OpenFlow-based Virtualized Testbed on KOREN

[Deployment of OpenFlow Testbed on KOREN]

Feb. 23, 2010

Seung-Joon Seok
(Kyungnam University)

Contents

- **OpenFlow Overview**
  - OpenFlow Concept, Technology, Protocol, etc.
- **Deploying a Testbed on KOREN**
  - Devices (Switch, Controller) Deployment
  - Testbed Configuration
  - Path Control System Development
- **Experiment**
  - Video Stream Transmission from SAPPORO to DEAJEON
- **Summary**

OpenFlow Technology Overview

Innovations in Legacy Internet

▶ **Problem with our network**
- ➢ Paths are fixed (by the network)
- ➢ IP-only
- ➢ Addresses dictated by DNS, DHCP, etc
- ➢ No means to add our own processing
- ➢ …

▶ **Experiments we'd like to do new**
- ➢ Mobility management
- ➢ Network-wide energy management
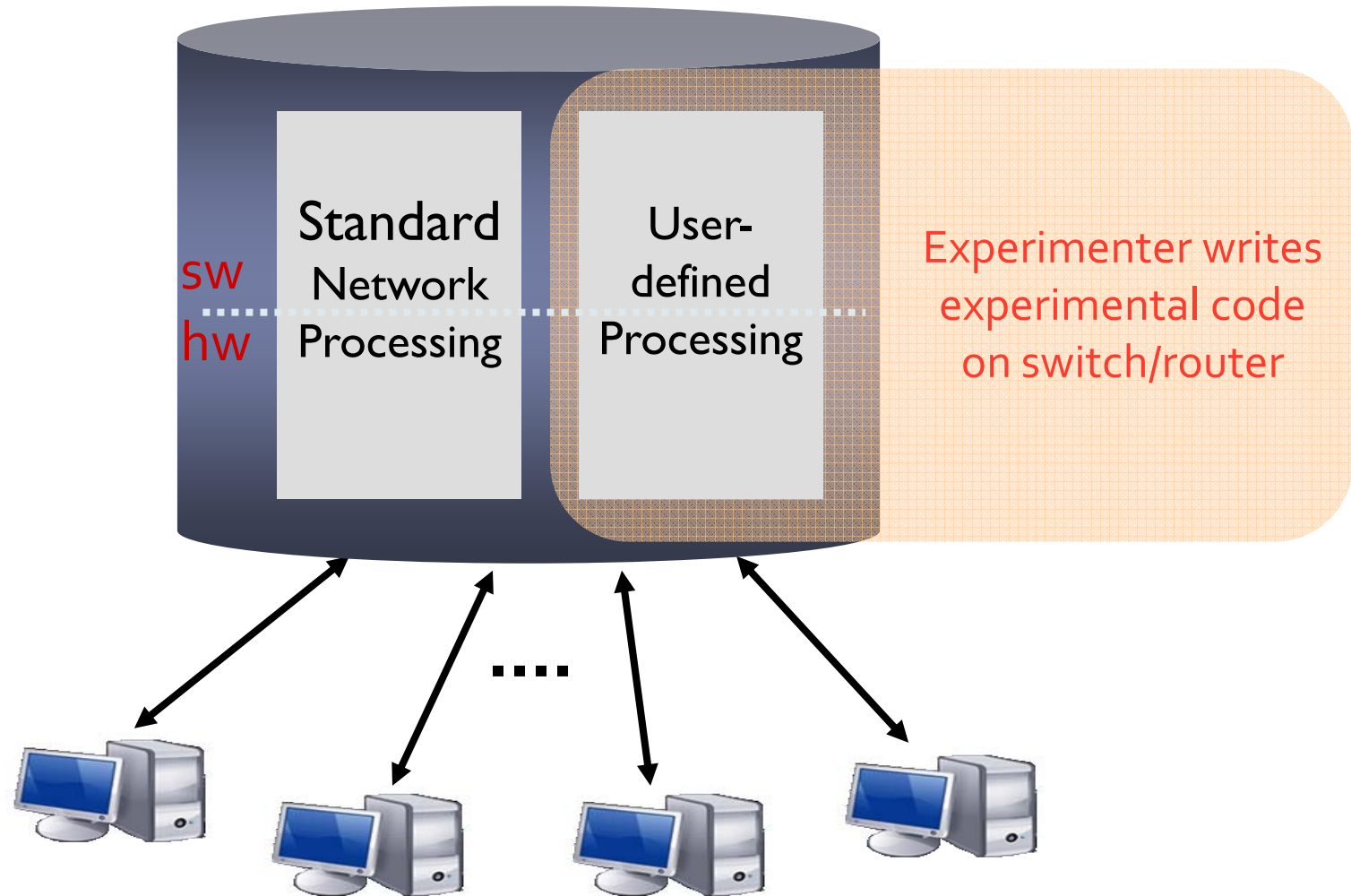- ➢ New naming/addressing schemes
- ➢ Network access control
- ➢ …

Why Internet Closed for Innovations?

▸ Commercial Vendor won't open software and hardware development environment

  ▸ Complexity of support

  ▸ Market protection and barrier to entry

▸ Hard to build my own

  ▸ Prototypes are unstable

  ▸ Software only contribution is Too Slow

  ▸ Hardware/software: Fanout too small
    (need > 100 ports for wiring closet)

Experimenter's Dream
(Vendor's Nightmare)

sw
hw

Standard Network Processing

User-defined Processing

Experimenter writes experimental code on switch/router

....

[The Stanford Clean Slate Program]
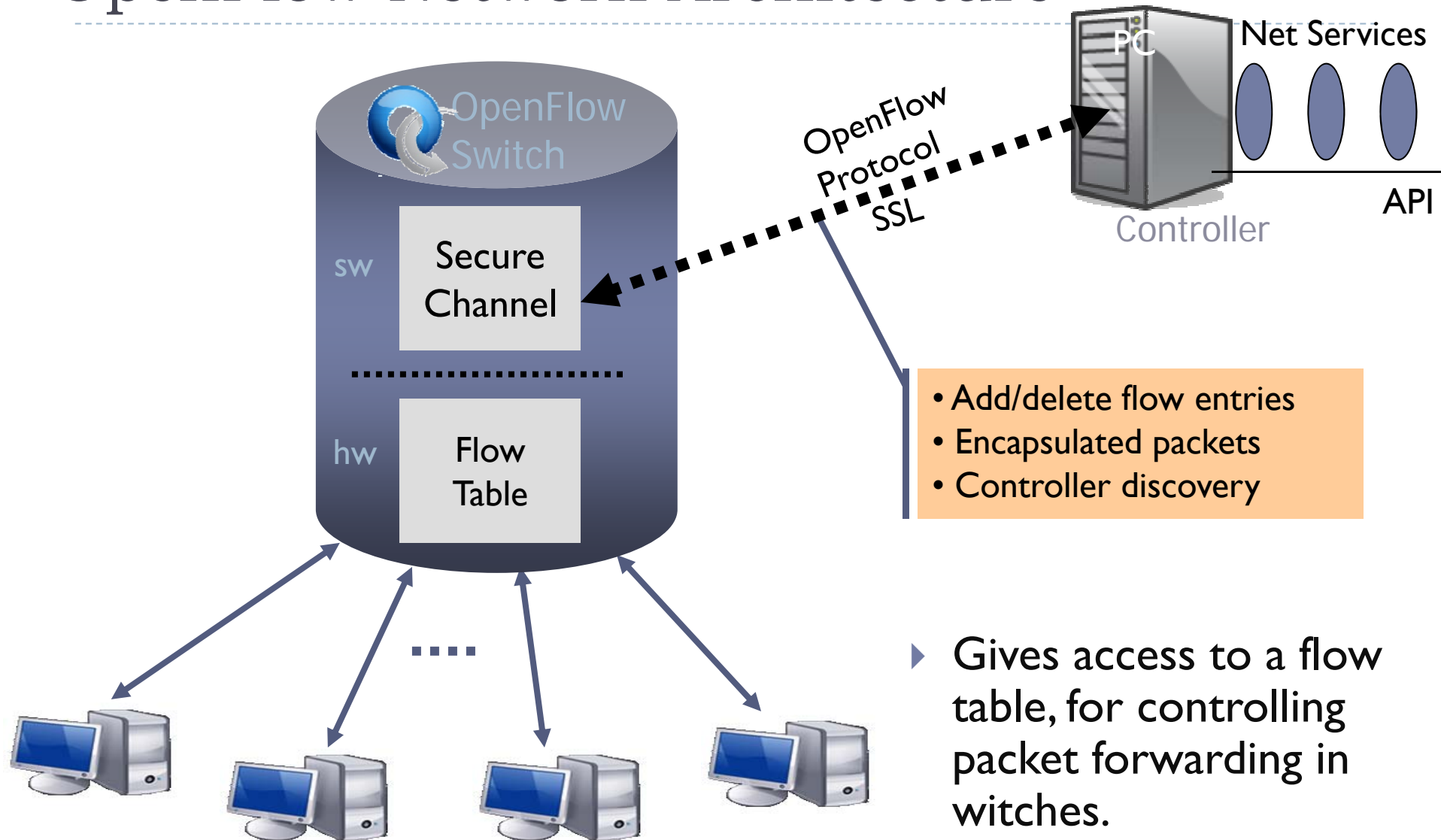
OpenFlow's Goal

- Put an open platform in hands of researchers/students to test new ideas at scale through production networks.
  - without requiring vendors to expose internal workings

- Bring Future Internet to legacy Internet
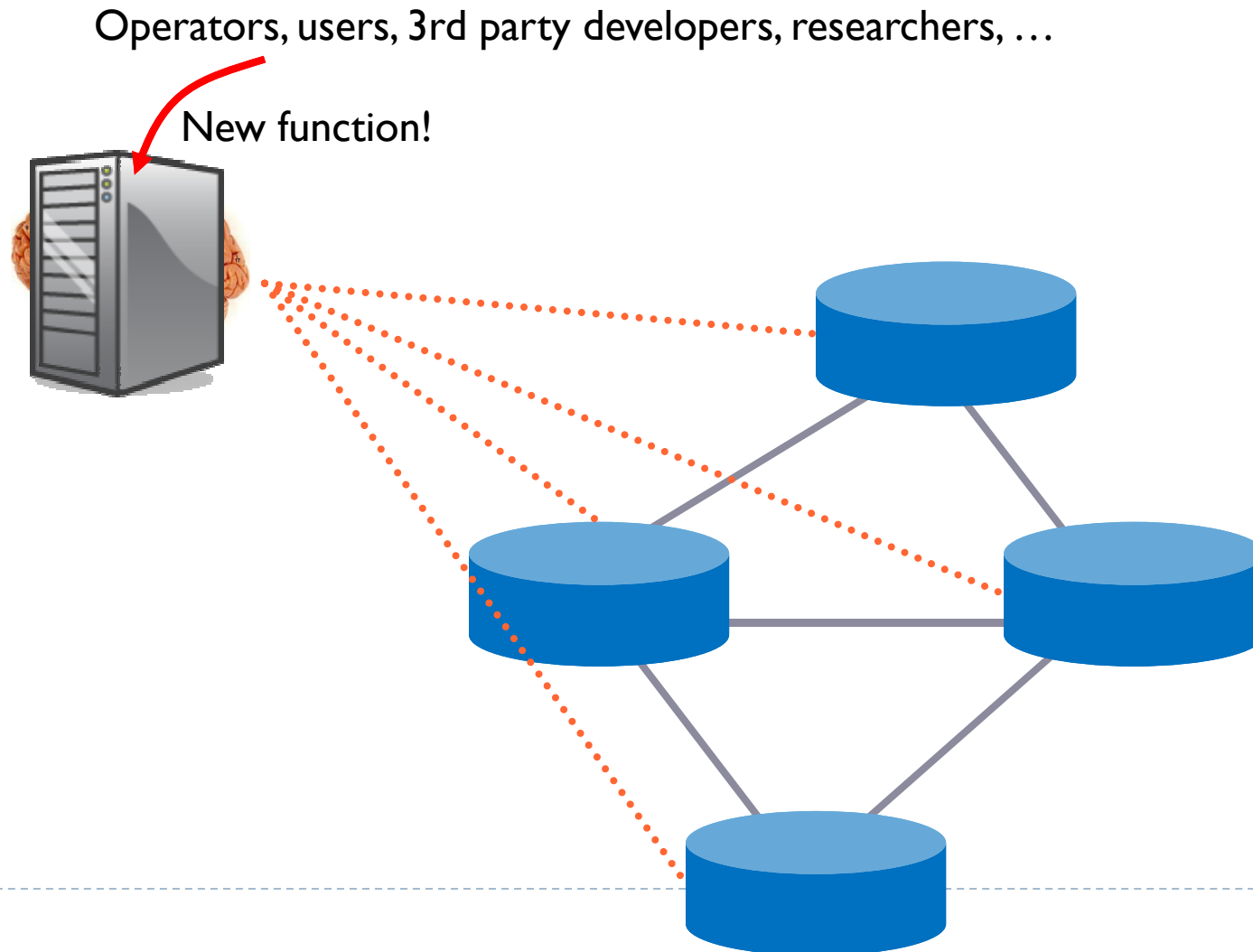- An open development environment for all researchers (e.g. Linux, Verilog, etc)
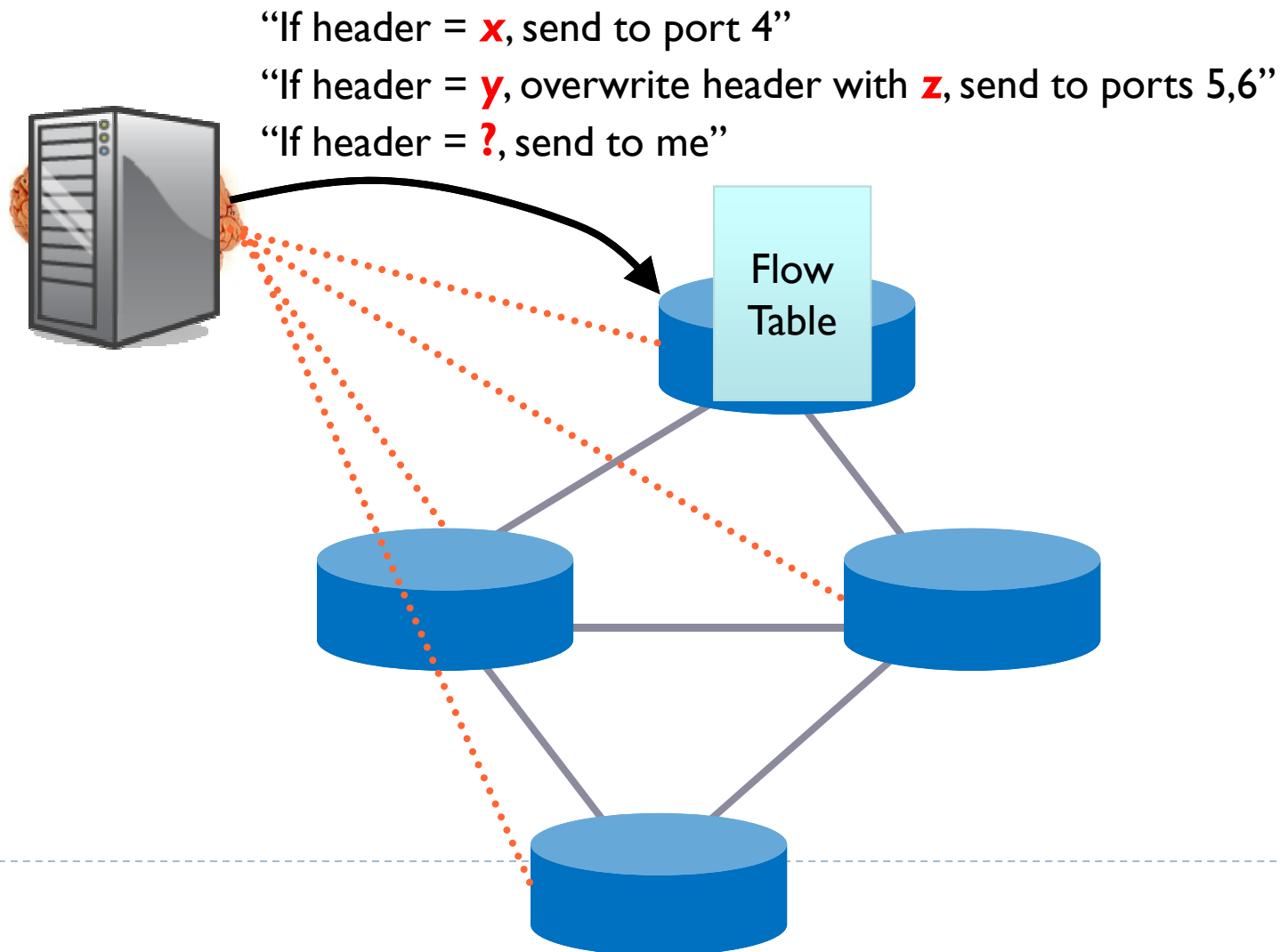
OpenFlow Network Architecture

OpenFlow Switch

sw — Secure Channel

hw — Flow Table

OpenFlow Protocol SSL

PC — Net Services

API

Controller

- Add/delete flow entries
- Encapsulated packets
- Controller discovery

▸ Gives access to a flow table, for controlling packet forwarding in witches.

▸

Step1:
Separating Intelligence from Datapath

Operators, users, 3rd party developers, researchers, …

New function!

Step2: Cache Decisions in Flow-based Datapath

"If header = **x**, send to port 4"

"If header = **y**, overwrite header with **z**, send to ports 5,6"

"If header = **?**, send to me"

Flow Table

Flow Table Structure

▸ Exploit the flow table in switches, routers, and chipsets

	Rule (exact & wildcard)	Action	Statistics
Flow 1.	Rule (exact & wildcard)	Action	Statistics
Flow 2.	Rule (exact & wildcard)	Action	Statistics
Flow 3.	Rule (exact & wildcard)	Action	Statistics
	. . . .		
Flow N.	Rule (exact & wildcard)	Default Action	Statistics

Flow Table Entry

Rule	Action	Stats

Packet + byte counters

1. Forward packet to port(s)
2. Encapsulate and forward to controller
3. Drop packet
4. Send to normal processing pipeline

Switch Port	MAC src	MAC dst	Eth type	VLAN ID	IP Src	IP Dst	IP Prot	sport num	dport num

+ mask what fields to match

Packet Processing

▸ OpenFlow SW's Packet Processing

  ▸ Search a matched entry of flow table with arriving packet's information.
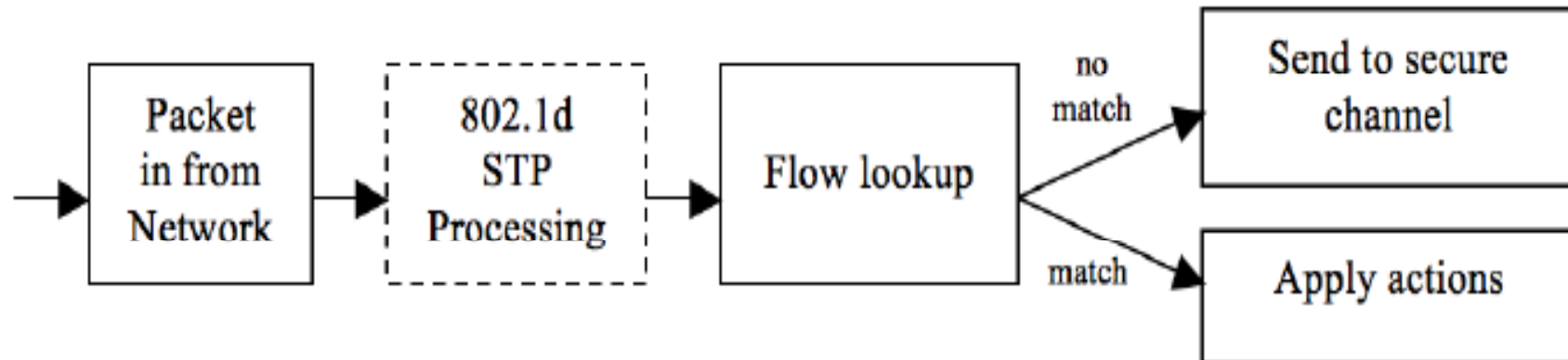
Flow Table Entry Examples (OpenFlow is Backward Compatible)

Ethernet Switching

Switch Port	MAC src	MAC dst	Eth type	VLAN ID	IP Src	IP Dst	IP Prot	TCP sport	TCP dport	Action
*	*	00:1f:..	*	*	*	*	*	*	*	port6

IP Routing

Switch Port	MAC src	MAC dst	Eth type	VLAN ID	IP Src	IP Dst	IP Prot	TCP sport	TCP dport	Action
*	*	*	*	*	*	5.6.7.8	*	*	*	port6

Application Firewall

Switch Port	MAC src	MAC dst	Eth type	VLAN ID	IP Src	IP Dst	IP Prot	TCP sport	TCP dport	Action
*	*	*	*	*	*	*	*	*	22	drop

▶[The Stanford Clean Slate Program]

Flow Table Entry Examples (OpenFlow allows layers to be combined)

Flow Switching

Switch Port	MAC src	MAC dst	Eth type	VLAN ID	IP Src	IP Dst	IP Prot	TCP sport	TCP dport	Action
port3	00:2e..	00:1f..	0800	vlan1	1.2.3.4	5.6.7.8	4	17264	80	port6

VLAN + App

Switch Port	MAC src	MAC dst	Eth type	VLAN ID	IP Src	IP Dst	IP Prot	TCP sport	TCP dport	Action
*	*	*	*	vlan1	*	*	*	*	80	port6, port7

Port + Ethernet + IP

Switch Port	MAC src	MAC dst	Eth type	VLAN ID	IP Src	IP Dst	IP Prot	TCP sport	TCP dport	Action
port3	00:2e..	*	0800	*	*	5.6.7.8	4	*	*	port 10

OpenFlow Hardware (Switches)

Juniper MX-series

NEC IP8800

WiMax (NEC)

HP Procurve 5400

Cisco Catalyst 6k

PC Engines
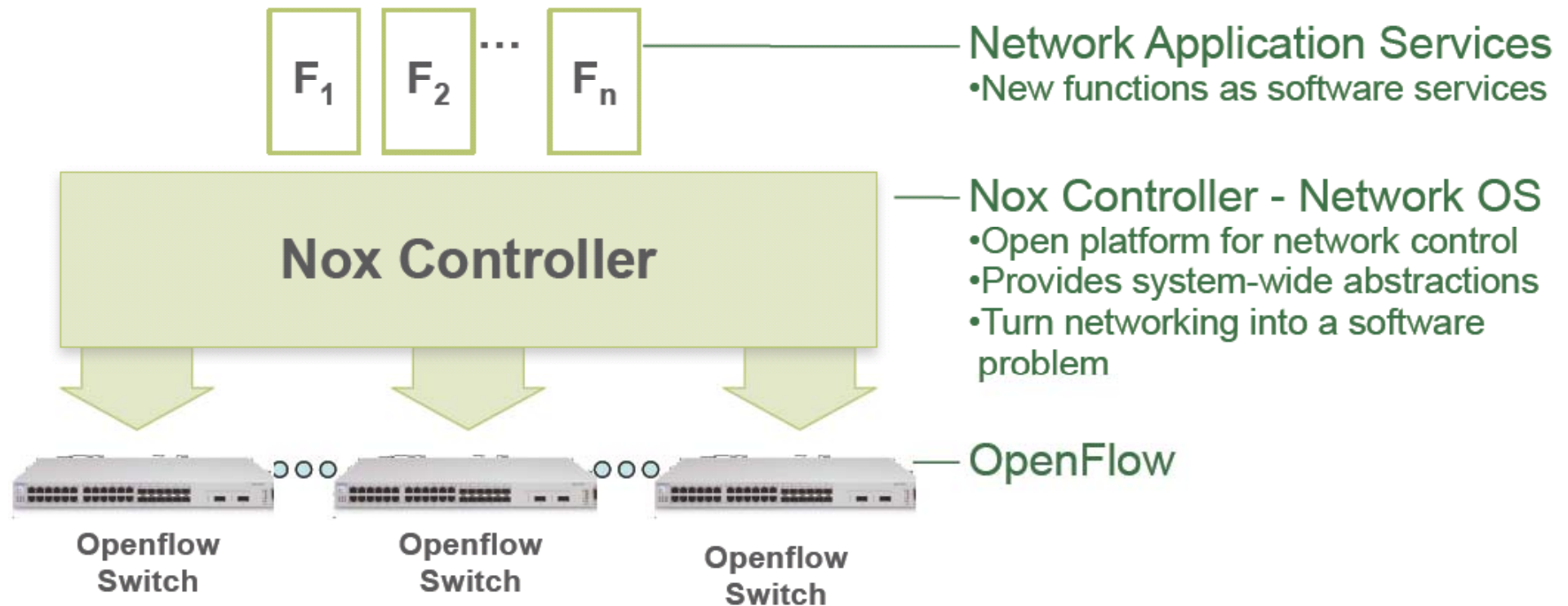
Quanta LB4G

More coming soon...

OpenFlow Controller

▸ **OpenFlow controller is a centralized Intelligential agency for entire OpenFlow network.**

▸ **NOX is an open-source OpenFlow Controller.**

  ▸ created by "Nicira"

  ▸ towards an Operating System for Networks

  ▸ Simplified platform for writing network control software in C++ or Python

  ▸ currently used in a number of large production network deployments

▸ **Researchers can insert their software code into NOX controller for testing their idea.**

Nox Controller

F_1 F_2 ... F_n — **Network Application Services**
- New functions as software services

Nox Controller — **Nox Controller - Network OS**
- Open platform for network control
- Provides system-wide abstractions
- Turn networking into a software problem

— **OpenFlow**

Openflow Switch   **Openflow Switch**   **Openflow Switch**

New Approach to Network Virtualization

New Approach to Network Virtualization

Broadcast

Multicast

http
Load-balancer

OpenFlow
Protocol

FlowVisor

OpenFlow
Switch
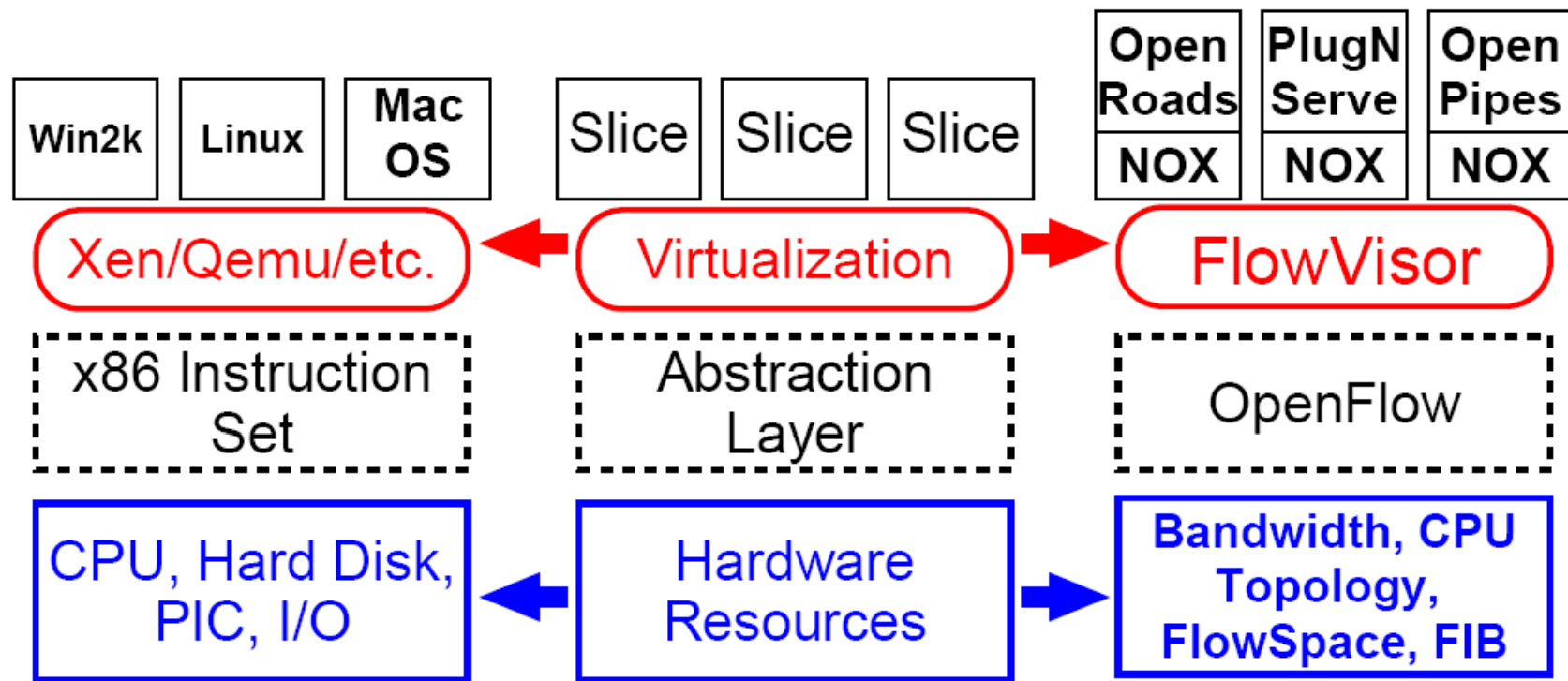
OpenFlow
Switch

OpenFlow
Switch
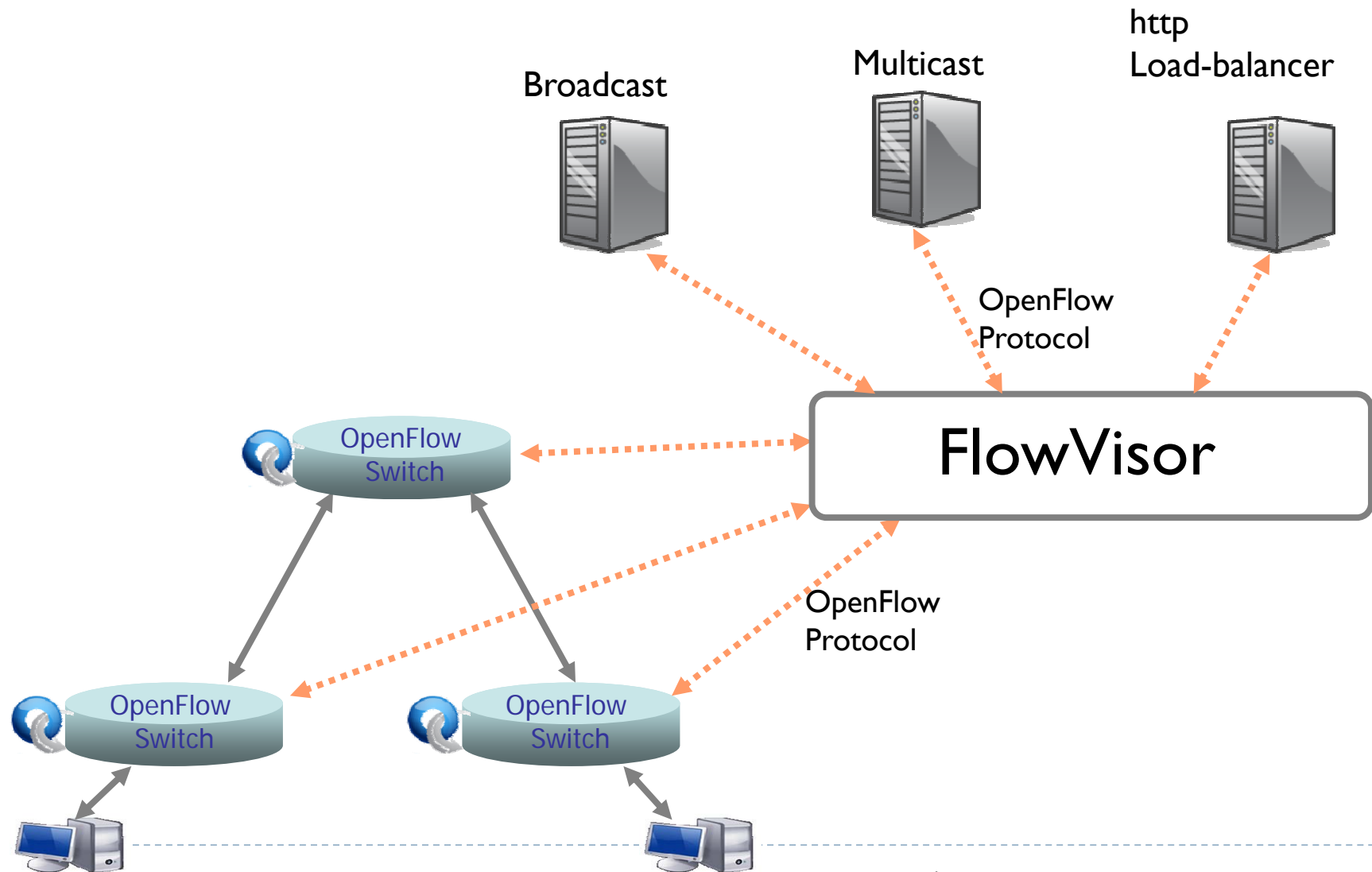
OpenFlow
Protocol

* Deutsche Telekom, "T-Labs"

FlowVisor

- A network virtualization layer in software
- A proxy between OpenFlow switches and guest controllers
  - Parses and rewrites OpenFlow messages as they pass
  - Ensures that one experiment doesn't affect another
- FlowVisor defines a slice as a set of flows running on a topology of switches
  - Allows rich virtual network boundaries
    - By port, by IP, by flow, by time, etc.
- FlowVisor partitions the flow-table in each switch by keeping track of which flow-entries belong to each guest controller.

FlowVisor's Operation
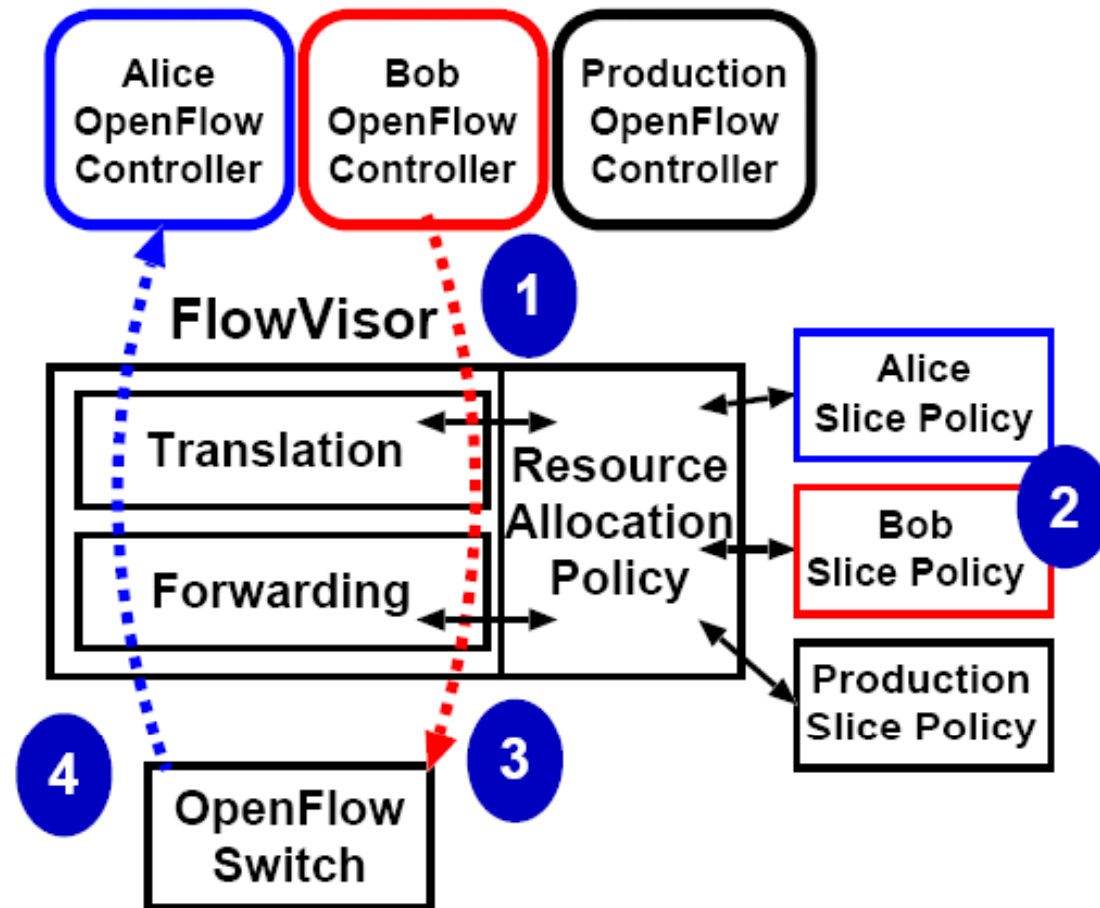
Deploying a Testbed on KOREN

OpenFlow Switch Deployment

▸ **Several vendors are developing the OpenFlow Devices (Switch, Controler)**

  ▸ HP, NEC, Juniper, Cisco, etc.

▸ **We use Linux Box**

  ▸ Open S/W sources

    ▸ http://www.openflowswitch.org/wp/downloads/

    ▸ OpenFlow SW. and Controller.

  ▸ Two types of OpenFlow switche

    ▸ Kernel Level Switching with two ore more GE I/Fs

    ▸ Use NetFPGA PCI card for Ethernet switching isolated from Linux OS. (www.netfpga.org)

▸

Devices of Our Testbed
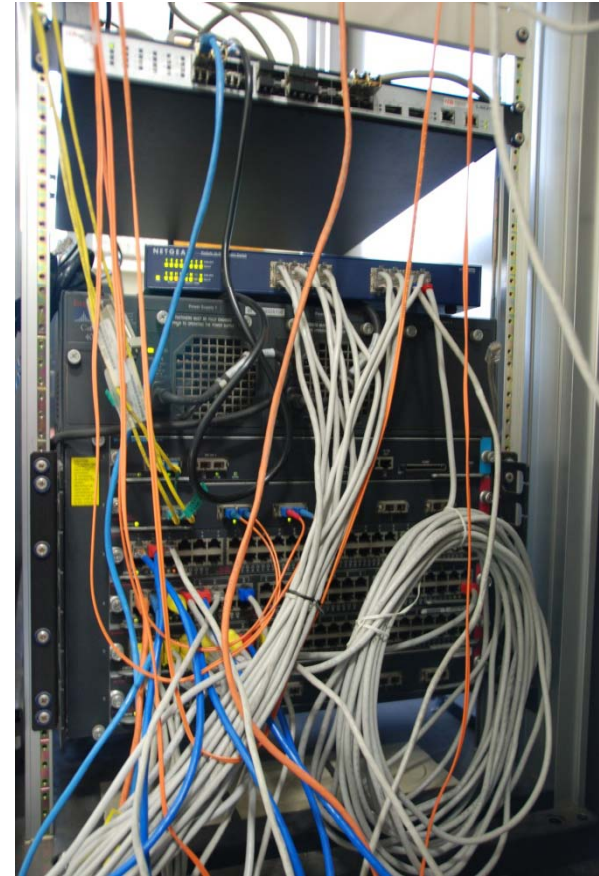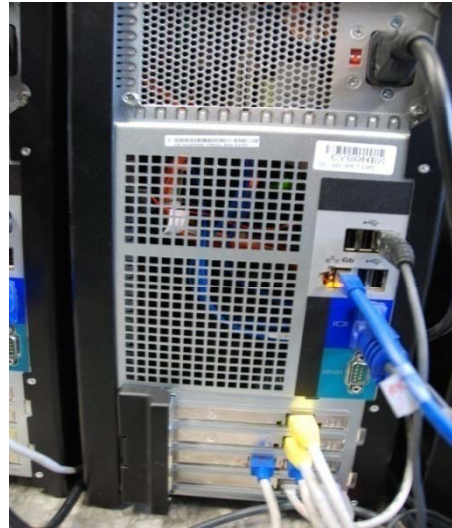
OpenFlow Testbed Toplogy

▸ **Deploy OpenFlow switch in KOREN nodes**

  ▸ Seoul, Suwon, Gwangju, Busan, Desjeon, Jeju, Deagu

▸ **Deploy OpenFlow controller and servers at Daejeon node**
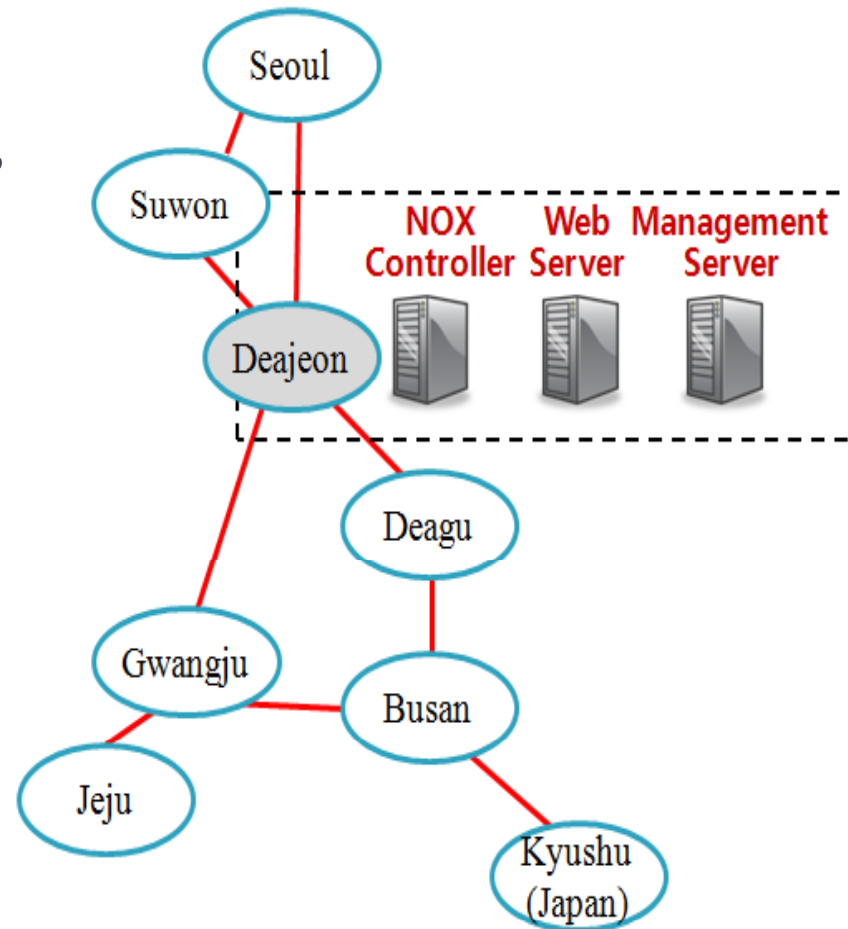
Testbed Configuration 1

▸ There is no enough Lambda in KOREN Links
  ▸ So, connect OpenFlow Switch to KOREN's Ethernet Switch
  ▸ use VLAN to configure OpenFlow testbed (OpenFlow over VLAN)
▸ OpenFlow Switch has a connection with OpenFlow Controller
▸ User and OpenFlow Switch can access to the OpenFlow Controller via Internet
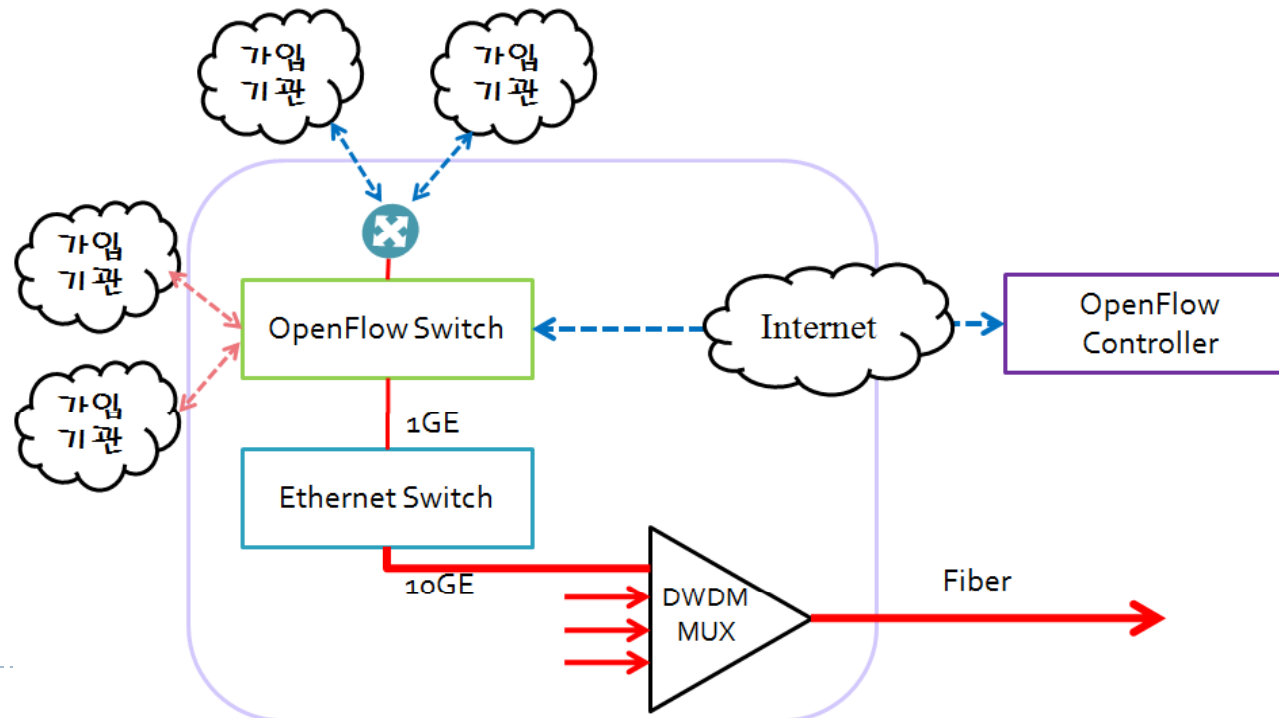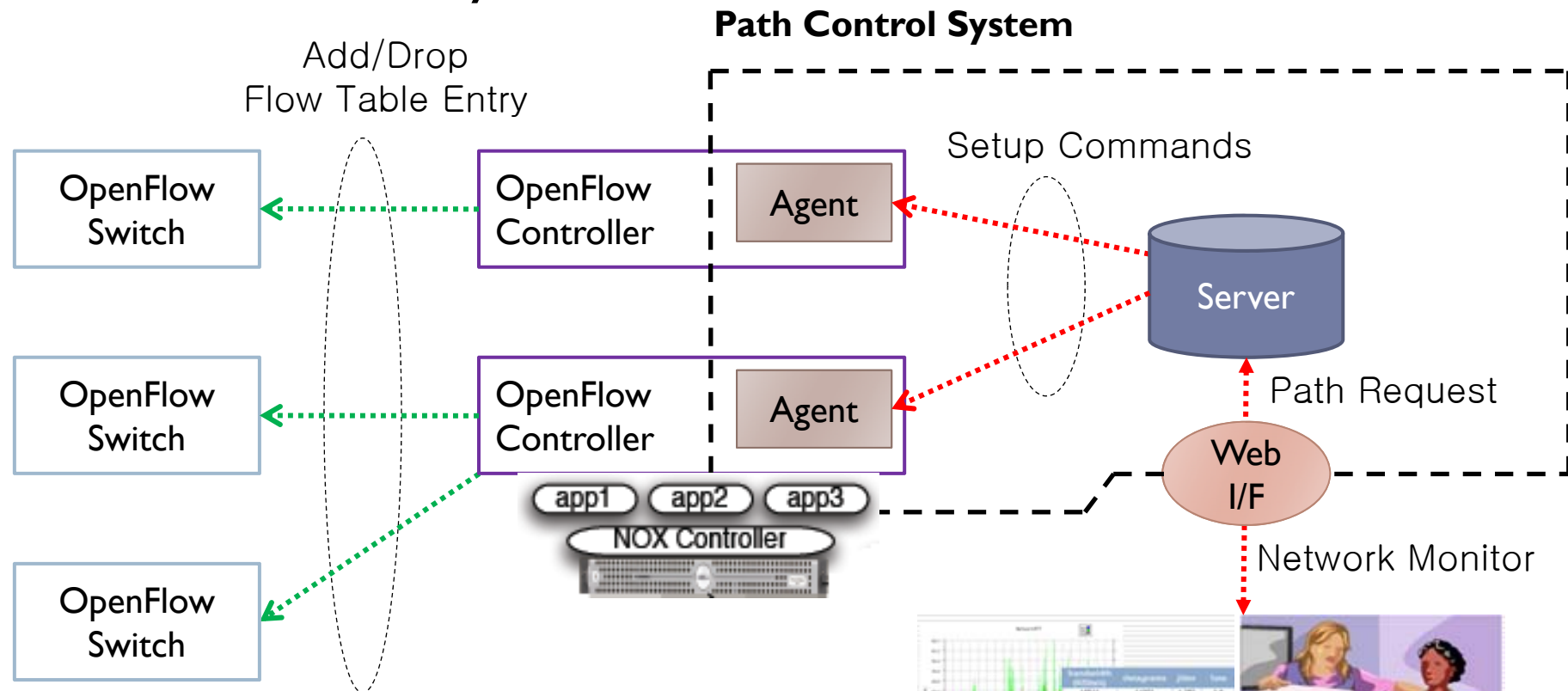
Testbed Configuration 2
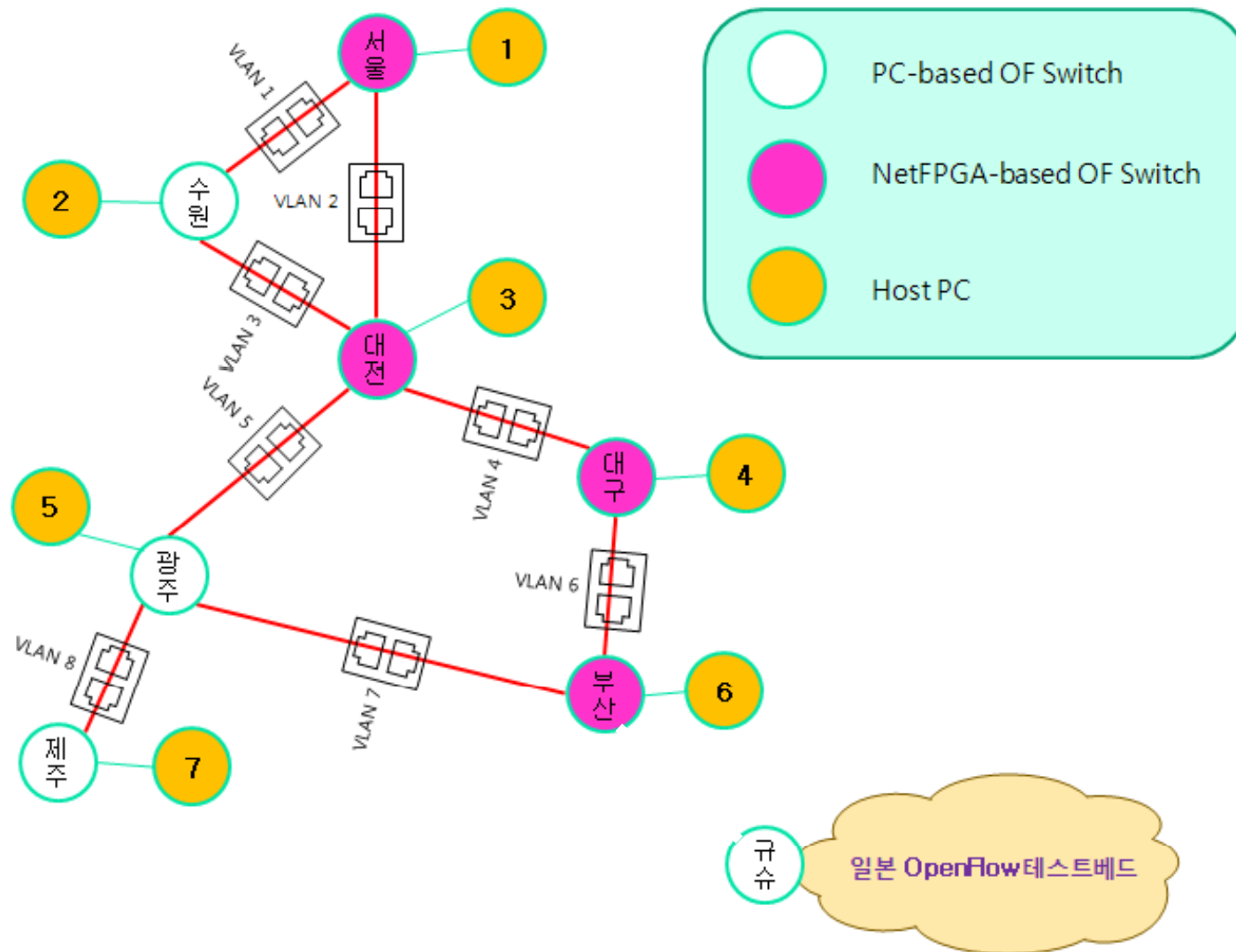
OpenFlow-Path Control System

▸ We want that users can monitor OpenFlow network status and decide flows' path for their experiments

▸ Develop a web-based system where user can request OpenFlow path setup to controllers

▸ Flow-Path Control System

  ▸ Web based system

  ▸ Monitor OpenFlow testbed status

  ▸ Server analyzes your requests and send setup commands to OpenFlow Controller

  ▸ OpenFlow Controller adds/drops flow table entry of OpenFlow switch according to the setup commands
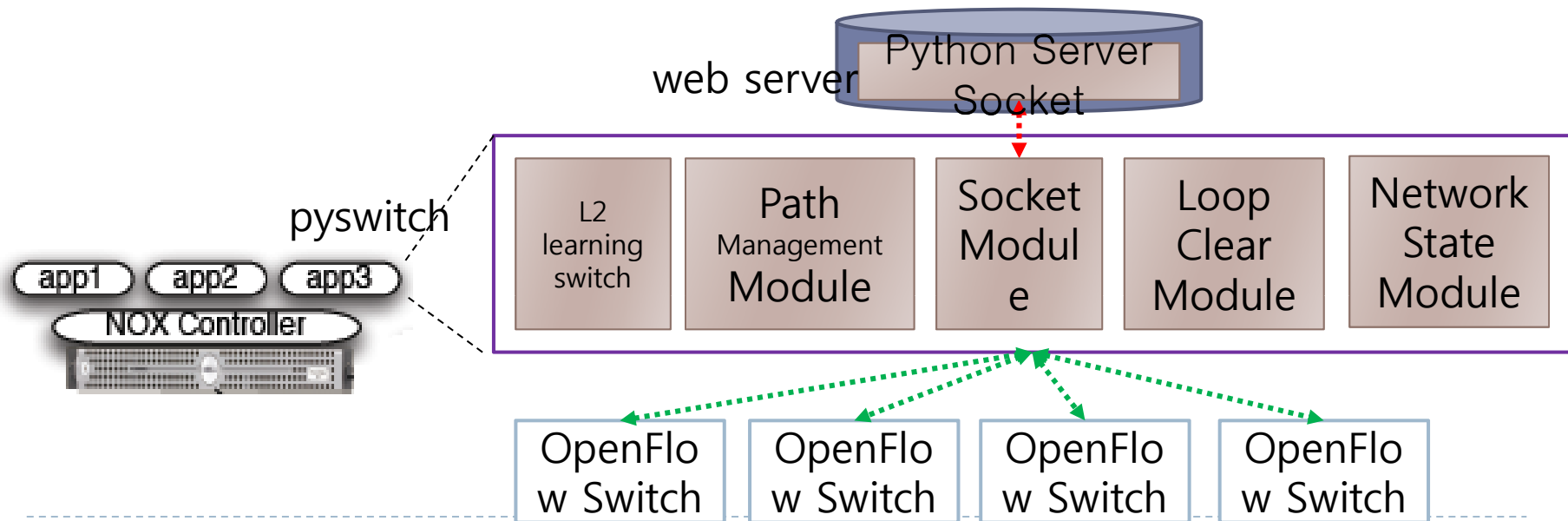
Path Control System

▸ Path Control System architecture

Path Control System

Add/Drop
Flow Table Entry

Setup Commands

OpenFlow Switch

OpenFlow Controller

Agent

Server

OpenFlow Switch

OpenFlow Controller

Agent

Path Request

app1 app2 app3

NOX Controller

Web I/F

Network Monitor

- OpenFlow Network can be monitored using the weather-map application of web server
- OpenFlow paths can be set up through web server

Path Control System (NOX Controller)

▶ **Develop NOX Agent**

  ▶ Modify the previous PYSWITCH app. and add function modules

  ▶ Create/delete flow table entry by user(server) requirements

 ▶ Periodically send network status information to web-server

Path Control System (Web-server)

▶ **Develop web-server module**

　▶ Socket module: communicate NOX Agent

　▶ Weather-Map module: inform network status information

　▶ User Control Path System

　▶ Administrator System module: Flow Path management, user management

User Control Path System	Weather-map	Administrator System

Web UI	DB

Socket 통신 Module	ENVI
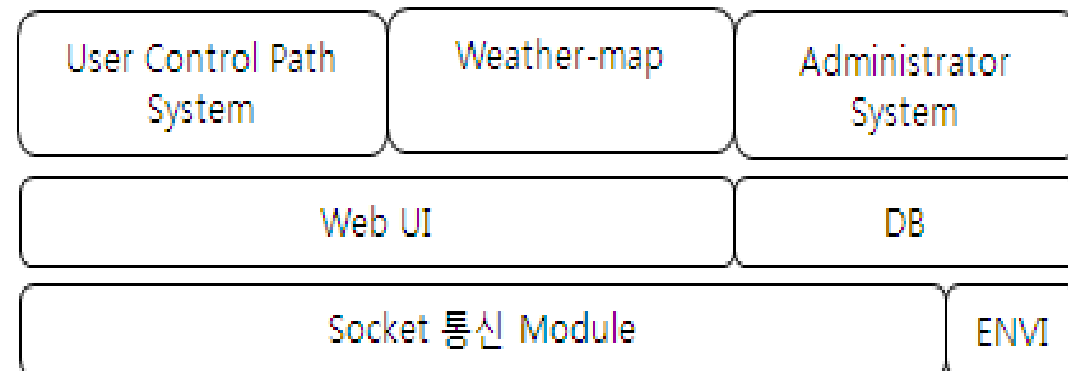
Path Control System (Weather-Map)

▸ **NOX Application**

  ▸ Dynamic topology display

  ▸ Collect switches status information using OpenFlow Protocol
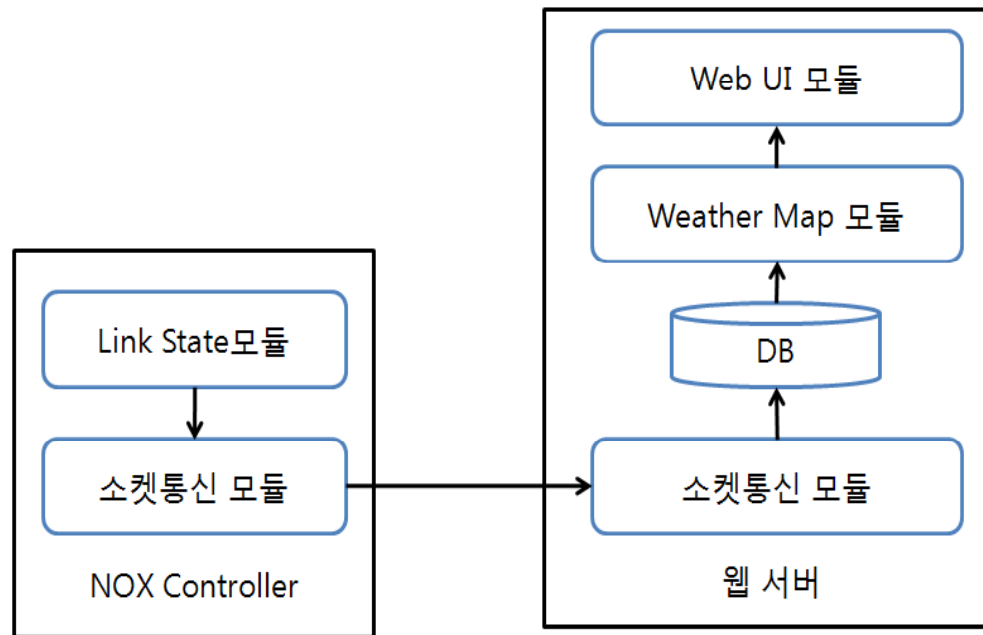
Web interface (Weather-Map)

Path Setup Request Procedure

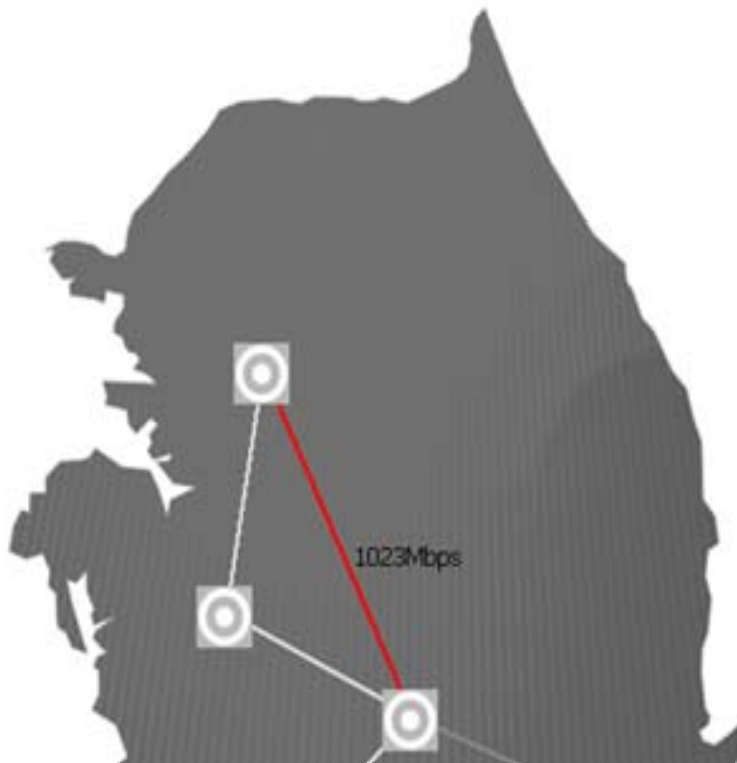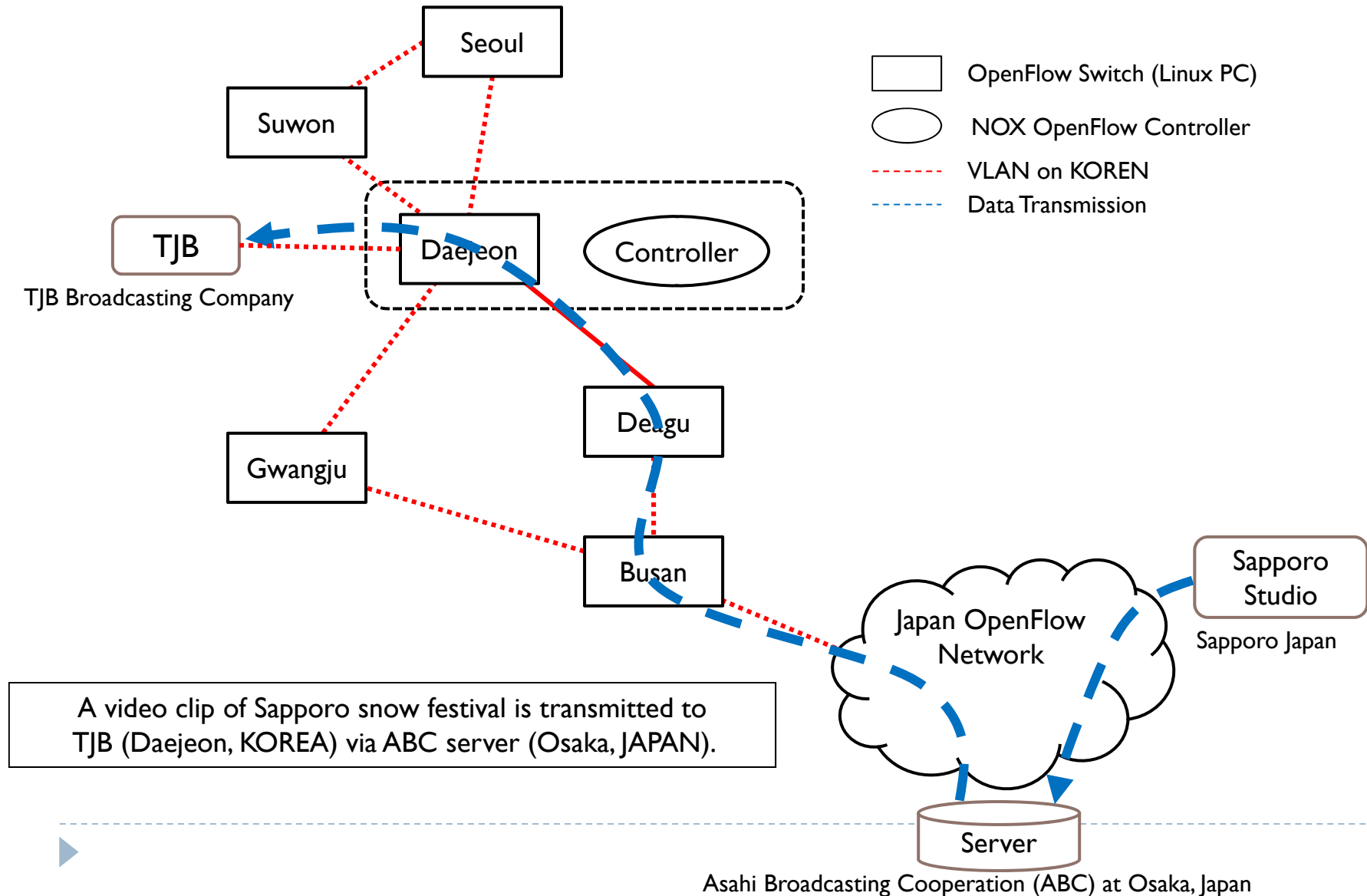# OpenFlow Testbed Path Control

▸ **Change OpenFlow Path**

   ▸ change the flow path using Path Control System
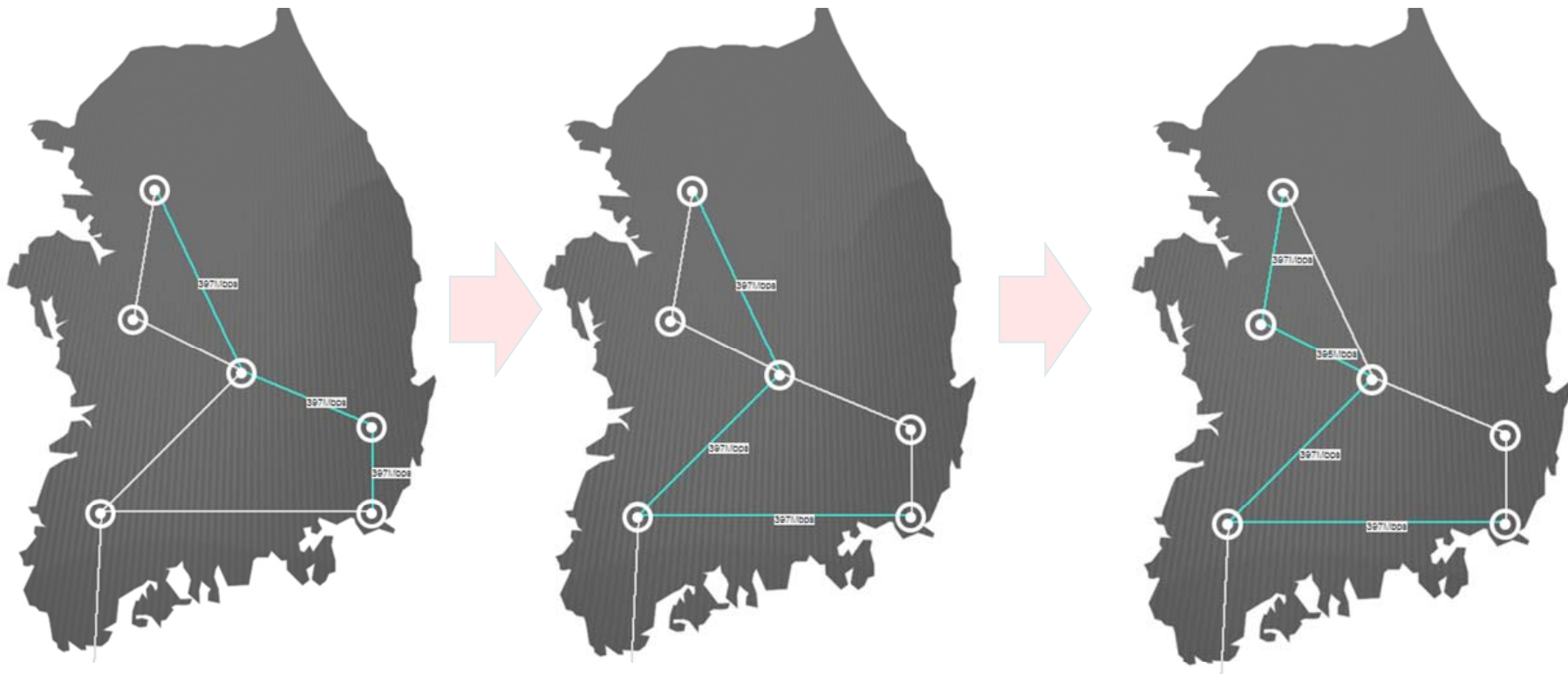
An Experiment
Feb. 2009 - Sapporo Snow Festival Video Transmission

Seoul

Suwon

TJB

TJB Broadcasting Company

Daejeon

Controller

Gwangju

Deagu

Busan

Japan OpenFlow Network

Sapporo Studio

Sapporo Japan

Server

Asahi Broadcasting Cooperation (ABC) at Osaka, Japan

OpenFlow Switch (Linux PC)

NOX OpenFlow Controller

VLAN on KOREN

Data Transmission

A video clip of Sapporo snow festival is transmitted to TJB (Daejeon, KOREA) via ABC server (Osaka, JAPAN).

An Experiment

Summary

▶ **OpenFlow Supports**
  ▶ Test Environment for Future Internet Technologies
  ▶ Network Virtualization
  ▶ High Performance Network Service
  ▶ Real Cross-Layer Network

▶ **OpenFlow Testbed on KOREN**
  ▶ The first national wide OpenFlow testbed in Korea
  ▶ Path control system is open to users
  ▶ Usage for advanced application (like live video streaming)

Thank you for your attention !!

Question or Comment ?