차세대 SDN을 어떻게 준비할 것인가

유 혁

고려대학교

2016. 4. 7

발표자 약력

- **유혁**
  - 고려대학교 컴퓨터학과 교수
  - 2013.07~현재 고려대학교 정보대학장
  - 2008.03~2014.02 SK브로드밴드 사외이사
  - 1990.08~1995.02 Sun Microsystems, Mountain View, CA, USA

- **주요 연구 분야**
  - 시스템 가상화: 임베디드 시스템, 자동차 플랫폼, SSD 기반 I/O, 클라우드
  - 네트워크 가상화: 가상 소프트웨어 라우터 (Xebra), Libera (차세대 SDN)

Contents

I. SDN overview

II. Next Generation SDN (SDN 2.0)

III. Libera: Implementation of SDN 2.0

PART Ⅰ.
SDN OVERVIEW

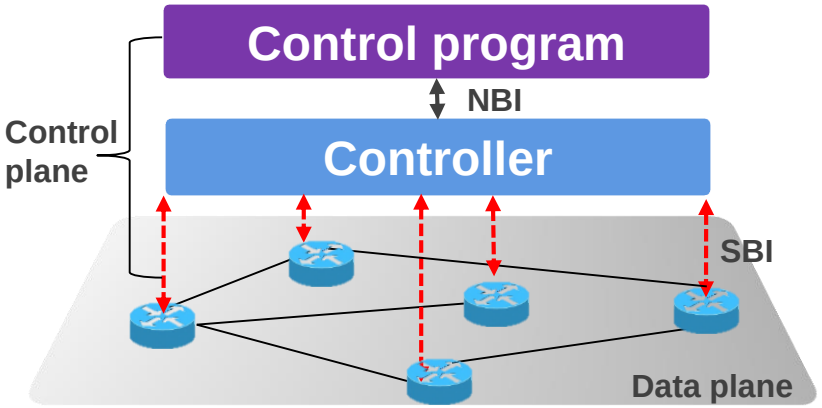
SDN (Software Defined Network)

- **Limitations of existing networks**
  - Difficulty of managing network resources
    - Currently, network resources are managed manually
  - Difficulty of evolving
    - Network is a black box
    - Monolithic architecture of control plane and data plane
  - Overpriced equipment
    - Dependency by device vendors

- **SDN Idea**
  - Separation of the network control plane from the data plane
  - Control plane and data plane become programmable

SDN Architecture

- **Control plane**
  - Control program
    - Express policy and configuration
    - Compute forwarding states (e.g. OSPF)
  - Controller (Network OS)
    - Provide global network view
    - Abstract data plane
  - Southbound Interface (SBI)
    - Interface between control plane and data plane
    - Example: OpenFlow
  - Northbound Interface (NBI)
    - API for control programs
- **Data plane (Switch)**
  - Physical network
    - Process packets with local flow table



Advantages of SDN

- **Flexible network management**
  - Automate management functionality
  - Centrally apply policies to network and services
- **Evolution without dependency**
  - Each layer can evolve independently
  - Protocols and services are upgradable
- **Low Cost**
  - Enable use of commodity hardware
  - Business opportunity can be realized quickly

Existing SDN controllers

Controller		Characteristics
NOX	**Nicira**	• First open source controller • The most primitive form of SDN controller
Floodlight	**Big Switch Networks**	• JAVA based controller • Module-type core to provide simple extension
OpenDayLight	**ODL**	• Aim to support legacy network • Multi-protocol support • Industry vendors play a key role of development
ONOS	**ON.LAB**	• Now evolving into cover the concept of “SDN 2.0” • Distributed architecture for scalability • Provide fault tolerance of controllers

PART Ⅱ.
NEXT GENERATION SDN (SDN 2.0)

Issues in SDN 1.0

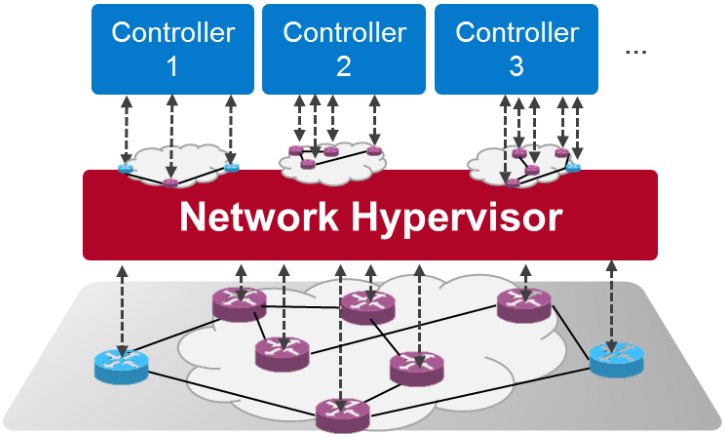
- **Network virtualization is not considered**
  - Adopting virtualization is necessary for efficient network provisioning and isolation
- **Assume that network switches are homogeneous**
  - Reality is that there are switches of various functions (more than forwarding)
- **NFV (Network Function Virtualization) is not considered**
  - Need to support middleboxes in the form of software switches

#1 Network Virtualization is not considered

- **Network Virtualization**
  - Allow multiple virtual networks on a physical network
  - Provision network resources for virtual networks
  - Isolate virtual networks for protection

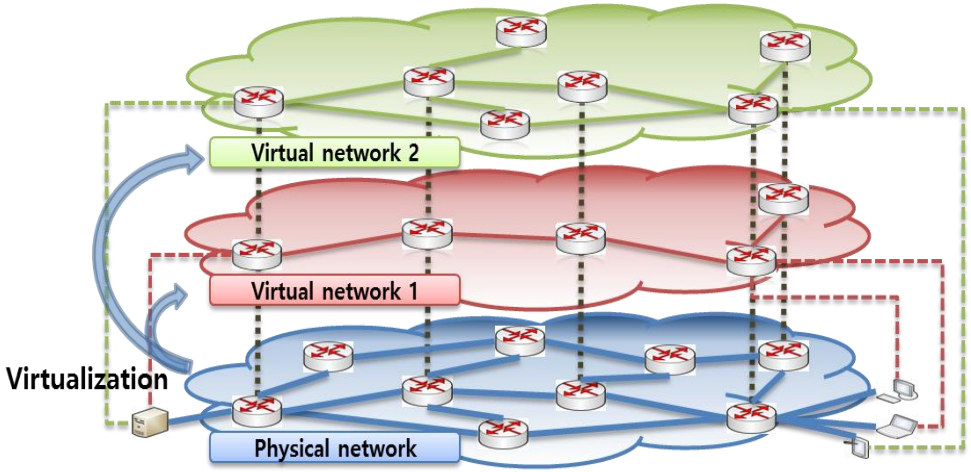
- **SDN can be a platform for "real" network virtualization**
  - SDN controller manages virtual networks
  - In SDN, a virtual network can have its own IP address space and topology

- **Key: Network Hypervisor**
  - Abstract physical network resources



#1 Network Virtualization – Killer App for SDN

- **Network IaaS(Infrastructure-as-a-Service)**
  - For example, enterprise network or campus network can be leased

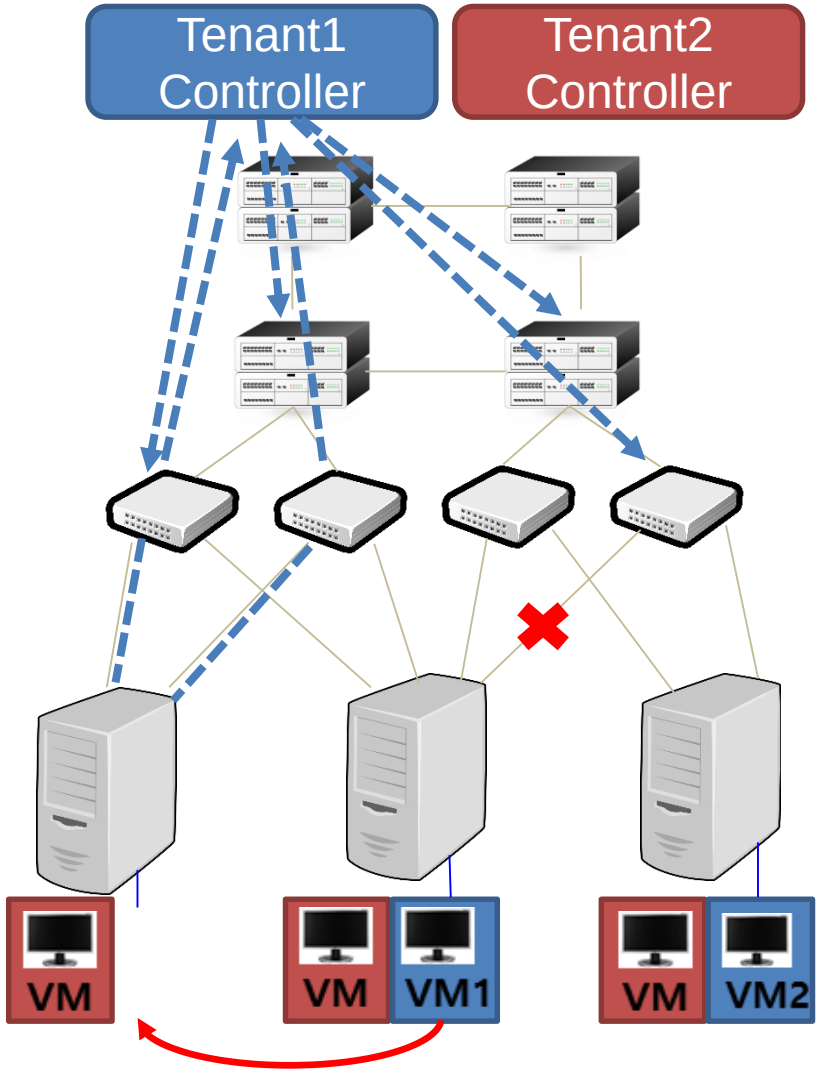


- **Benefits**
  - Reduce time to provision
  - Better utilization of network resources
  - Easy and cheaper to manage networks
  - Strong isolation

#1 Network Virtualization – Killer App for SDN

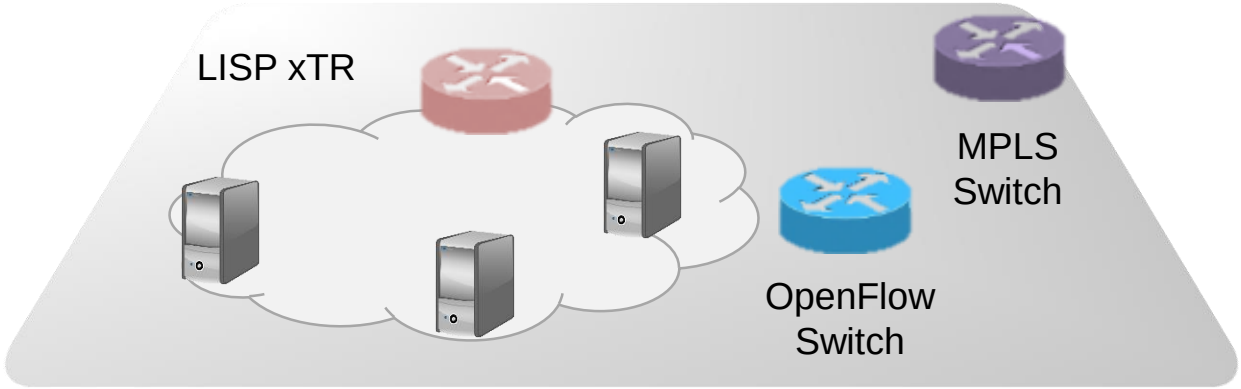
- **Fully automated cloud environment**

- **Seamless migration scenario**
  1. Tenant1 and Tenant2 share infrastructure (servers and switches)
  2. Tenant1's VM1 and VM2 communicate each other
  3. If VM1 migrates to other machine, network configuration should be changed
  4. In SDN, controller automatically detects the migration
  5. Then, controller makes a new rule and passes it to switches
  6. Switches apply the rule



#2 Assume that switches are homogeneous

- **Actual network does not just consist of forwarding machines**
  - Protocols such as LISP and MPLS are widely adopted



- **Current status**
  - OpenFlow has provided labeling, tunneling actions since v1.1
  - Need to support MPLS and LISP along with OpenFlow

#3 NFV is not considered

- **Middlebox commonly used to augment data plane**
  - Carrier network includes middleboxes for various functions
    - E.g. firewall, load balancer, and path recovery
  - The number of middleboxes is already on par with the number of routers in current networks
  - But, SDN 1.0 was designed without consideration of middleboxes

- **NFV makes middlebox as software switches**
  - Single x86 server handles over 16Mpps IPv4 forwarding*
  - Low price compared to commodity hardware switches

- **SDN needs to support NFV**
  - For better programmability and flexibility

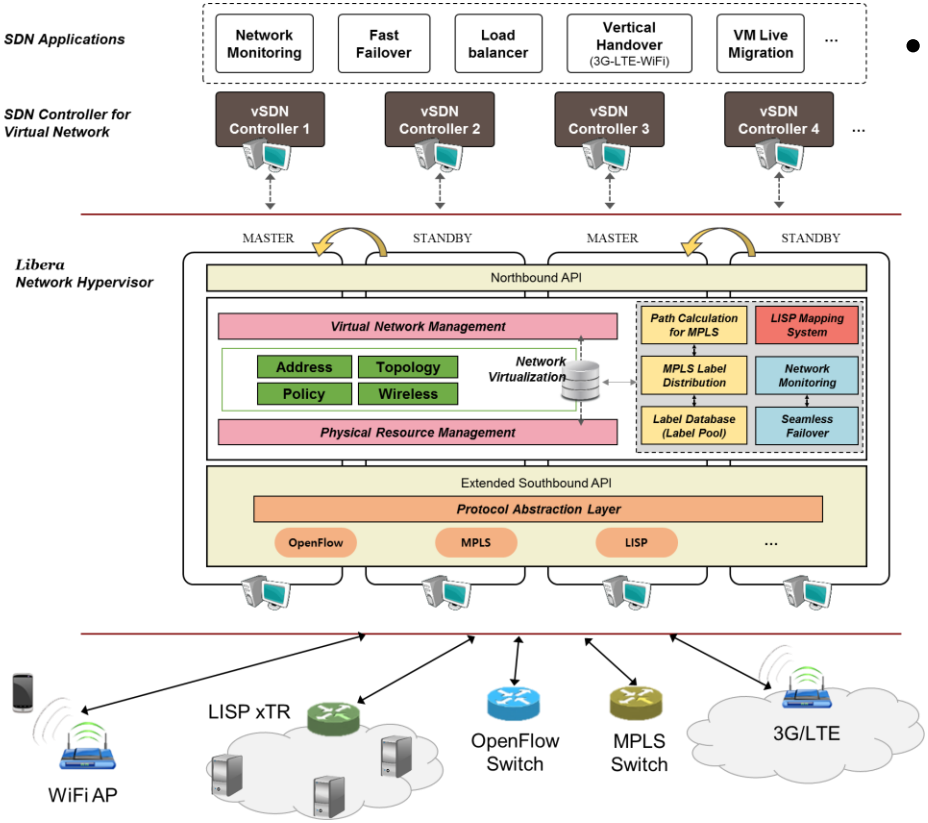
*Intel Data Plane Development Kit (DPDK) White paper. Feb. 2016.

Our Approach

- **Apply virtualization expertise to make SDN 2.0**
  - Extend existing network hypervisor: OVX (OpenVirteX)
  - Add Wireless AP support (authentication among tenants)
- **Support MPLS and LISP switches**
  - Implement extended SBI and NBI
- **Improve scalability and fault tolerance**
  - Multiple tenants and virtual networks
  - Master-standby architecture

PART Ⅲ.
IMPLEMENTATION OF SDN 2.0 (LIBERA)

Libera Architecture



- **Components**
  - **SDN Application**
    - Implement network functions
  - **SDN Controller**
    - Control virtual network
  - **Network Hypervisor**
    - Virtualize network resources
  - **OpenFlow enabled switches**
    - OVS (software switch)
    - MPLS switch
    - LISP xTR
    - Wireless AP

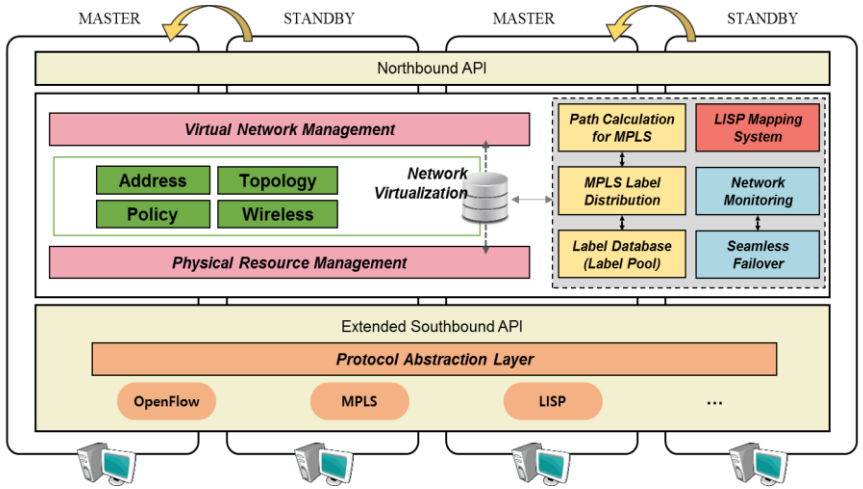
Network Hypervisor (1/2)

- **Design and develop world-class SDN Hypervisor**

- **Network resources virtualization**
  - Topology virtualization
  - IP address virtualization
  - Policy virtualization

- **Wireless network virtualization**
  - AP virtualization, end-host authorization mechanism for security

- **Scalability and fault tolerance**
  - Master-standby architecture



Network Hypervisor (2/2)

- **Topology virtualization**
  - Allow a tenant to specify its own arbitrary topology (i.e. a big switch)
  - LLDP (Link Layer Discovery Protocol) is used to expose virtual topology
- **Address virtualization**
  - Grant tenants to choose IP address assignments for end hosts in virtual networks
  - Tunneling is used to differentiate tenants
- **Policy virtualization**
  - Allow tenants to use their own policy to control the network resources in their virtual network (via controller)
  - Example: bandwidth isolation, flow table consumption per virtual network

SBI Extension Technology (1/4)

SBI extension technology for adapting carrier network

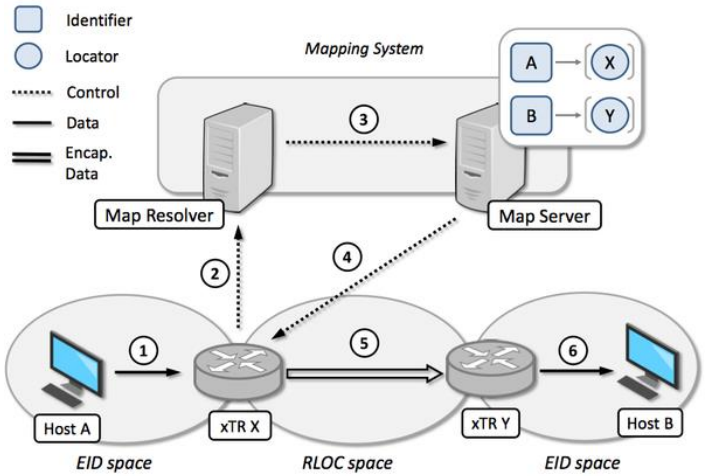
- **MPLS (Multiprotocol Label Switching)**
  - Used for label-based forwarding in core switches
  - Widely used in carrier-grade network

- **Libera design for MPLS**
  - Design management scheme of label distribution and path calculation
  - Develop a new interface to deliver MPLS rules to switches by OpenFlow

SBI Extension Technology (2/4)

- **LISP (Locator/Identifier Separation Protocol)**
  - IP address has two orthogonal functions
    - Routing locators (where a client is attached to the network)
    - Identifiers (who the client is)
  - LISP supports the separation of the IP address space through mapping between:
    - EID (Endpoint ID): uniquely identify a network interface within its local network
    - RLOC (Routing Locator): Identify where a network interface is located
    - IP address is used for EID and RLOC

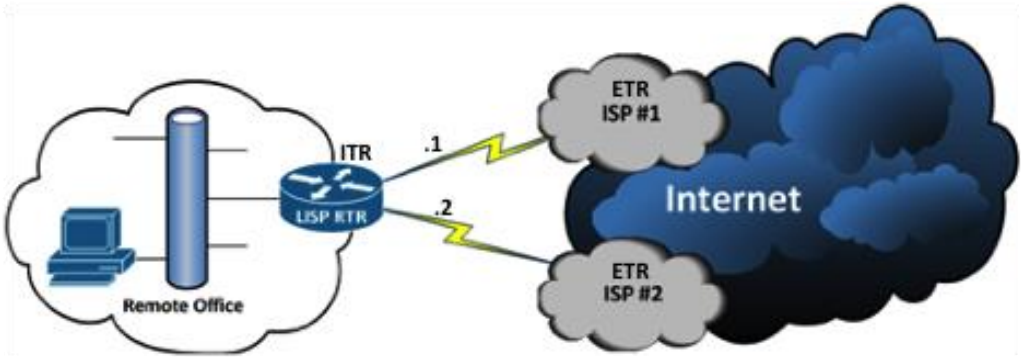
1. Host A wants to communicate with its peer B, from which it only knows its EID
2. xTR X sends a Map-Request to obtain the RLOC of the peer
3. This Map-Request is routed through the Mapping System to finally reach the Map Server containing this information
4. The Map Server replies to xTR X with a Map-Reply message
5. xTR X sends the data to the xTR Y
6. Finally Y forwards it to the peer B



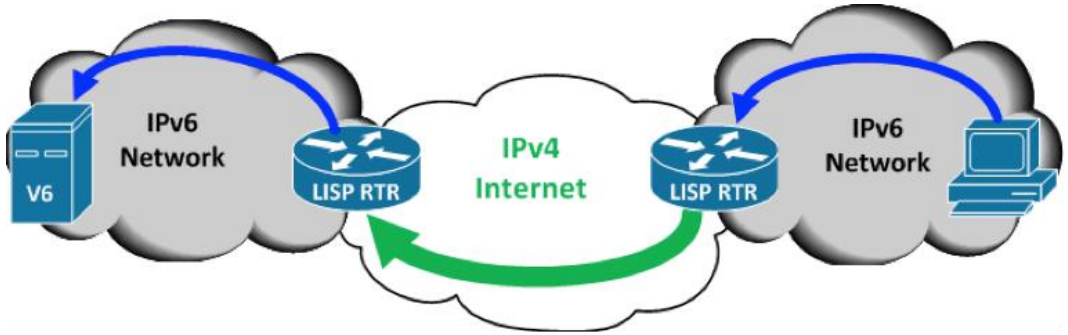
Ref : https://labs.ripe.net/Members/alberto_rodriguez_natal/lispmob-a-flexible-lisp-implementation

SBI Extension Technology (3/4)

- **Advantages of decoupling Location and Identifier**
  - Easy to support multi-homing and mobility
  - Improved routing scalability
  - Address family traversal: IPv4, IPv6



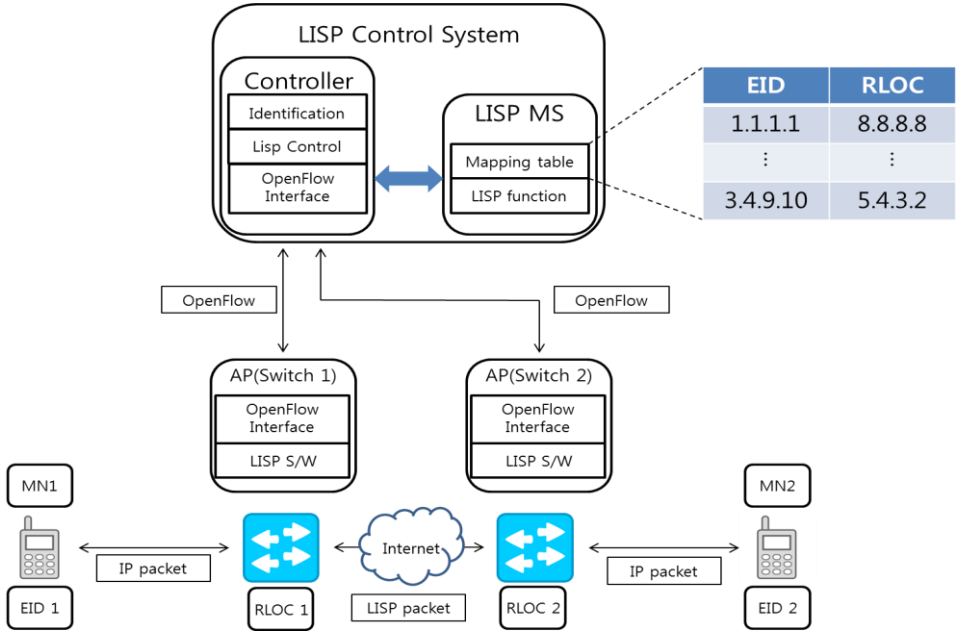
Multi-homing



IPv6 Support

SBI Extension Technology (4/4)

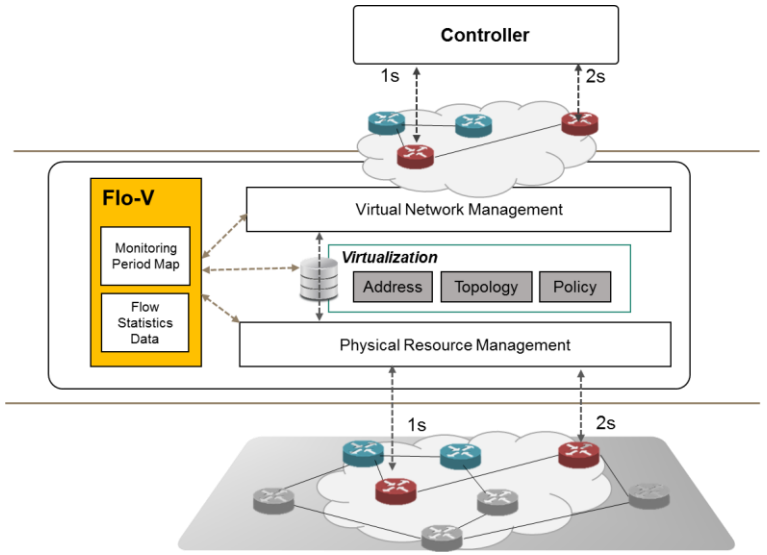
- **Libera design for LISP**
  - Develop LISP controller and mapping system
    - Mapping table exists per virtual network
  - Develop interfaces between LISP control system and OpenFlow protocol
  - Extend OpenFlow actions for LISP tunneling
  - Verify test scenarios (vertical handover, live migration)



Traffic engineering (1/2)

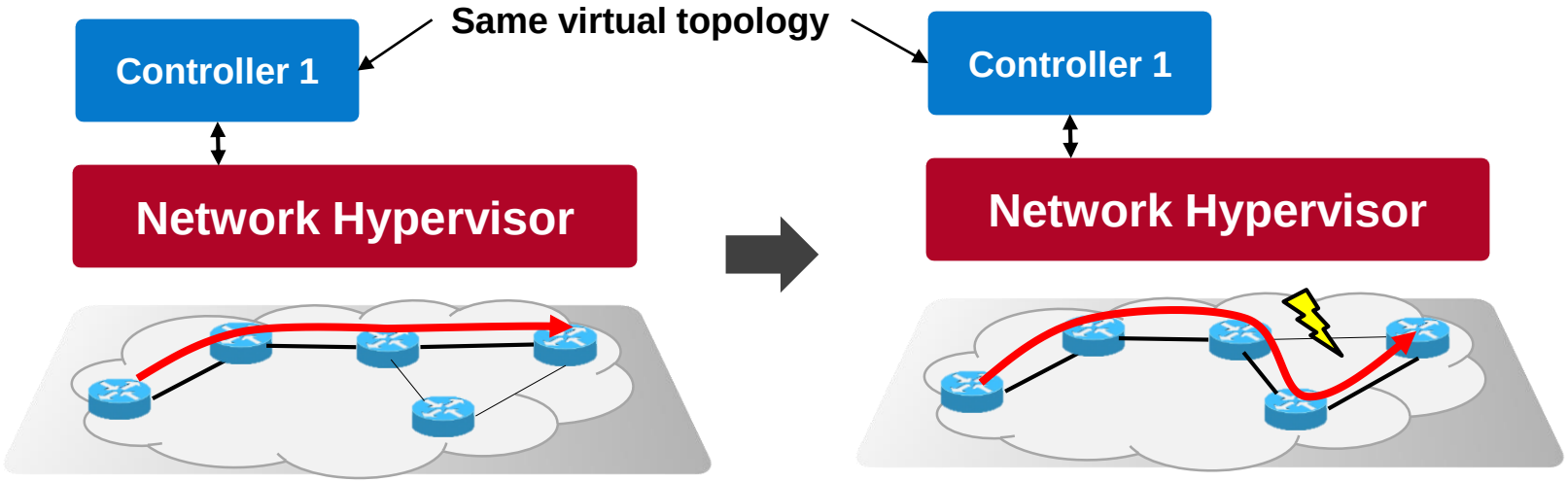
- **Existing research does not consider virtualized SDN (vSDN)**
  - Libera aims to provide quality-guaranteed virtual network

- **Flo-v: traffic monitoring framework in Libera**
  - Provide real-time network status to vSDN controller with low overhead
    - Adaptive monitoring
      - "Monitoring period map": Measure the interval of received monitoring requests from vSDN controllers
      - Differentiate the polling interval per switch
    - Selective monitoring
      - Monitor only the requested resources from vSDN controller
      - Store the gathered information into "Flow Statistics Data"



Traffic engineering (2/2)

- **Seamless failover : fast failover in network hypervisor**
  - Goal: recover link failure within 50ms* to maintain the quality of service
  - When a physical link fails
    - migrate the corresponding virtual links to another physical resources
    - re-construct the virtual network



* Niven-Jenkins, B., et al. Requirements of an MPLS transport profile. No. RFC 5654. 2009.

Existing Network Hypervisors

- **FlowVisor**
  - For OpenFlow-based vSDN
  - Allocate “Flowspace” to each vSDN
  - Multiple virtual networks share same IP address space
- **FlowN**
  - Utilize applications running on a NOX controller
  - Low communication overhead between a controller and a network hypervisor
- **OpenVirtex**
  - Open source network hypervisor based on FlowVisor architecture
  - Support policy and topology virtualization

Comparison with existing techniques

	Libera	FlowVisor	FlowN	OpenVirtex
IP address virtualization	**O**	X	O (VLAN)	△* (IP rewriting)
Extended SBI	**O**	X	X	X
Wireless network virtualization	**O**	X	X	X
Traffic Engineering support	**O**	X	X	△**

*: virtual IP is 1:1 mapped with physical IP
**: lack of MPLS and LISP support

Q & A