14주차 1차시 가상화와 SDN

【학습목표】

1. 가상화에 대해 플랫폼가상화와 리소스가상화로 나누어 설명할 수 있다.
2. SDN의 개념과 OpenFlow에 대해 설명할 수 있다.

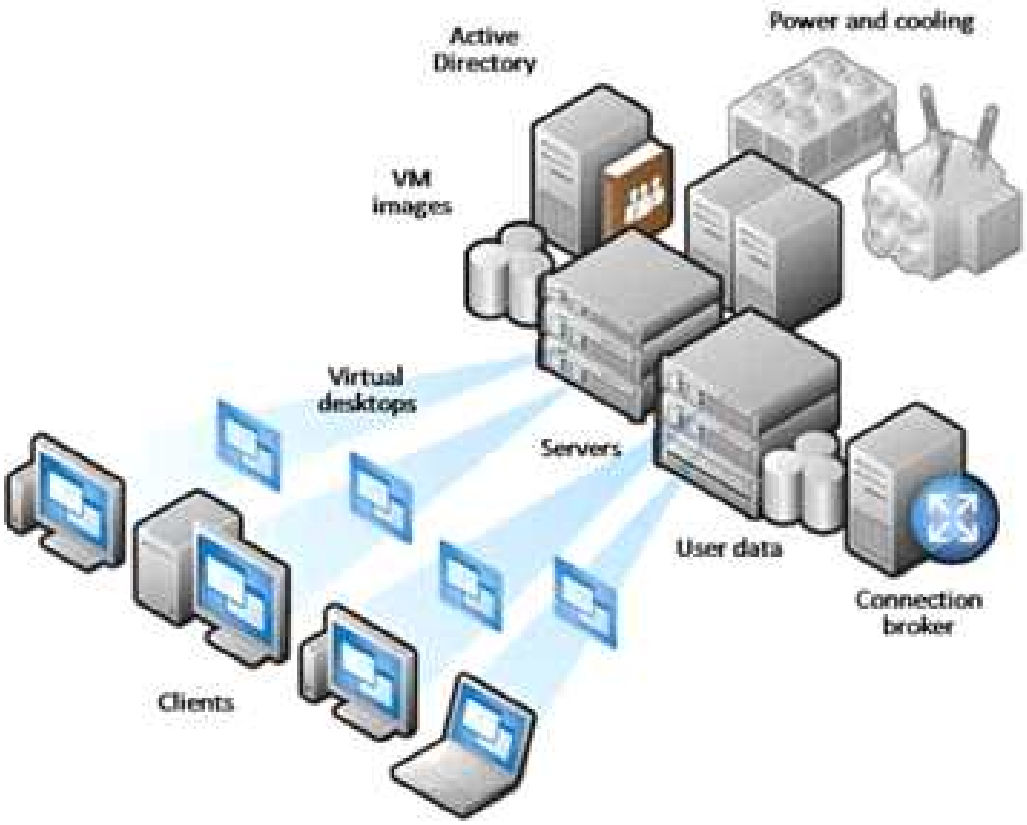
학습내용1 : 가상화

1. 개요

* 물리적인 컴퓨터 리소스의 특징을 다른 시스템, 응용 프로그램 및 최종 사용자들로부터 감추는 기술
  - 동일한 하드웨어에서 여러 다른 종류의 운영체제를 실행
  - 전산 자원들의 효용률(Utilization)을 높일 수가 있음

* 1960년대 이후에 주로 사용

*가상화
  - 플랫폼 가상화 : 모든 컴퓨터들을 시뮬레이트
  - 리소스 가상화 : 리소스들을 시뮬레이트

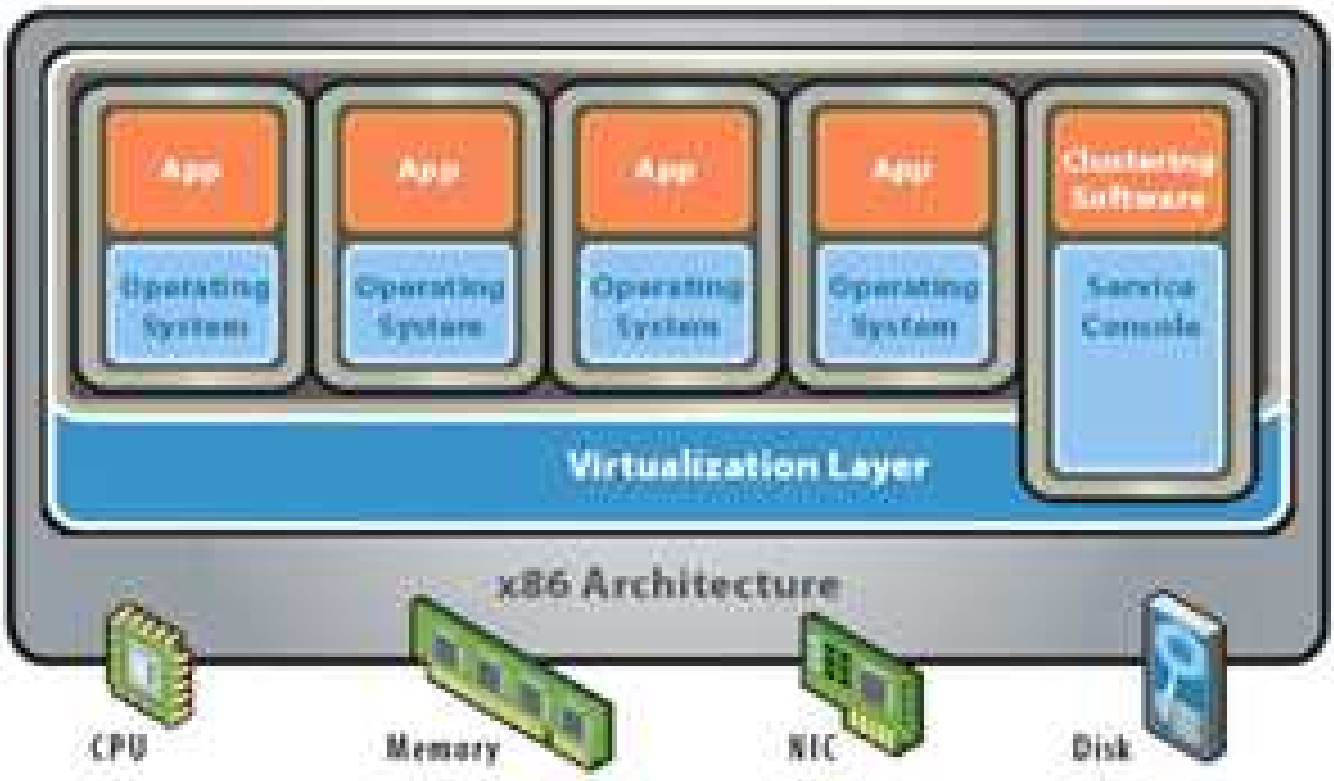


2. 플랫폼 가상화

* 가상화(Virtualization)란 단어의 어원은 1960년대에 하드웨어와 소프트웨어를 결합하는 가상 머신이라는 용어에서 유래

* 플랫폼 가상화의 기원은 IBM M44/44X 시스템에서 시작
- 가짜 컴퓨터(Pseudo Machine)를 구성한다고 하는 경우에 사용

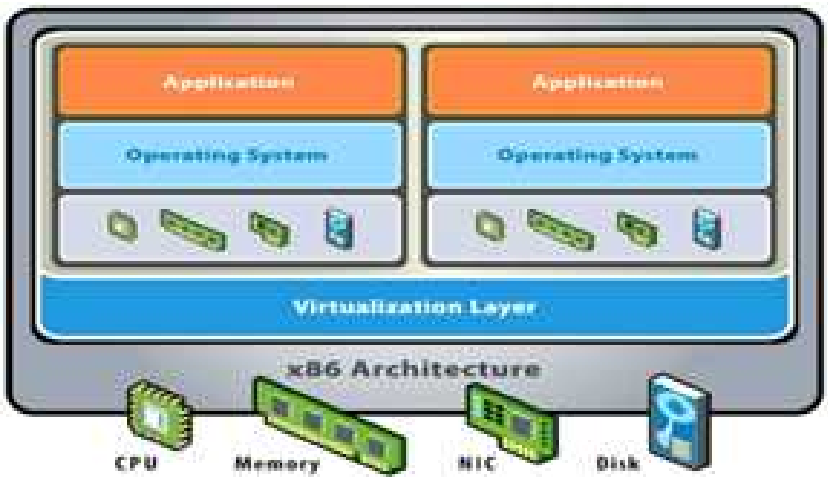
* 플랫폼 가상화
- 주어진 하드웨어 플랫폼 위에서 제어 프로그램, 곧 호스트 소프트웨어를 통해 실행
- 호스트 소프트웨어는 호스트 아래의 게스트 소프트웨어에 맞추어 “가상 머신”이라는 새로운 시뮬레이트된 환경을 생성
- 게스트 소프트웨어는 완전한 운영 체제를 말하며, 독립된 하드웨어 플랫폼에 설치된 것처럼 실행
- 가상 컴퓨터들은 하나의 단일 물리적인 컴퓨터 위에서 시뮬레이트되며, 수량은 호스트 하드웨어의 리소스에 제한을 받음
- 게스트 운영체제가 호스트 운영체제와 같을 필요는 없음



3. 리소스 가상화

* 플랫폼 가상화의 개념에서 저장 볼륨, 시스템 명, 네트워크 리소스 등과 같은 특정한 시스템 리소스들의 가상화로 개념의 확장

- 가상 메모리, 저장 장치 가상화, 가상 사설 네트워크(VPN) 및 가상 주소(NAT) 등



* 가상화(Virtualization)

[가상화 기술]

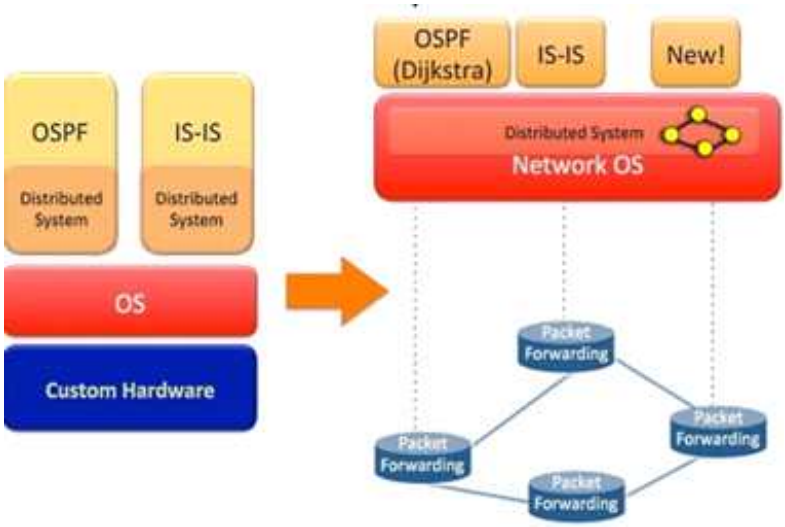
학습내용2 : SDN

1. SDN 등장 배경

- 기존 트리 구조 형태의 클라이언트-서버 중심 네트워크 환경에 모바일, 신규 컨텐츠 그리고 클라우드 기반 가상화 서비스 등의 변화 수용 요구

- 네트워크 구조와 관리 구조의 기술 한계 변화
  - 트래픽 패턴의 변화
  - 가상화 기술의 전개
  - 정체의 원인인 네트워크 구조의 복잡성
  - 네트워크 관리의 어려움
  - 벤더 의존성

- 네트워크의 데이터 평면(Data Plane)과 제어 평면(Control Plane)의 분리 추진



2. SDN 개요

* 소프트웨어로 네트워크를 제어하는 기술
- 1980년대 지능망(IN, Intelligent Network) 기술에 도입
- AT&T가 No.7 신호 교환기에서 이를 적용하여 신호교환기가 제어 평면 기능을 수행
- 90년대 ATM 기술에 적용

* IETF 2003년 forCES(forwarding and Control Element Separation) 워킹그룹 구성
- IP 포워딩, IntServ, DiffServ QoS를 제어대상으로 하는 RFC 3654 발표

* ITU-T 2008년 9월 SG13 워킹 그룹을 구성
- iSCP(independent Scalable Control Plane) 구조 제안
- 2006년 FIND(Future INternet Design) 프로그램의 일부로 스탠포드와 버클리 대학에서 SANE(clean-slate Security Architecture for Enterprise Network)과 Ethane 프로젝트 수행
- 이것이 OpenFlow로 이어짐

* 2011년 3월 OpenFlow 기술 상용화를 위한 ONF(Open Networking Foundation) 표준화 단체 구성

* SDN 기술로 확장되고, 표준화 추진

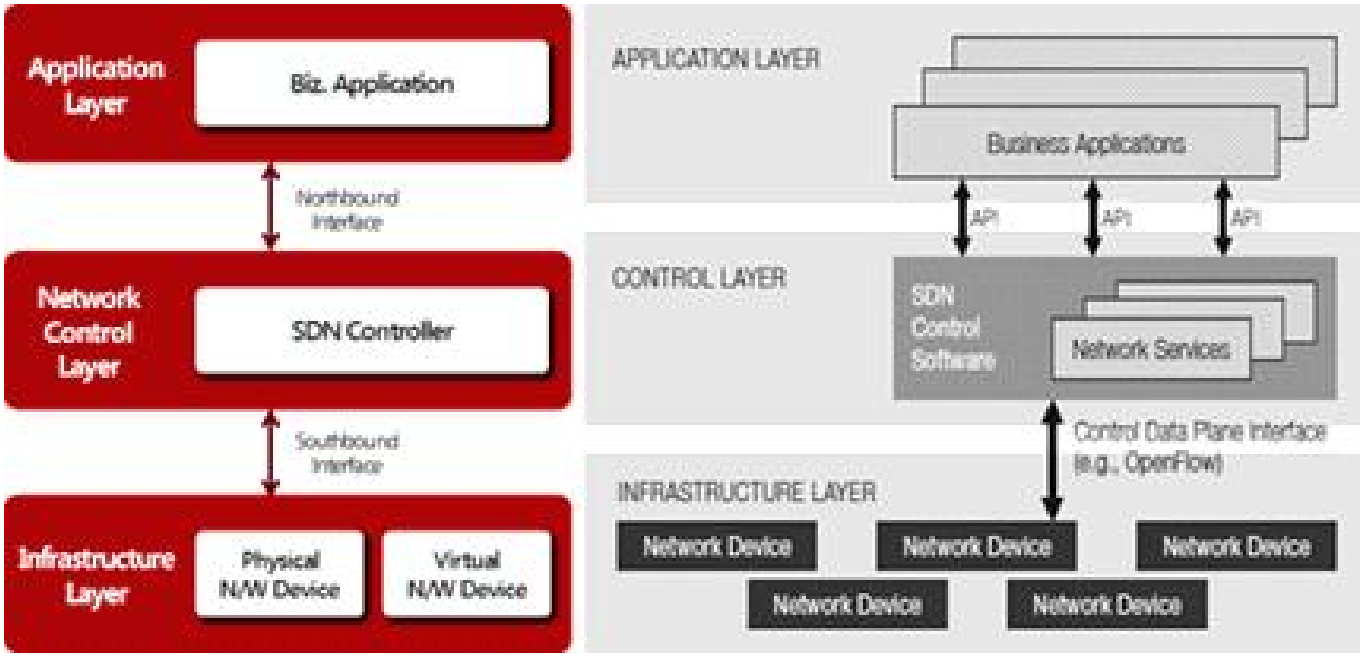
3. SDN 기술

* OpenFlow 인터페이스 기술 포함

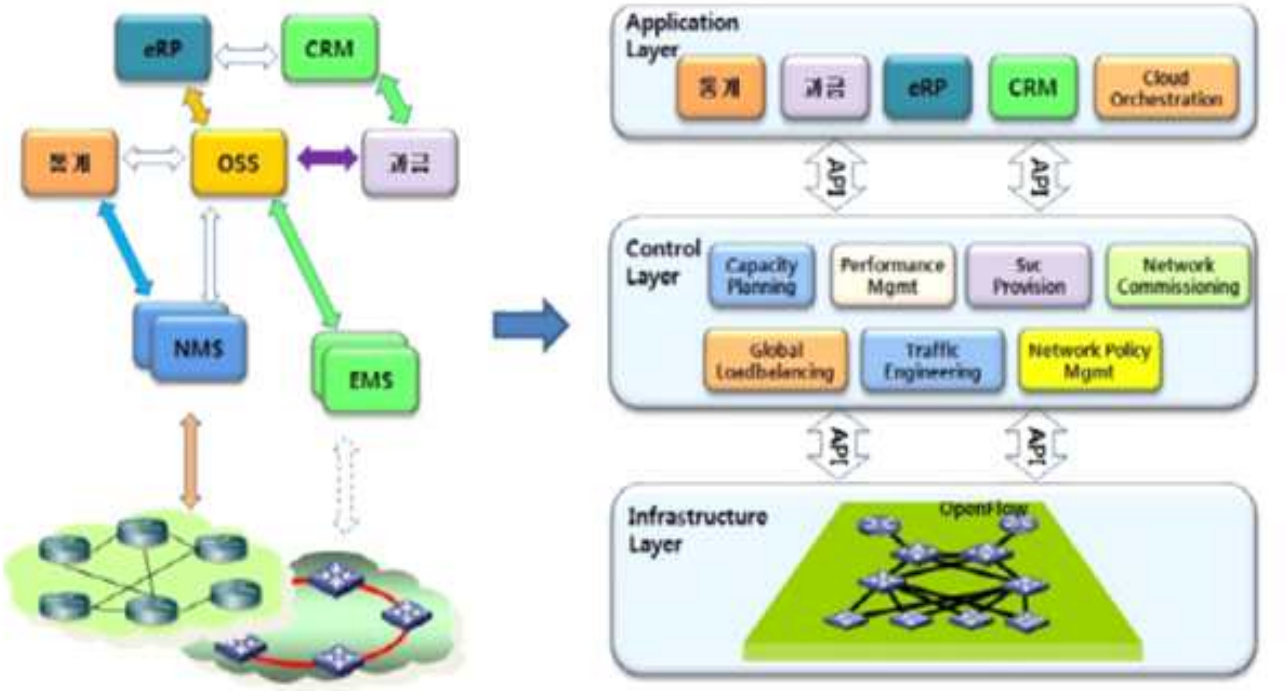
* 소프트웨어 정의 포워딩(Software Defined Forwarding)을 지원
- 기존의 장치에서 수행하던 데이터 포워딩 기능을 개방형 인터페이스와 소프트웨어를 통해서 제어

* 글로벌 관리 추상화(Global Management Abstraction)를 목표
- 전체 네트워크의 상태를 보면서 네트워크 요소의 제어 수행

* SDN의 기본 구조
- 네트워크 주요 기능이 SDN 컨트롤러에 집중화
- 컨트롤러로 전체를 관리하고 네트워크를 하나의 장비로 취급
- 이를 통해 벤더에 의존하지 않고 제어할 수 있는 네트워크를 설계 및 운영할 수 있게 단순화



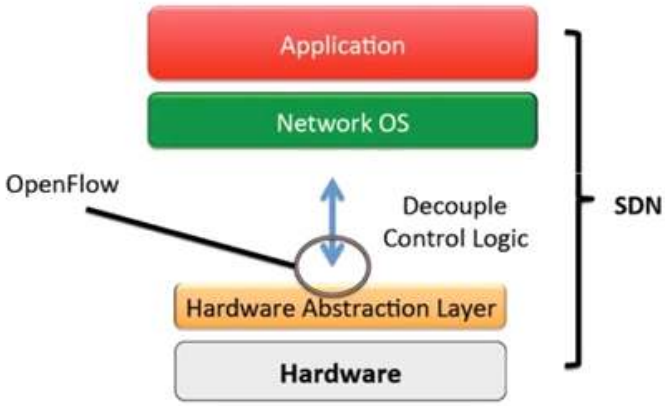
* SDN 관련 어플리케이션



4. OpenFlow

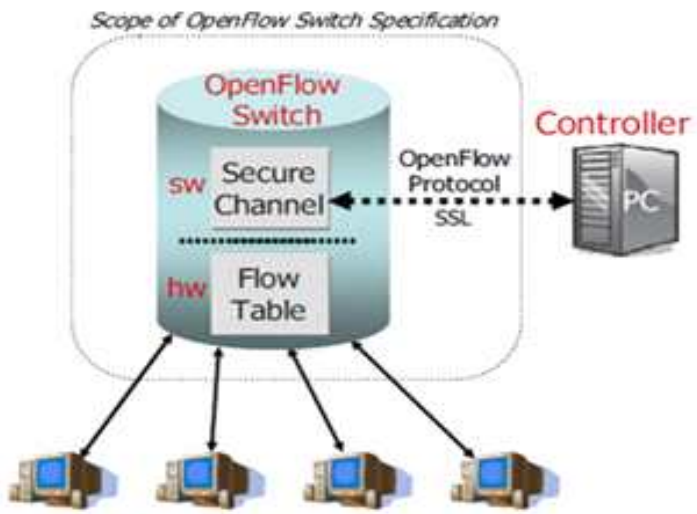
* SDN의 가장 중요한 기술
* SDN에서 컨트롤러와 네트워크 장비간의 인터페이스를 위한 규격으로 사용되는 기술

* OpenFlow 개요
  ① 제어 평면과 데이터 평면의 분리
  - 분리된 제어 평면과 데이터 평면을 연결하는 인터페이스 표준 기술
  - 기존의 하드웨어가 아닌 소프트웨어로 구현



Source: ONF Forum

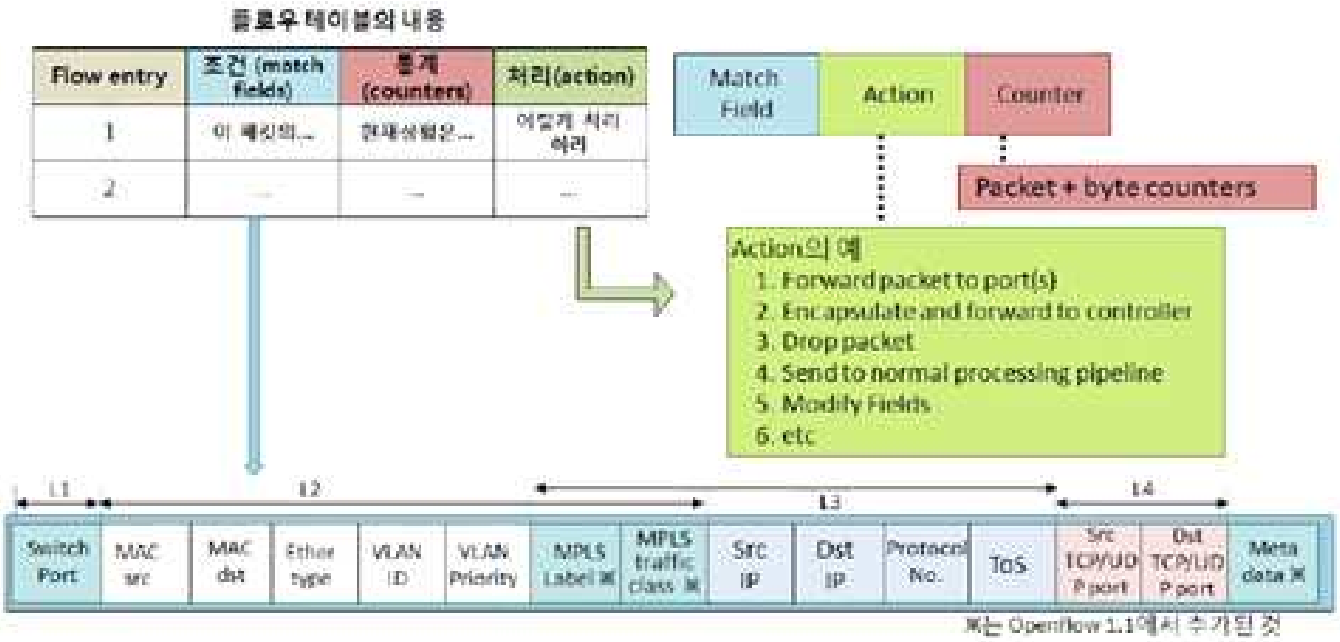
② 컨트롤러와 스위치로 구성
- 컨트롤러는 스위치에게 동작을 명하고, 스위치는 그 명에 따라 해당 동작을 수행
- 컨트롤러는 패킷의 포워딩 방법, VLAN 우선순위 값 등을 전송
- OpenFlow 컨트롤러의 주 역할은 경로 계산
  - 다양한 매개 변수들을 감안하여 경로를 결정
  - 이를 스위치에 전송하여 플로우 테이블에 저장
  - 패킷을 수신할 때마다 해당 플로우 테이블을 참조하여 해당 동작을 수행



- 다양한 매개 변수들을 감안하여 경로를 결정, 이를 OpenFlow 스위치에 전송하여 플로우 테이블에 저장되게하고, 스위치는 패킷을 수신할 때마다 해당 플로우 테이블을 참조하여 해당 동작을 수행

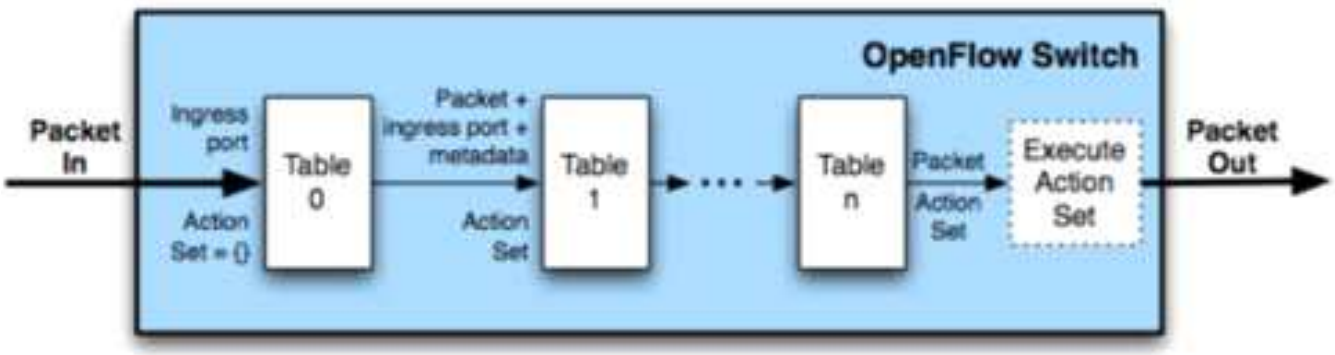
③ 플로우 테이블

- 스위치는 복수개의 플로우 테이블을 갖는다
- 조건(Match Field), 처리(Actions) 및 통계(Counters)라는 3개의 정보로 플로우 엔트리(Flow Entry)를 생성하여 테이블을 구성
  - 조건: 1 계층 스위치 포트 번호에서 4 계층의 TCP/UDP 포트 번호
  - 처리: 전송, 폐기, 지정된 필드의 값 다시 작성 등



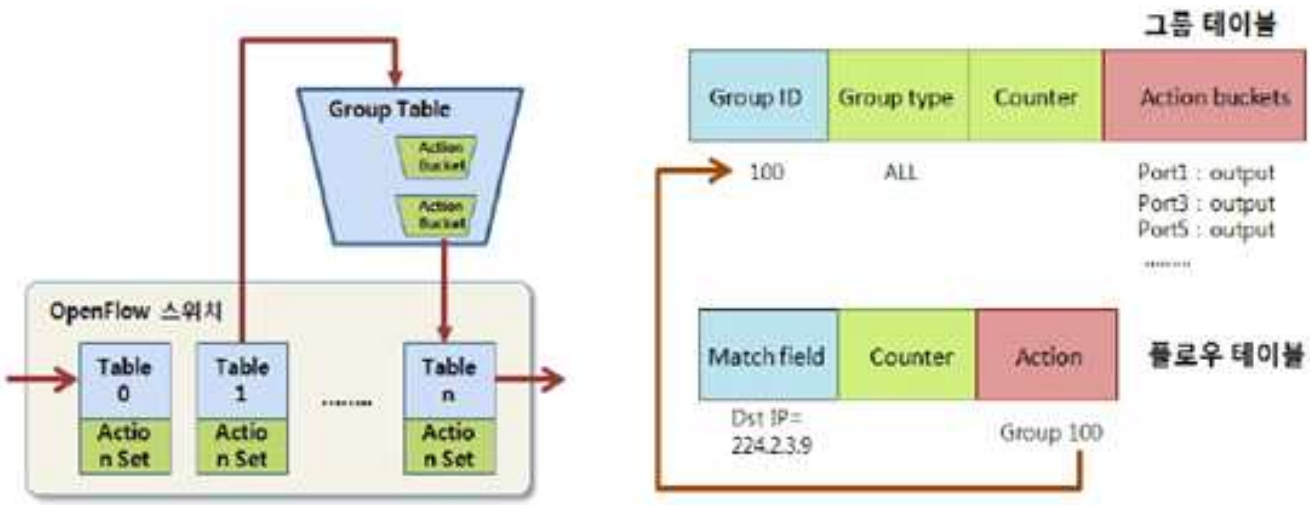
④ 파이프라인 처리 (Pipelining)

- 플로우 테이블은 번호 0 부터 번호를 부여
- 엔트리가 있으면 플로우 통계(Counter) 값 증가 후, 처리(Action) 실행



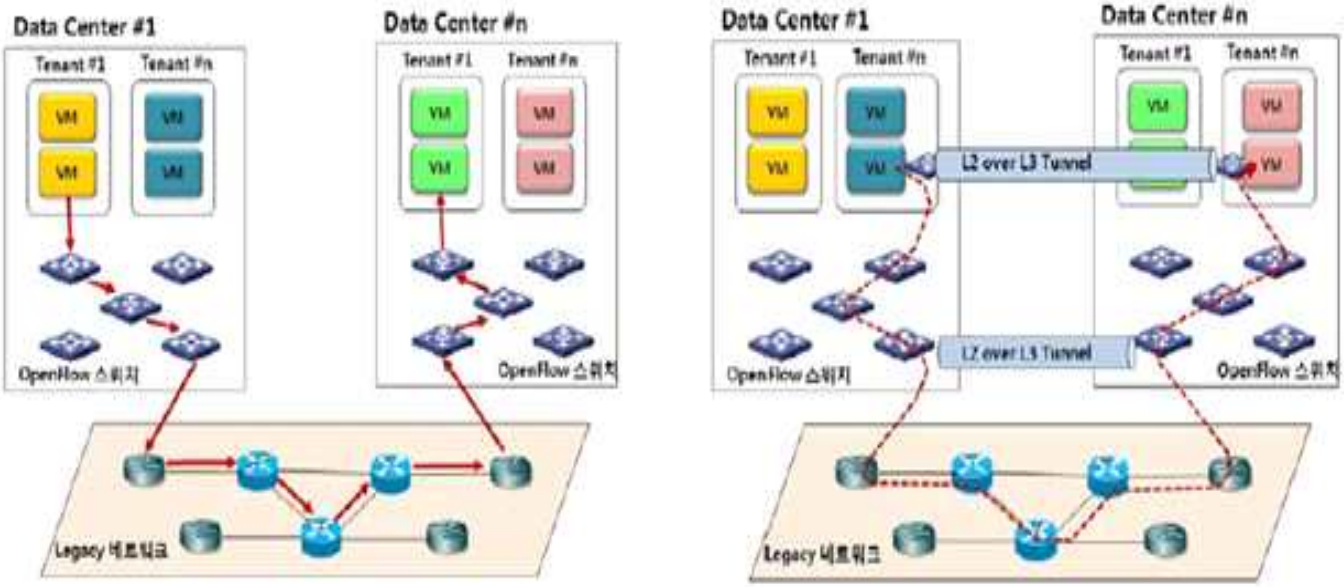
⑤ 그룹 테이블

- OpenFlow는 그룹 테이블 정의 가능
- 그룹이란 어떤 플로우에 Action Bucket들이 실행되도록 하는 것
  - Action Bucket이란 실행할 Action들의 집합 (파라미터 포함)
- 그룹은 그룹 테이블 내의 엔트리로 스위치에서 정의



* 라우팅 방식

- Hop-by-Hop 방식
  - 컨트롤러가 모든 스위치의 상태와 서비스 별로 해당 경로를 알고 있어야 하는 방식
- Overlay 방식
  - 컨트롤러가 모든 경로를 제어하지 않고 터널링 기술을 이용하여 해당 경로를 설정하는 방식



5. SDN 구조

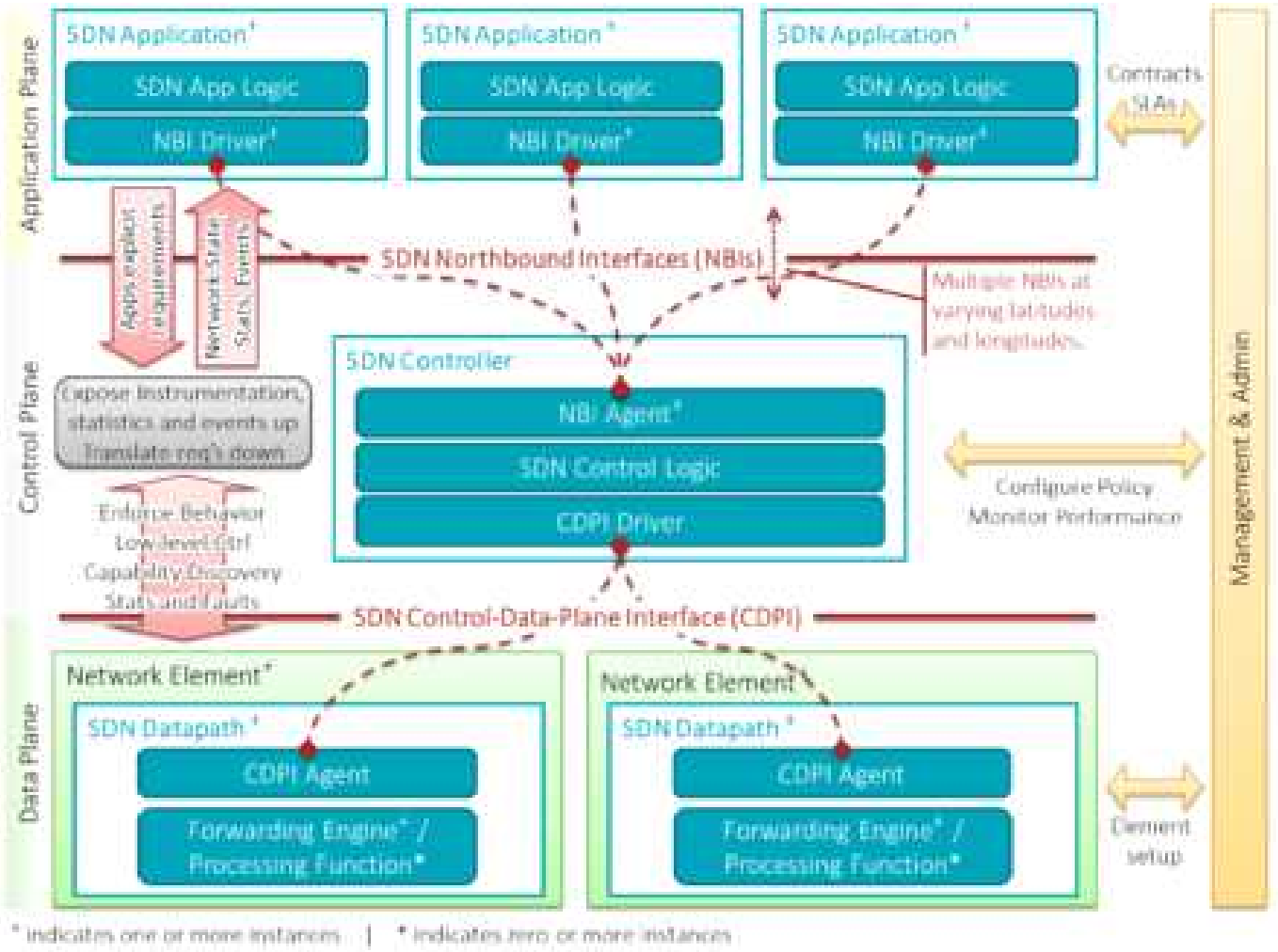
* SDN 구조
  - 데이터 평면 (Data Plane) 패킷을 단순히 전달하는 계층
  - 제어 평면 (Control Plane) 네트워크 구성 및 운영/제어를 총괄하는 계층
  - 응용 평면 (Application Plane) 다양한 네트워크 서비스를 위한 응용 계층

* Datapath는 제어-데이터 평면 인터페이스 에이전드를 통하여 제어 평면과 연결

* 응용 프로그램은 응용 평면에 상주

* NorthBound Interface(NBI) 드라이버를 통하여 관련 내용을 전달

* 중앙의 제어 평면은 명령을 번역하고 수행



* 컨트롤러 (Controller, 제어기)
  - 중심 기술로서 도메인내에 있는 네트워크 기기와 통신, 토폴로지 학습 및 프로그램 수행 영역
  - 소프트웨어 구동 플랫폼이며, 통신시 사용하는 게이트웨이
  - 컨트롤러 수행 통신
    • 내부 통신(Control Data Plane Interface 또는 SouthBound Interface) : 네트워크 기기 프로그래밍시 또는 기기들로부터 관련 데이터 수신시의 통신
    • 외부 통신(NorthBound Interface) : 어플리케이션과 컨트롤러 간의 통신

* 스위칭 (Switching)
  - 하드웨어 스위치와 소프트웨어 스위치로 분리
  - 소프트 스위치: 액세스 리스트, 속도 제한 및 트래픽 우선순위 부여를 위한 QoS 매개 변수 및 가상 포트에 적용된 전달 기능 수행
  - 하드웨어 스위치: 트래픽의 신속한 전송 수행
  - 이들 스위치들에게 전달 형태 프로그래밍을 위해 OpenFlow가 활용

* 오버레이 (Overlay)
  - 논리적 분리되어 있으나 하단의 실제 네트워크는 공유하는 가상 네트워크 그룹을 생성
  - VXLAN(Virtual eXtensible LAN)
    • 2 계층 프레임을 3 계층 UDP 패킷 내부에 캡슐화하는 방법
  - NVGRE(Network Virtualization with GRE) - 마이크로 소프트
    • 브로드캐스트, 언노운 유니캐스트 및 멀티캐스트 전송 시 멀티캐스트가 불 필요
  - NVO3(Network Virtualization Overlays) - IETF 워킹그룹
    • 3 계층 분리 없이 네트워크 어디에서나 가상 머신을 배포할 수 있는 방식

* 기존 vs. SDN 운영 방식 비교

【학습정리】

1. 플랫폼 가상화는 주어진 하드웨어 플랫폼 위에서 제어 프로그램, 곧 호스트 소프트웨어를 통해 실행되며, 리소스 가상화는 플랫폼 가상화의 개념에서 저장 볼륨, 시스템 명, 네트워크 리소스등과 같은 특정한 시스템 리소스들의 가상화로 개념이 확장된 것을 말한다.

2. OpenFlow는 SDN에서 컨트롤러와 네트워크 장비간의 인터페이스를 위한 규격으로 사용되는 기술을 말한다.