SDN Applications

James Won-Ki Hong, Jian Li, Seyeon Jeong

Dept. of Computer Science & Engineering

POSTECH

http://dpnm.postech.ac.kr/~jwkhong

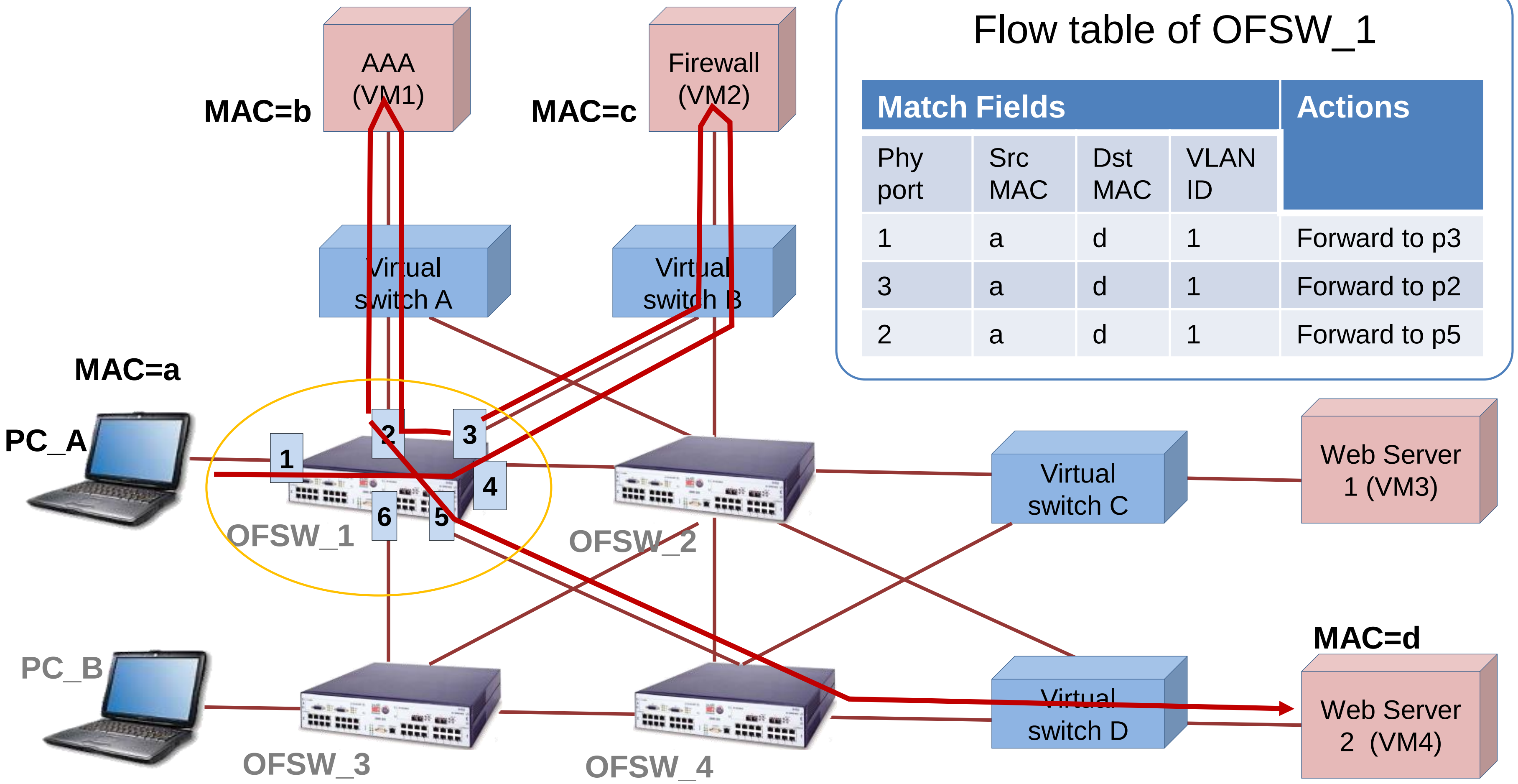
jwkhong@postech.ac.kr

Outline

- W3-1: SDN Applications
- W3-2: SDN Controllers – 1
- W3-3: SDN Controllers – 2

OpenFlow Example

- **Example of Routing Control (hop-by-hop routing)**
  - Service Chaining: Firewall → AAA → Web Server



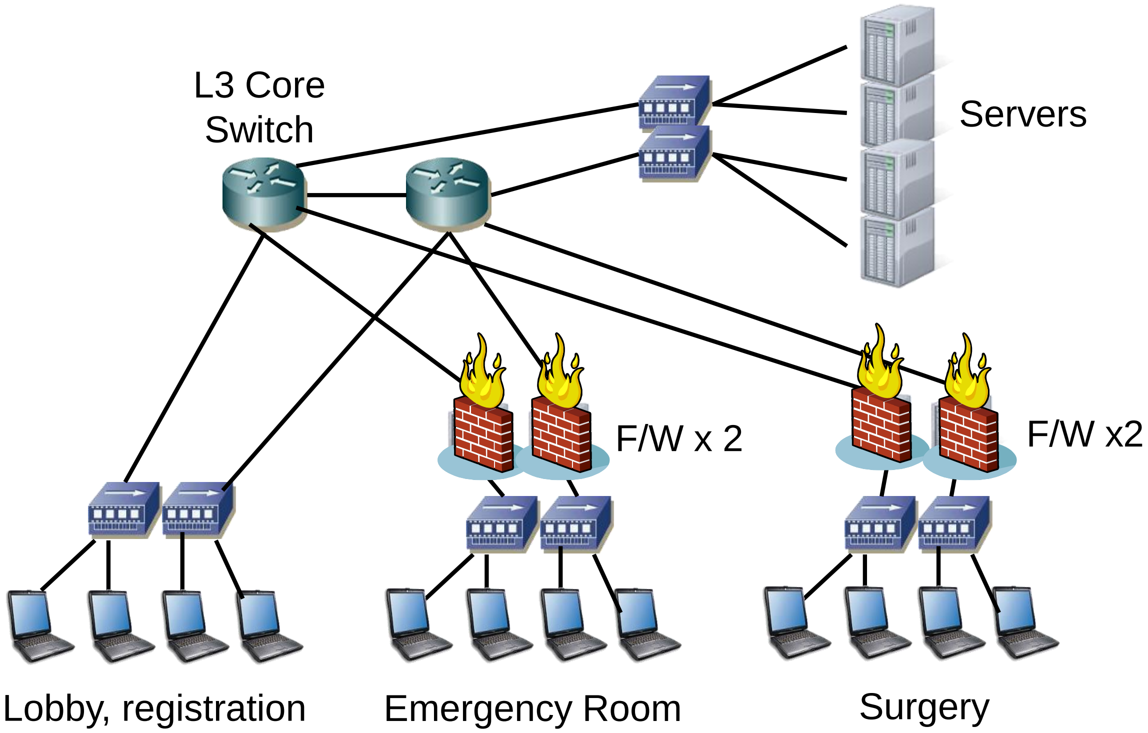
Flow table of OFSW_1

Match Fields				Actions
Phy port	Src MAC	Dst MAC	VLAN ID	
1	a	d	1	Forward to p3
3	a	d	1	Forward to p2
2	a	d	1	Forward to p5

Example: Kanazawa General Hospital (with NEC solution)

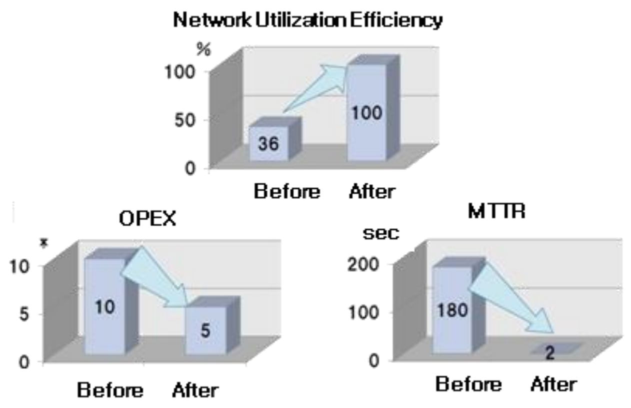
❖ Problem

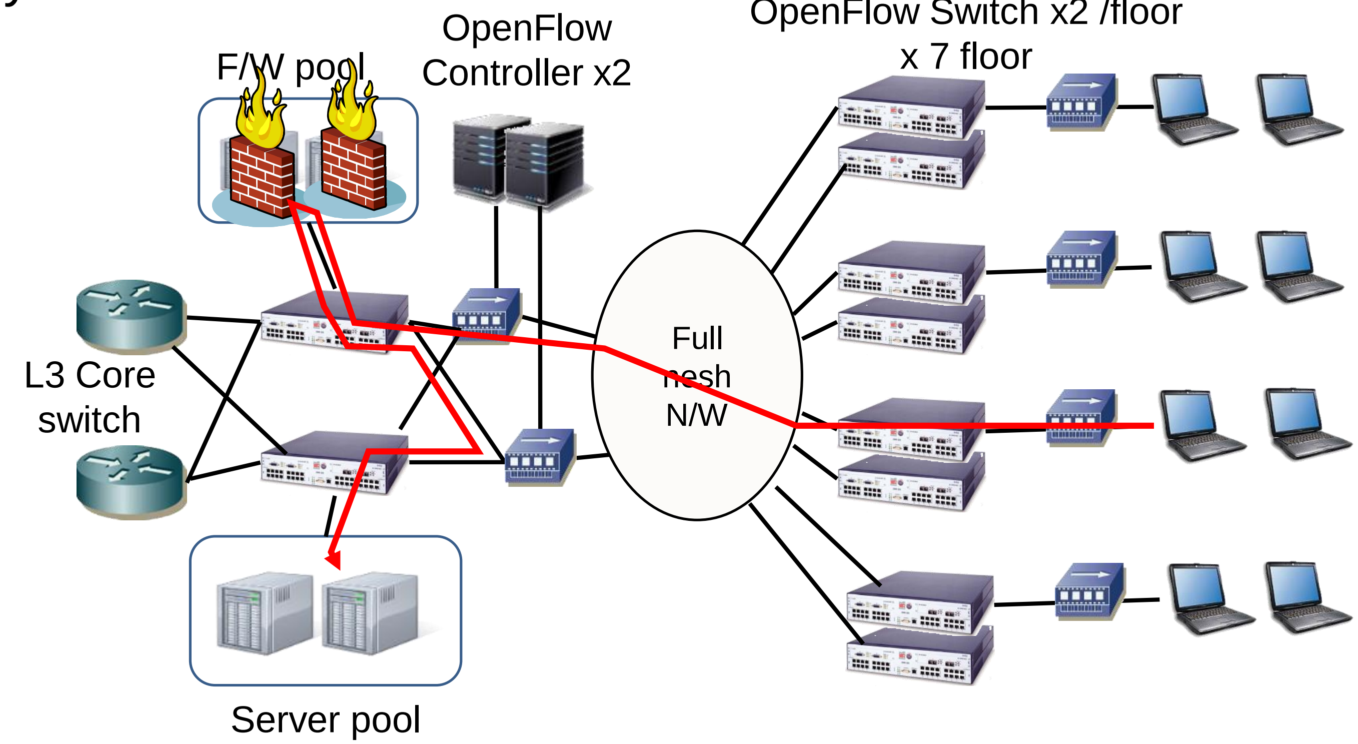
- Individual network optimization led to complex network structure
  - Configuration errors
  - Rewiring whenever a new equipment is connected
  - Difficult to find fault location



❖ Solution

- 16 OpenFlow switches and 2 controllers
- Create a virtual network/department
- Flow path control
  - Save CAPEX and OPEX
- Fast recovery from failure





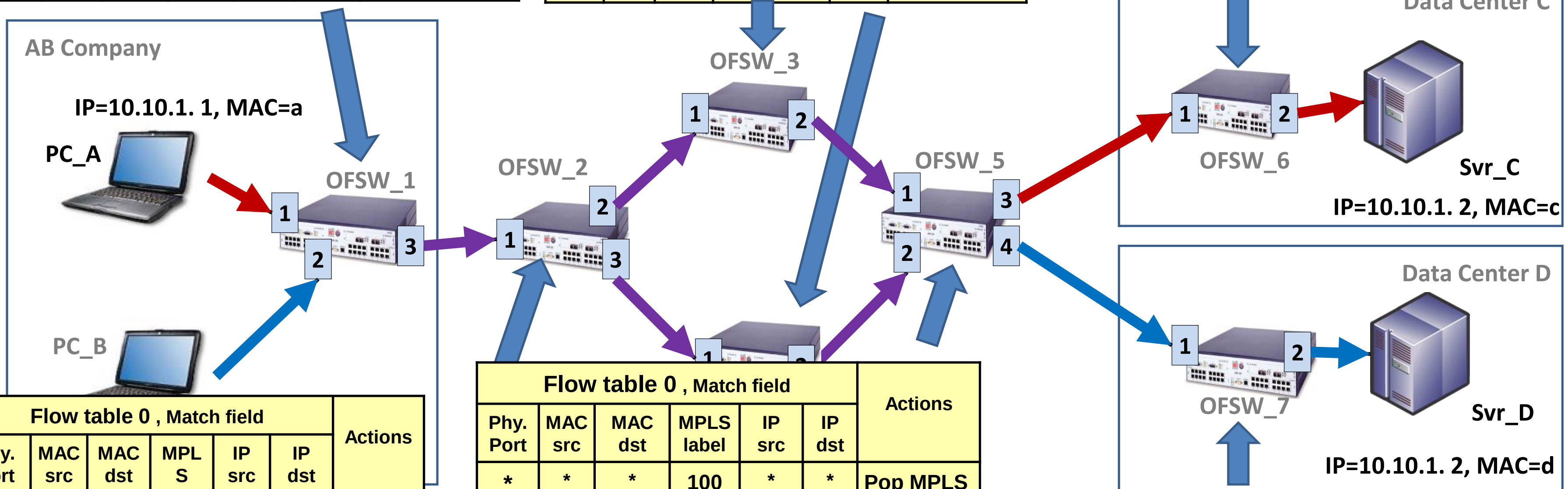
MPLS using OpenFlow

❖ Implementation of MPLS and Load Balancing

Match field						Actions
Phy. Port	MAC src	MAC dst	MPLS label	IP src	IP dst	
*	*	*	*	*	10.10.1.2	Push MPLS label 100 Output port3

Flow table 0 , Match field						Actions
Phy. Port	MAC src	MAC dst	MPLS label	IP src	IP dst	
*	*	*	100	*	*	Output Port 2

Flow table 0 , Match field						Actions
Phy. Port	MAC src	MAC dst	MPLS label	IP src	IP dst	
*	*	*	*	*	10.10.1.2	Set dst_MAC c Output Port 2



Flow table 0 , Match field						Actions
Phy. Port	MAC src	MAC dst	MPLS label	IP src	IP dst	
1	*	*	*	*	*	Group 100

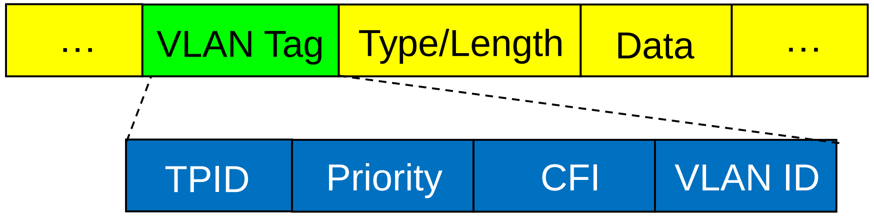
Group table			
Group ID	type	Counter	Action bucket
100	select	999	Output port2 Output port3

Flow table 0 , Match field						Actions
Phy. Port	MAC src	MAC dst	MPLS label	IP src	IP dst	
*	*	*	100	*	*	Pop MPLS label

Flow table 1, Match field						Actions
Phy. Port	MAC src	MAC dst	MPLS label	IP src	IP dst	
*	a	*	*	*	*	Output Port 3
*	b	*	*	*	*	Output Port4

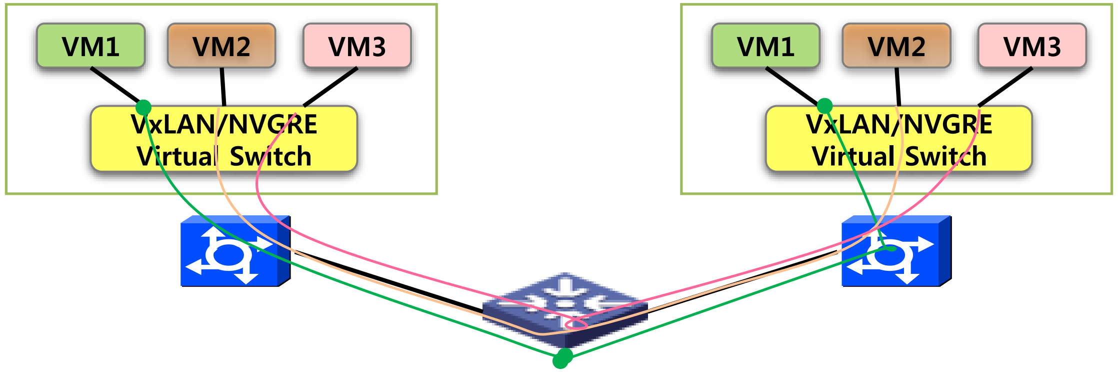
Flow table 0 , Match field						Actions
Phy. Port	MAC src	MAC dst	MPLS label	IP src	IP dst	
*	*	*	*	*	10.10.1.2	Set dst_MAC d Output Port 2

VLAN using OpenFlow



❖ VLAN

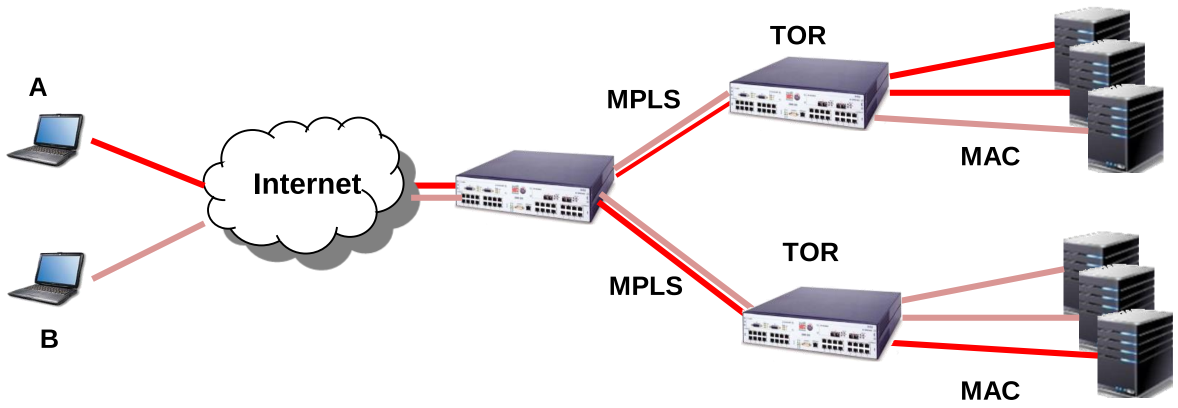
- VLAN is used to isolate networks
  - Uses VLAN tag or switch port number
  - Isolate L2 broadcast domain per user
- Problems
  - VLAN ID = 2^{12} = 4,096 → Multi-tenants problem in Cloud Computing env.
- Solutions
  - VxLAN (CISCO, VMWare), NVGRE(MS), extends VLAN ID to 2^{24}
    - Installed in Virtual Switches in Hypervisor
    - VMware vSphere 5.x & CISCO Nexus 1000v VEM (Virtual Ethernet Switch) support VxLAN
    - Microsoft Hyper-V supports NVGRE



VLAN using OpenFlow

❖ VLAN Implementation with OpenFlow

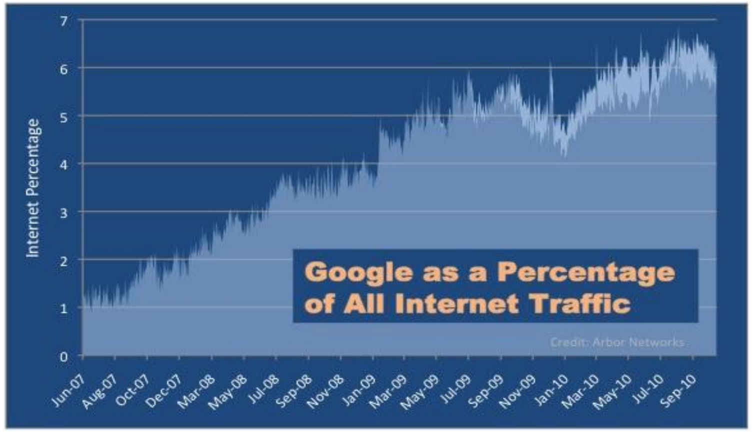
- OpenFlow can identify Virtual Networks only with source & destination MAC address without the need of VLAN IDs
- If MPLS labels are used with MAC addresses, then more Virtual Networks can be supported
  - MPLS label = I/F name + label number (20bits)
    - Static: 0 – 1023
    - Dynamic: 1024 - 1048575



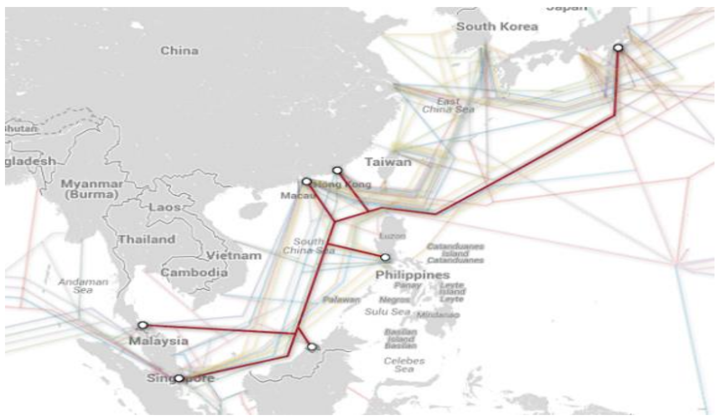
Google SDN

❖ Google SDN

- Google IP Traffic
  - Increases 40~45% every year
    - 8~12% of total Internet traffic
- 36 Google Data Centers in the World
  - 3 DCs under construction
    - USA, Taiwan,… $600M/DC, 60 staff/DC
- DCs connected with submarine cables and long distance dedicated optical cables
  - Large-scale investment, but 30~40% link utilization



Google Data Centers



28 Tera bps cable

(6 companies including Google, KDDI invested, 2010 open)

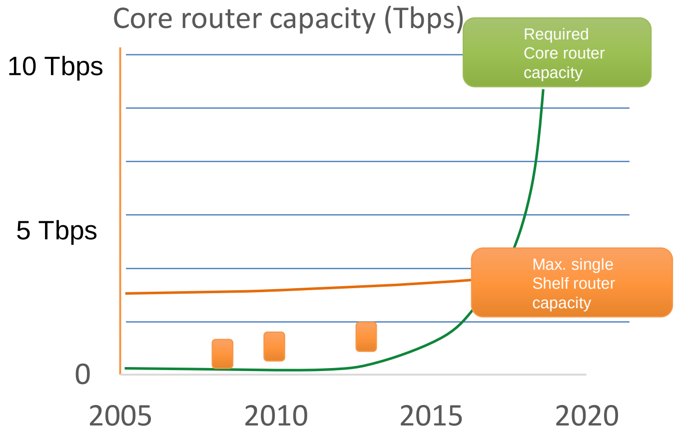
Google SDN

❖ Problems

- WAN Routers treat all bits the same
- WANs links are provisioned to 30% ~ 40% average utilization
  - To protect against failures and packet loss…
- Multi-vendor routers and switches
- Commercial HE/HA Routers
  - Traffic increases → need expensive Tera bit routers
  - Per port Router cost
  - switch failures typically result from software

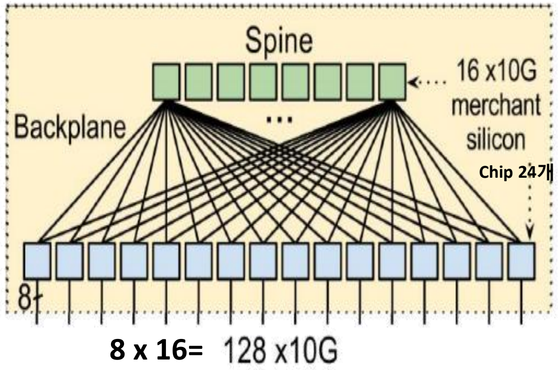
❖ Adoption of SDN and TE

- Commercial routers cannot follow the increase of Google traffic volume
- As a solution for IP based WAN technology problems



Google SDN

- ❖ **Design of B4 SDN**
  - Thousands of individual applications categorized into three classes:
    - user data copies (e.g., e-mail, documents, audio/video files) to remote data centers for availability/durability → latency sensitive → highest priority
    - remote storage access for big data analysis
    - large-scale data push synchronizing state across multiple data centers
  - Design of Centralized Traffic Engineering System
    - Assign relative application priority and control burst at the edge
- ❖ **Development of OpenFlow Switch**
  - No existing platform could satisfy Google's requirements
  - 10G x 128 ports
  - Installation of OpenFlow Agent



Google SDN

❖ Design

- Traffic Engineering System
  - For scalability, TE cannot operate at the granularity of individual applications
  - TE maps FGs to tunnels and corresponding weights
  - Uses ECMP
- Network Control System (NCS) (3 replicas)
  - OpenFlow controller
    - Modified Nicira's ONIX (distributed OF control platform to support large scale network)
    - Manages flow tables and ECMP group table
  - Quagga stack
    - Support BGP/IS-IS, exchange routing protocol information among switches
  - Paxos
    - Detect the failure and elects one of the available NCS