRAM/ Plash 메모리/ ROM
- 외부 연결을 위한 포트 : LAN / Serial Port / Console Port 등
- LAN : Ethernet 포트 – 네트워크 연결
- 운영체제 : 네트워크 운영체제

네트워크의 이해

• 네트워크 운영체제

하나의 PC 라고 생각하면 : 운영체제가 필요

- NOS : Network Operation System
- 네트워크 장치에서 사용되는 특화된 운영체제
- 패킷 처리 / 자료구조(Queue) / 분산 환경 / 보안

네트워크 운영체제

- 컴퓨터 운영체제와 같다?
- 컴퓨터와 장치를 랜이나 인터넷에 연결하는 특별한 기능
- OSI 모델의 상위 5계층은 많은 네트워크 운영체제를 기반으로 하는 토대를 제공
- 근거리 / 광역과 엔터프라이즈 네트워크 시스템의 App 의 형태를 갖추기도 함

네트워크 장치(장비)

• 네트워크 장치의 구조

물리적 장치

- 모니터 없는 물리적 하드웨어
- 랜 카드가 있는 컴퓨터?
- 통신 및 연결이 중점

네트워크 장치의 구조

- Plane : 평면
- Data Plane : 패킷의 송수신 담당
- Control Plane : 패킷의 경로 설정, 관리 및 제어 담당
- Management Plane : 동작 상태 및 성능 관리 기능 담당

네트워크 구조의 이해와 한계

• 네트워크 장치와 엔터프라이즈 환경

네트워크 디자인

- 대역폭
- 보안 및 관제
- 많은 장치들의 관리

디자인 시 참고

- Flat Network : 네트워크 홉의 최소화
- Fat Tree : 대규모 서버와 장비들의 접속 환경과 최적의 경로
- Fast Switching : 낮은 네트워크 대기 시간
- Simplicity : 구성과 운영에 대한 편의성과 용이성

네트워크 구조의 이해와 한계

• 네트워크 구조의 한계

운영 자동화와 중앙 관리

- 하나의 장비에서 3가지 기능 수행 : Data / Control / Management
- 구성 변경이나 확장 시 관리자들이 모든 네트워크 장치에 대한 변경 설정 및 정책 구성
- 만약 네트워크 장치가 여러 회사의 장치들이라면?

효율과 비용의 문제

- 네트워크의 중요한 기능 : STP(spanning Tree Protocol) / Routing Protocol
- 안정성을 위한 이중화 구성 : 이중화 시 루프 발생 가능성이 높음
- 네트워크 다운 가능성을 줄이기 위해 이중화 된 링크의 한 구간을 논리적으로 중단하여 루프방지
- 장비 복구 시간과 링크 한쪽을 논리적으로 끊는 것이기 때문에 비 효율적

네트워크의 복잡성 증가

- 네트워크 장비 중 라우팅 알고리즘은 보통 하나의 Control Plane에 네트워크 전체를 관리하는 중앙 집중 방식 후 분산 방식으로 변경
- 중앙 집중 방식 : 간편하고 효율적
- 장치를 만드는 회사마다 호환성 및 RFC 에 근거하여 장치를 만들지만 각각의 특징이 있거나 알고리즘이 조금 다른 경우가 발생하여 호환성 이슈가 발생

SDN이란?

• SDN 알아보기

SDN : 소프트웨어 정의 네트워크

- Software Defined Networking
- 네트워크 장비의 패킷 전달 기능에서 제어 부분을 물리적으로 분리하는 기술
- 네트워크 장비 내의 제어부(Control Plane)와 전송부(Data Plane)의 분리

SDN의 필요성

- 트래픽 패턴변화
- Server-Client 간 트래픽 패턴에서 Peer-to-Peer 형태의 트래픽 패턴
- 다양화 인프라 가상화 서버 가상화 뿐 아니라 네트워크 오버레이 기반 네트워크 분리할 필요성
- 병목현상/관리
- 복잡성 해소 : VLAN, QoS, ACL 등 트래픽 복잡성 증가에 따른 병목 현상 및 트래픽 관리 필요

SDN이란?

• SDN 알아보기

OpenFlow

- 네트워크를 경유하는 네트워크 스위치나 라우터의 포워딩 Plane에 권한을 제공하는 통신 프로토콜
- 네트워크 장치 업체가 OpenFlow 지원하지 않을 경우 SDN을 구현하기 어려움

ONF

- Open Network Foundation : 2011년 Microsoft , Google, Yahoo 등 8개의 회사를 중심으로 설립된 단체
- 구글, Goldman Sachs, 야후, 마이크로소프트, NTT, Facebook 등 세계적인 기업들이 가입되어 있고 활발하게 활동하는 중
- 표준화 작업을 지원

NFV

- Network Function Virtualization : 네트워크 기능들을 고집적 장비에 몰아넣어 비용 절감 및 효율성 극대화에 목적
 - 기존 네트워크 개념은 각 요소를 하드웨어 단위로 구분
 - 물리적 자원 최소화를 통한 전체의 효율성 향상과 시스템 복잡성을 감소시키는 것에 초점
- NFV가 반드시 SDN을 사용하는 것은 아니나 SDN에 적극적인 주요 벤더들이 NFV의 주요 벤더들이기 때문에 서로 상호 보완적인 관계가 보는 것을 권장

SDN의 이해

- Review ? 네트워크 장치

Data Plane : 데이터를 전송 (하드웨어 영역)

Control Plane : 운영체제 기능을 담당(소프트웨어 영역)

Management Plane : 네트워크 지능화 기능을 담당(소프트웨어)

SDN의 이해

- SDN과 기존 네트워크 장치

구분	SDN	장치 네트워크
관점	소프트웨어 중점	하드웨어 중점
주도권	User	네트워크 장치 회사
개방성	Open Source / 개발형	폐쇄적
호환성	표준 프로토콜로 호환성 높음	장치 회사들만의 독자 프로토콜 또는 표준 프로토콜
관리 효율성	합리적 / 효율적	고 비용 / 비효율
새로운 기술	User 및 신 기술 적용 가능	장치 회사들에 따라 다름

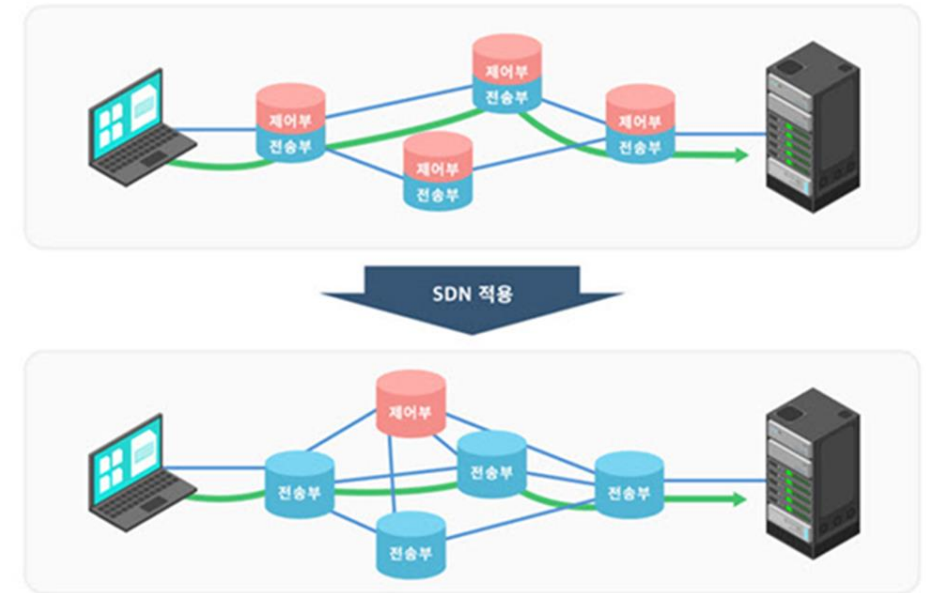
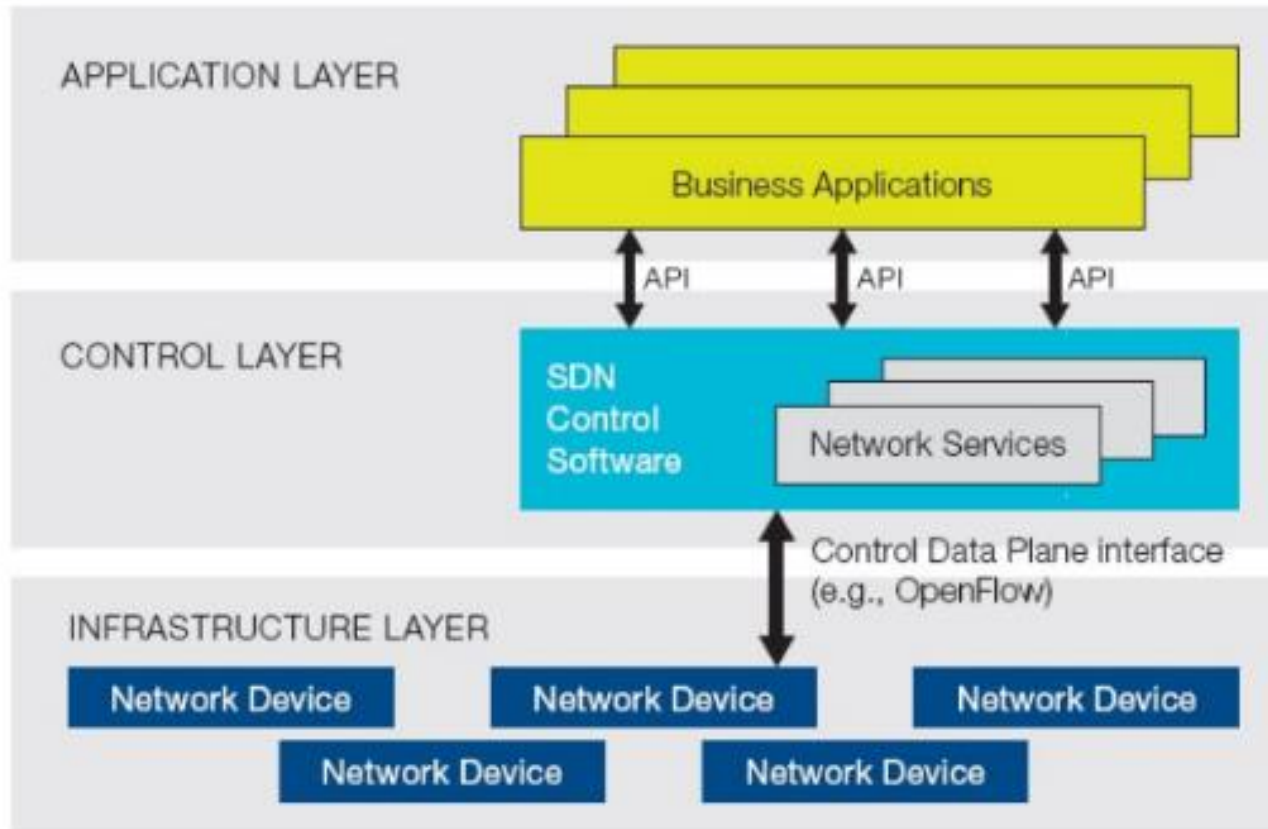
SDN의 이해

• 참고 사항 : SDN과 NFV 비교

구분	SDN	NFV
이유	네트워크의 중앙화 및 관리 네트워크 프로그래밍 Control / Data Plane 분리	특정장비 또는 회사에 종속 네트워크 환경에 대한 재 배치
적용 범위	학교 / 데이터센터	네트워크 서비스 회사
장비	상용서버와 스위치	상용서버와 스위치
앱	네트워크와 클라우드	라우터, 방화벽, 게이트웨이, WAN 및 CDN 등
Protocol	OpenFlow	x(현재 사용하고 있는 프로토콜 참고)
표준화	Open Networking Foundation	ETSI NFV Working Group

SDN의 이해

- 인터넷에 나와 있는 그림들



<출처: Netmanias>

SDN의 이해

SDN의 3가지 Layer

- Network Control Layer & Infrastructure Layer 의 Interface : Southbound API => OpenFlow
- Network Control Layer & Application Layer 의 Interface : Northbound API => 표준화 없음

SDN의 이해도 Up – 네트워크 장비

- 네트워크의 장비의 Data Plane => Infrastructure Layer
- 네트워크의 장비의 Control Plane => Network Layer
- 개발 Data Plane : Infrastructure 부르기 때문에 다양한 Data Plane 존재가 가능하지만 SDN에서는 논리적으로 한대의 네트워크 장비가 동작하는 것 처럼 운영되는 것이 목적
- Infrastructure 구축하기 위해 개별 스위치들은 Control Layer와 통일된 정보를 주고 받는 프로토콜이 필요 : OpenFlow가 많이 사용됨
- 개별 Data Plane마다 서로 다른 Flow Table
- OpenFlow는 표준 인터페이스로 주고 받는 Message에 대한 표준을 제시한 것
- Flow Table의 장비별 동기화를 하는 프로토콜이 아님

SDN의 이해

Flow Table의 구성

- Rule : 어떤 패킷을 처리할지를 정의하는 영역 – OSI 1 ~ 4까지 12개의 구분자를 가지고 처리
- Action : Rule에 정의된 패킷을 어떻게 처리할지 정의
- Stats : Flow Table에 얼마나 많은 Packet이 매칭되고, 어느 정도의 Byte가 전송되었는지 보여주며, 정보를 Controller 에게 전송

Controller

- OpenFlow 통하여 스위치 정보를 가져와서 네트워크 토폴로지(Topology)를 구성
- 스위치를 통하여 최적의 경로로 패킷이 전송되도록 해당 스위치에 Flow Table 를 전송하는 것 : ReActive 방식
- 패킷이 도착하지 전에 Controller에서 Flow Table을 스위치에 전송 : ProActive 방식
- 실제 이 2가지 방식을 혼용

SDN의 이해

Infrastructure Layer

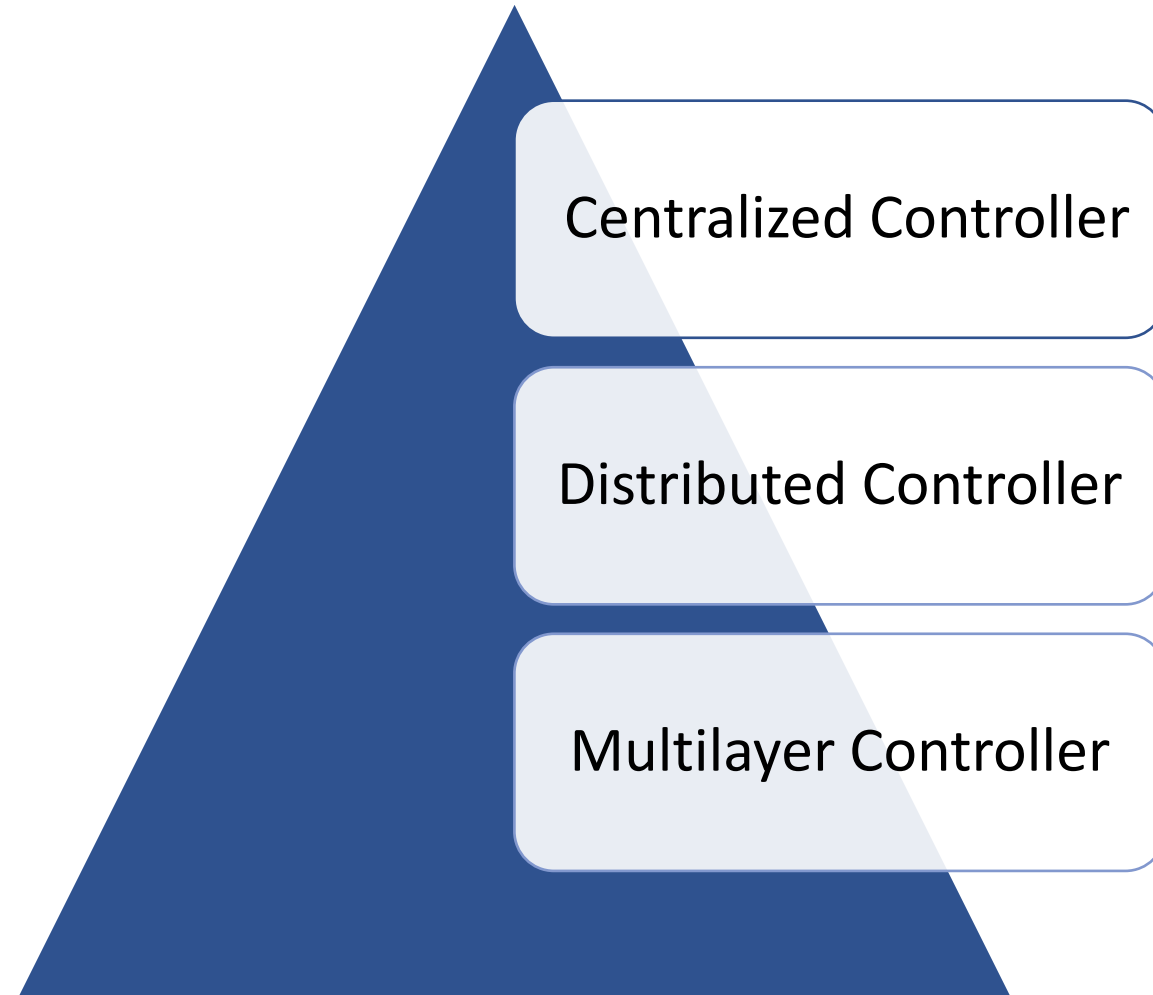
- 데이터 전송을 담당하는 영역인 Data Plane 영역을 다른 말로 Infrastructure
- 네트워크 장비들이 비싸지고 있는 원인 : 기존에 있던 Legacy 장비에 OpenFlow Agent 기능만을 올렸기 때문
- Infrastructure를 구현하기 위해서는 현재의 하드웨어를 OpenFlow에 최적화된 구조로 바꾸는 것이 필수

Controller Layer

- 기존의 Legacy와 SDN의 가장 먼저 보이는 차이점이 Control Plane의 위치
- 공통부와 Application의 구분
- 공통부 영역 : OpenFlow와 직접적으로 관련된 영역
- Controller에서 Topology Management, Path Management, Link Discovery 그리고 Flow Management 기능을 공통으로 제공
- Application 영역 : 공통부 영역의 공통 기능을 이용하여 네트워크 지능화를 하는 영역

SDN의 이해

Controller 디자인 방식



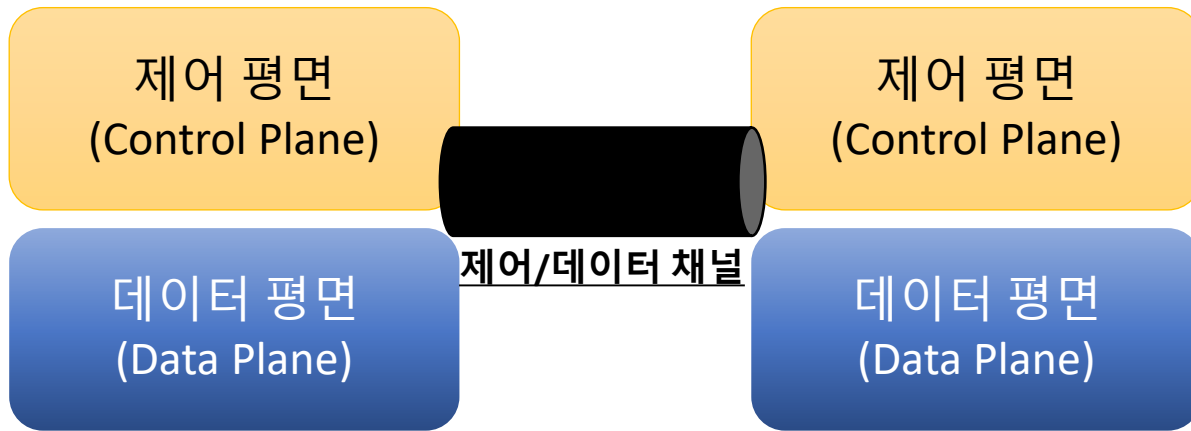
SDN의 이해

Application Layer

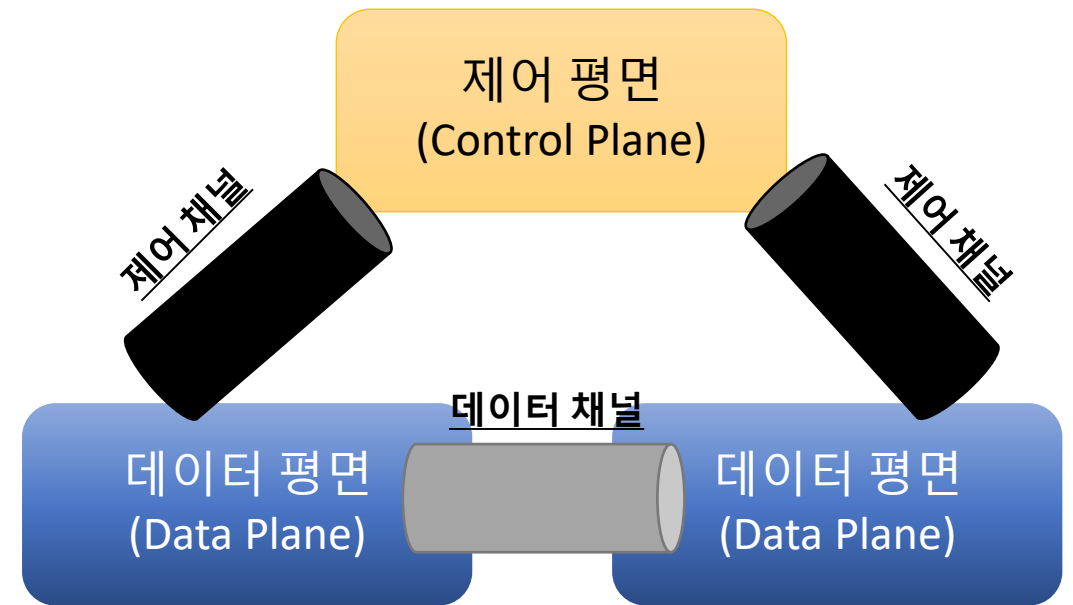
- Layer 4 ~ Layer 7에 이르는 서비스를 제공하는 솔루션
- 운영을 위해 다루는 툴들도 Application
- 모니터링을 하는 것도 Application
- Controller와 Infrastructure Layer를 제외하고 거의 대부분 Application이라고 해도 될 정도로 넓은 의미

SDN의 구조

- SDN의 구조



기존 네트워크에서의 제어와 데이터 평면 구조



SDN 네트워크에서의 제어와 데이터 평면 구조

SDN의 이해

• SDN Controller

SDN 컨트롤러는 SDN의 핵심이자 네트워크의 뇌에 해당하는 부분

- 여러 네트워크 장비와 통신할 수 있도록 API(South-bound API)를 제공 및 추가
- 여러 기능의 애플리케이션을 개발하고 다른 운영 도구와 통신할 수 있게 해주는 API(North-bound API)도 제공
- 상용 제품도 많이 개발되었으나 현재는 오픈 데이라이트(Open Daylight), ONOS(Open Network Operating System) 두 오픈 소스 컨트롤러가 주도
- 가장 많이 사용되고 있는 오픈 데이라이트는 2014년에 네트워크 장비 회사 시스코(Cisco)가 이끄는 오픈 커뮤니티에 의해서 개발이 시작
- 장비 회사가 개발을 주도한 덕에 오픈 데이라이트는 네트워크 장비와 궁합이 좋아 널리 퍼짐

실제 네트워크 장비 사용자인 서비스 프로바이더의 요구 사항을 적극적으로 반영하지 못한다는 한계점

- 한계점을 극복하고자 스탠퍼드와 버클리 대학교수 및 AT&T가 ONOS를 개발

SDN의 이해

• OpenFlow Protocol Message 종류

Controller-to-Switch

- Controller가 생성하여 Switch에 전달하는 Message로써 주로 Switch의 상태를 관리하거나 점검하기 위해 사용
- Features : OpenFlow Channel을 형성할 때 사용하며, 해당 Switch의 Capability를 확인하기 위해 사용
- Packet Out : Packet In Message를 통해 스위치로부터 수신한 Packet을 해당 스위치 상의 특정한 포트로 전송하기 위해 사용

Asynchronous

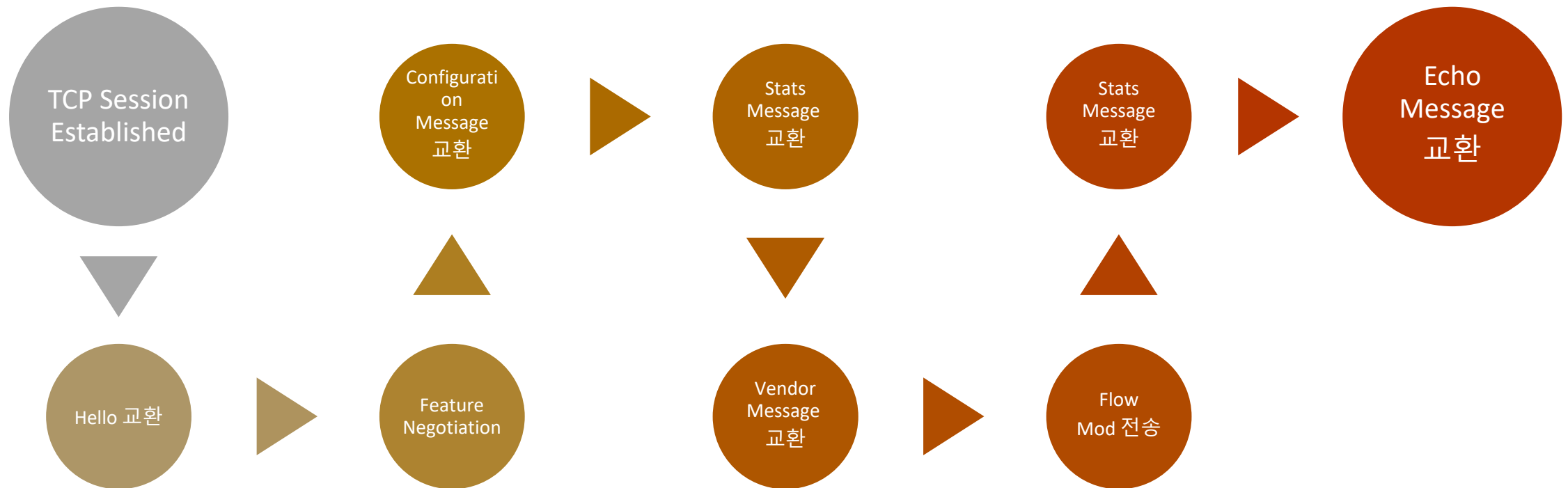
- Switch가 생성하는 Message로써, Switch의 상태 변경 및 Network Event를 Controller에서 업데이트하기 위해 사용
- Packet In : Switch가 Controller에게 Packet을 전송하여 Packet에 대한 Control을 받기 위해 사용

Symmetric

- Controller와 Switch에서 모두 생성되며, 상대방의 요청이 없어도 전송되는 특징
- Hello : Controller와 Switch 간에 연결을 시작할 때 사용
- Echo : Controller와 Switch 간 연결에 이상이 없음을 확인하기 위해 주로 사용

SDN의 이해

- Controller와 OpenFlow Switch간 Session 연결 순서



Demo – Virtual Box : SDN

수고하셨습니다.

질문사항 : sirmd@outlook.com