Empowered by Innovation NEC

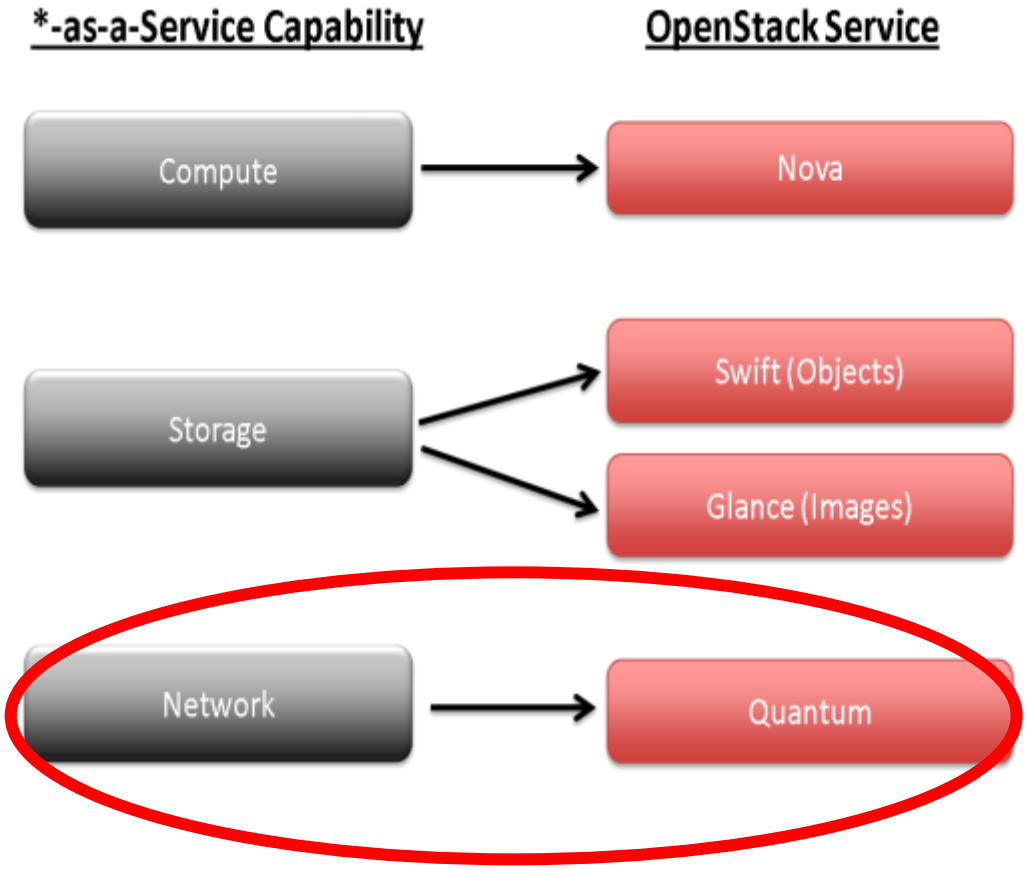
OpenStack with OpenFlow/SDN

Takashi Torii
NEC
Aug. 11, 2012

INTRO - QUANTUM

What is Quantum

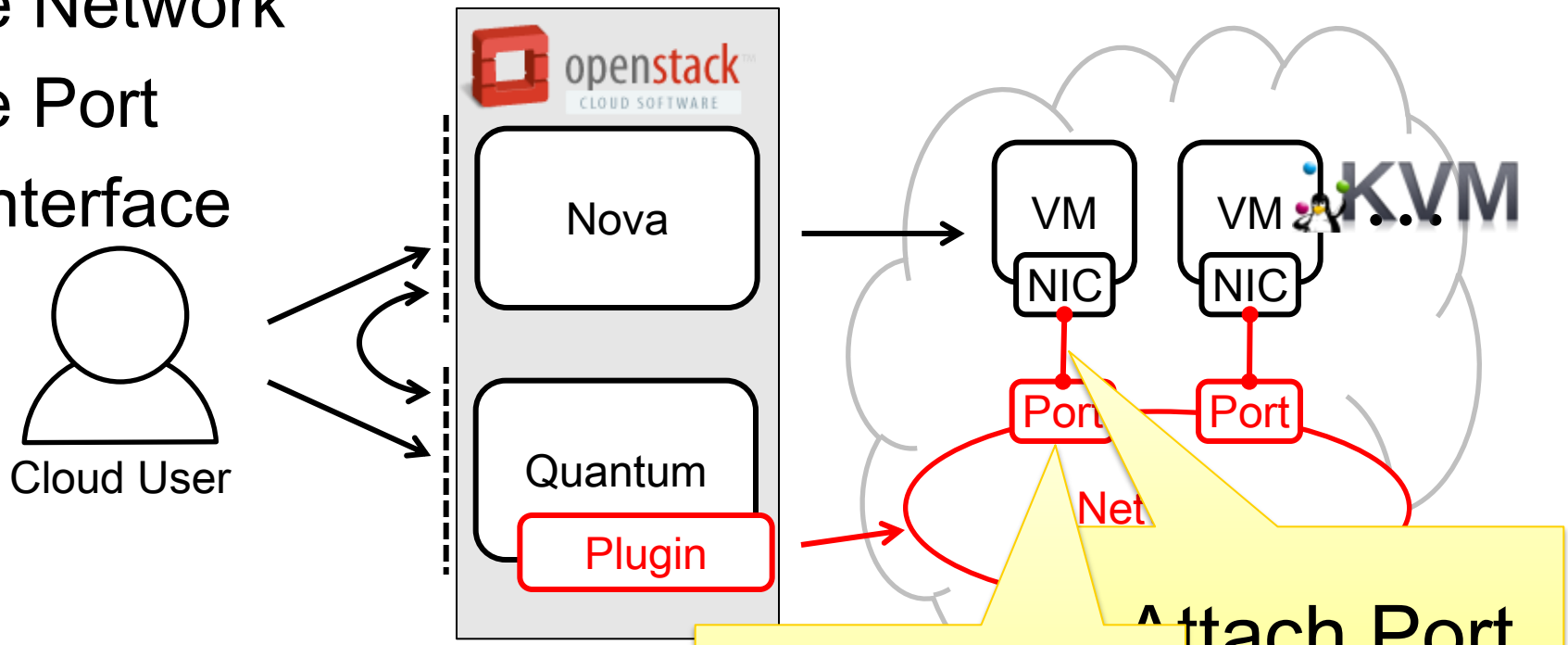
Quantum is an incubated OpenStack project to provide ***"network connectivity as a service"*** between interface devices (e.g., vNICs) managed by other Openstack services (e.g., nova).



"Intro to OpenStack Quantum for Cloud Operators", Dan Wendlandt

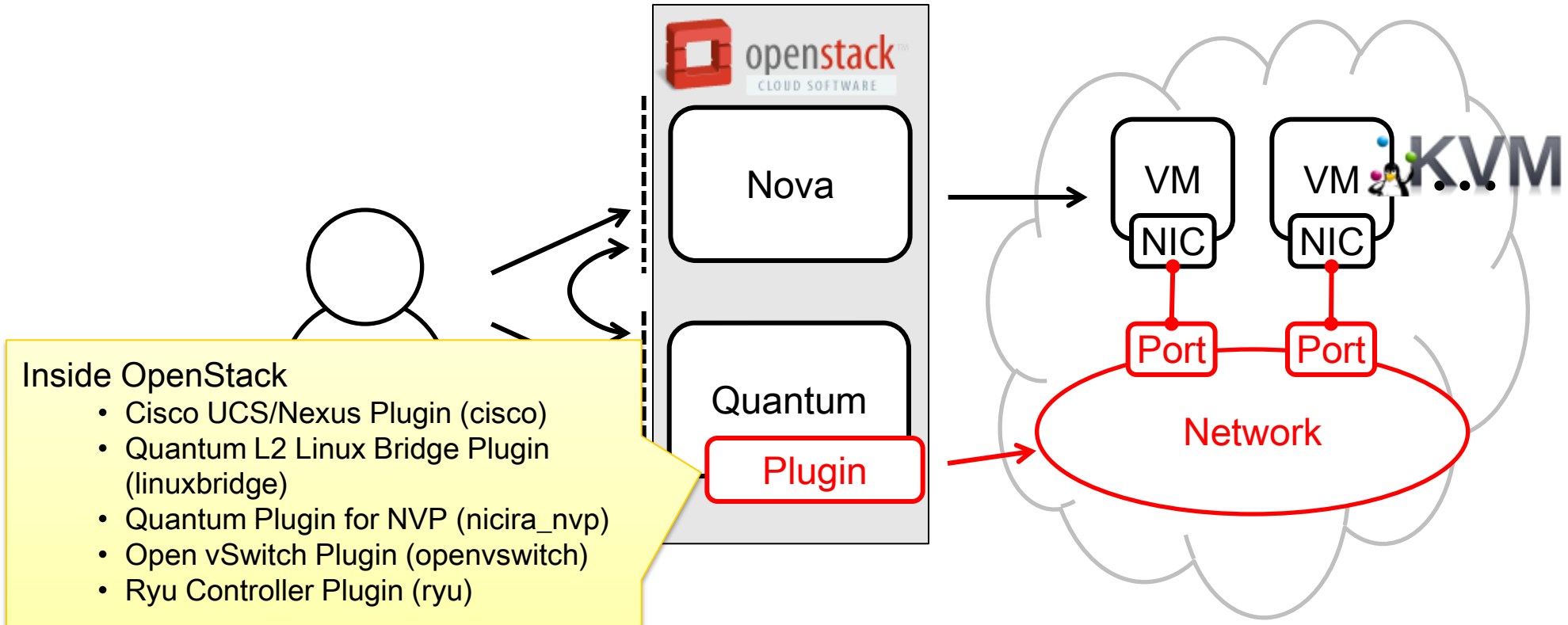
How Quantum works

1. Create Network
2. Create Port
3. Plug Interface



you can create network on
logical network and
ecify

Quantum Plugins



Inside OpenStack

- Cisco UCS/Nexus Plugin (cisco)
- Quantum L2 Linux Bridge Plugin (linuxbridge)
- Quantum Plugin for NVP (nicira_nvp)
- Open vSwitch Plugin (openvswitch)
- Ryu Controller Plugin (ryu)

Blueprints

- NEC OpenFlow Plugin
- Quantum plugin for bigswitch controller
- Layer 2 Network driver for Brocade's Ethernet Fabric Network OS
- A Plugin for Infiniband Infrastructure

INTRO - WHAT IS OPENFLOW/SDN

Question: How old is the Internet?

Answer: 40 years old!

- TCP/IP borned 1970@DARPA
- World Wide Web borned 1989

TCP/IP is long life technology

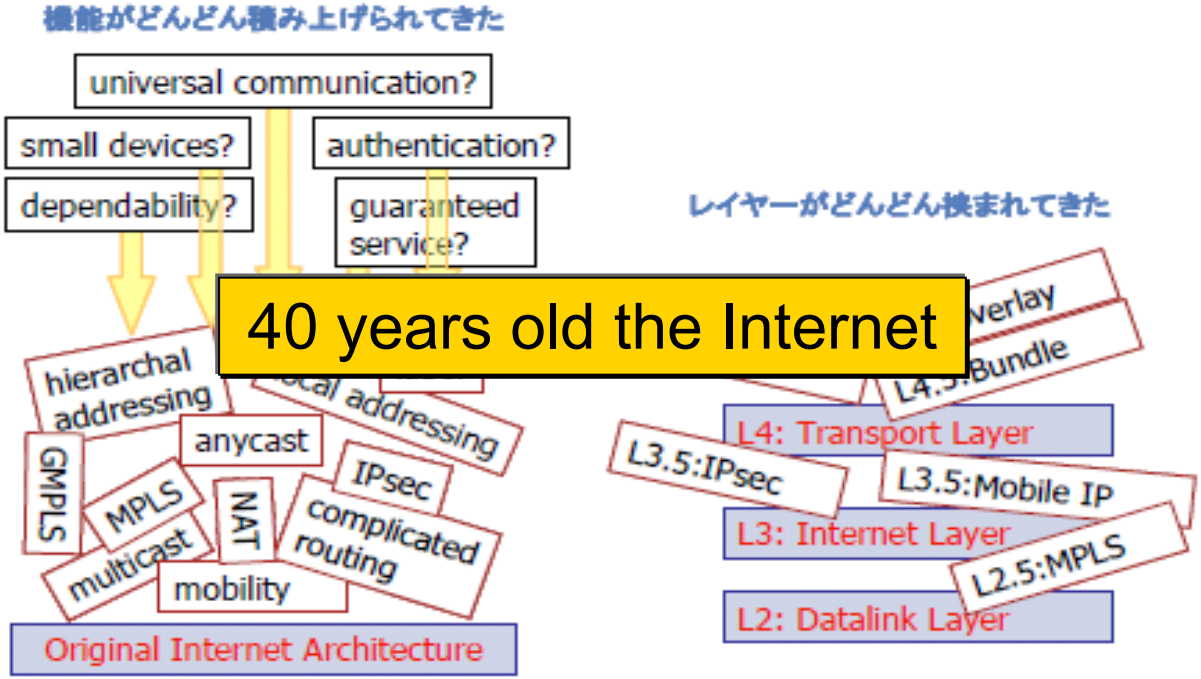
But, usage of the Internet has chaged in this 40 years...

- Telephone by the Internet
- Watching TV by the Internet
- Shopping, trading, chatting, xxing, xxxing, xxxxxing...

Current Internet

New application

Gap and Inconvinient



Future Internet

What is the Internet can not do?

- PC : new idea or application can do by written software. Innovation!
- The Internet: new functions will be implemented next renewal. Please wait 10 years... No Innovation!

How to make innovative technology in the Internet?

- Several project have started about 2007.
- GENE@USA, FP7@EU, ...
- OpenFlow born in Stanford Univ.

Keywords

OpenFlow

- New architecture of network switching
- Network virtualization and programmability

Network virtualization

- You can create “my network”

Programmability

- You can control network by application program

“So, what you want to do?”

Page 10

Background of OpenFlow/SDN

- 2007: Stanford started “Clean Slate Program”
- 2009: Stanford established “Clean Slate Laboratory”
  - Contributed to OpenFlow Consortium to specify OpenFlow spec(v0.8.9, v1.0) and campus trial
  - http://www.openflow.org

- Mar.2011: Open Networking Foundation Founded

Industry standard

NEC active from day #1

- May.2012: Open Networking Research Center (ONRC) established

Industry Open Source activity

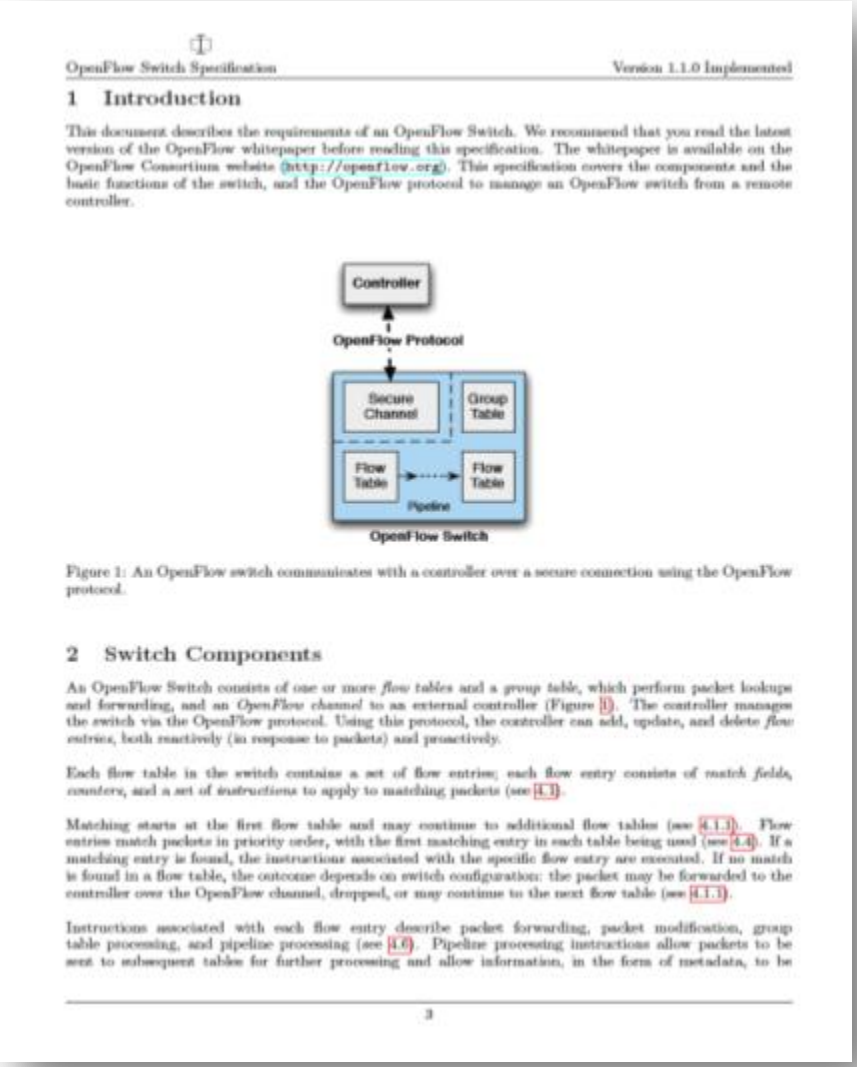
Open Networking Research Center
at Stanford University

Open Networking Foundation

- Established for promotion of Software Defined Networking
- Definition of OpenFlow protocol

OpenFlow Protocol Standard

- OpenFlow Switch Specification
  - 1.0 (2010/3)
    - Mostly used version
  - 1.1 (2011/2)
    - MPLS shim header, multiple table, etc
  - 1.2 (2011/12)
    - IPv6, etc
  - 1.3 (2012/4)
    - PBB, etc
- OF-Config
  - 1.0 (2012/1)
  - 1.1 (2012?)
- OF-Test
  - 1.0 (2012?)



OpenFlow Switch Specification — Version 1.1.0 Implemented

1 Introduction

This document describes the requirements of an OpenFlow Switch. We recommend that you read the latest version of the OpenFlow whitepaper before reading this specification. The whitepaper is available on the OpenFlow Consortium website (http://openflow.org). This specification covers the components and the basic functions of the switch, and the OpenFlow protocol to manage an OpenFlow switch from a remote controller.

Figure 1: An OpenFlow switch communicates with a controller over a secure connection using the OpenFlow protocol.

2 Switch Components

An OpenFlow Switch consists of one or more *flow tables* and a *group table*, which perform packet lookups and forwarding, and an *OpenFlow channel* to an external controller (Figure 1). The controller manages the switch via the OpenFlow protocol. Using this protocol, the controller can add, update, and delete *flow entries*, both reactively (in response to packets) and proactively.

Each flow table in the switch contains a set of flow entries; each flow entry consists of *match fields*, *counters*, and a set of *instructions* to apply to matching packets (see 4.1).

Matching starts at the first flow table and may continue to additional flow tables (see 4.1.1). Flow entries match packets in priority order, with the first matching entry in each table being used (see 4.4). If a matching entry is found, the instructions associated with the specific flow entry are executed. If no match is found in a flow table, the outcome depends on switch configuration: the packet may be forwarded to the controller over the OpenFlow channel, dropped, or may continue to the next flow table (see 4.1.1).

Instructions associated with each flow entry describe packet forwarding, packet modification, group table processing, and pipeline processing (see 4.6). Pipeline processing instructions allow packets to be sent to subsequent tables for further processing and allow information, in the form of metadata, to be

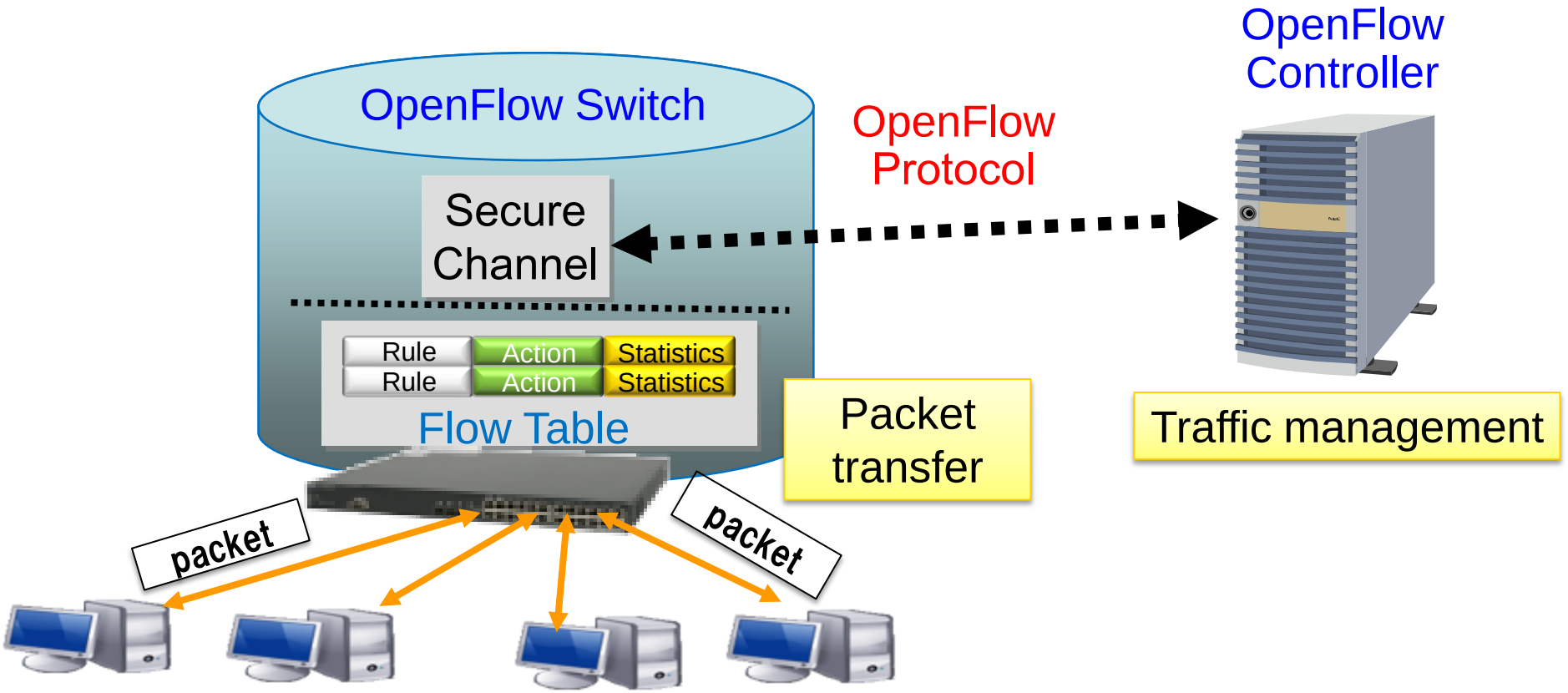
3

http://www.openflow.org/documents/openflow-spec-v1.1.0.pdf

OPENFLOW BASICS

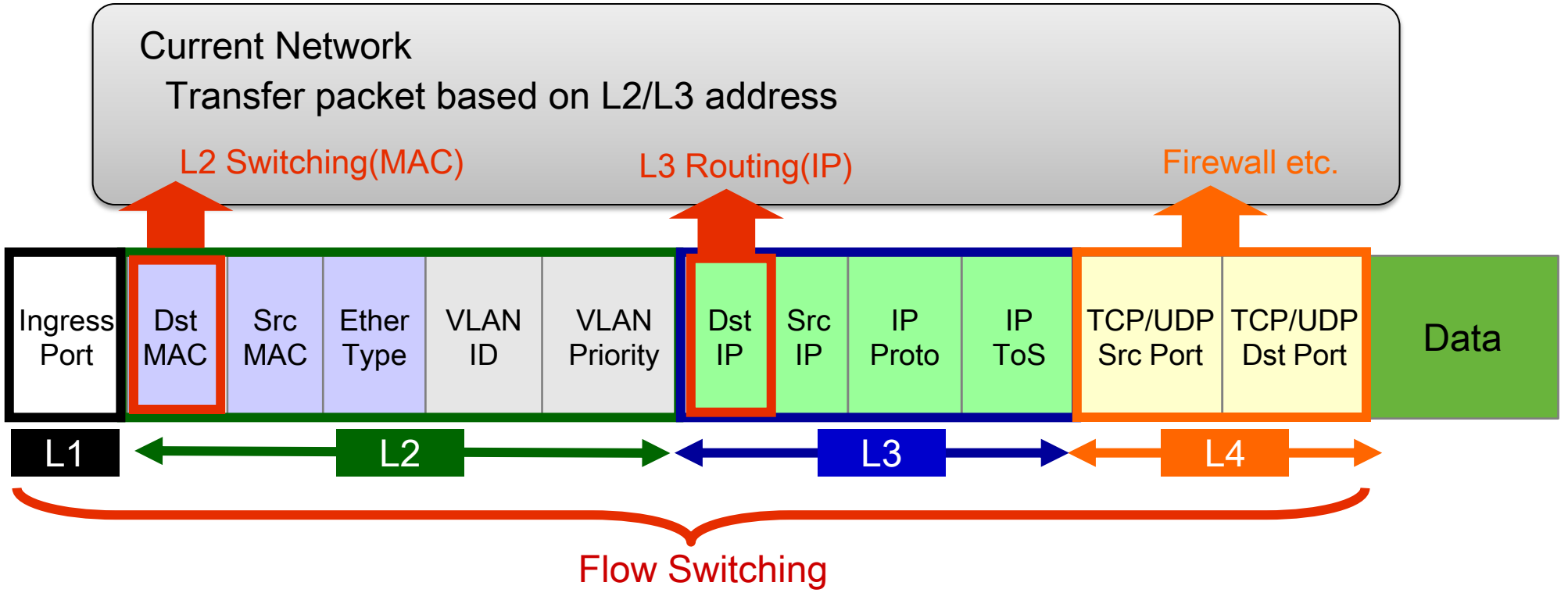
OpenFlow Basics: Architecture

- Separate Data Plane and Control Plane
- OpenFlow is the protocol between switch and controller
- L1-L4 field are used for switching



OpenFlow Basics: Flow Switching

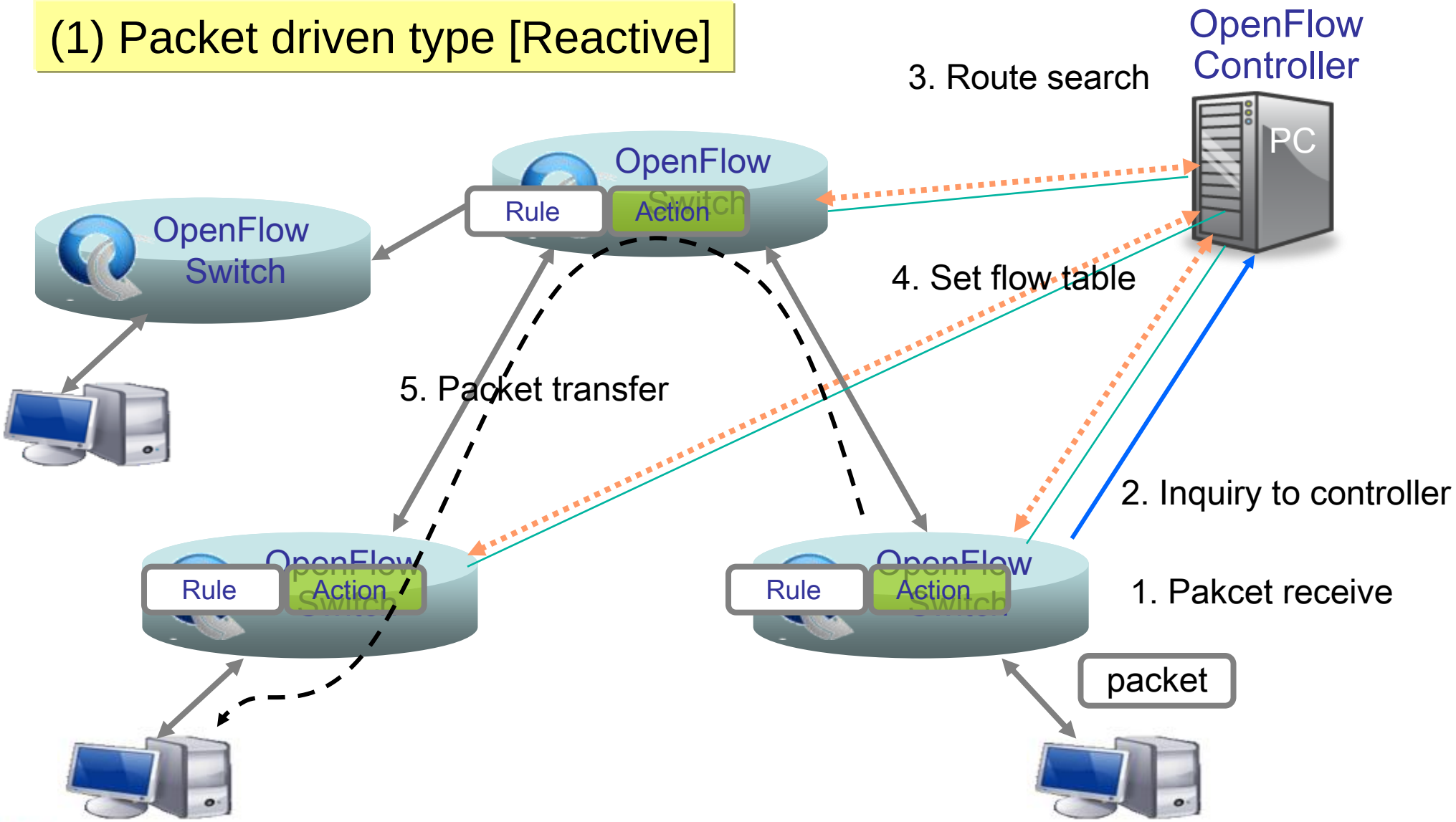
Transfer packet based on “Flow”



Flow is distinguished by rule of combination through L1(port), L2(MAC), L3(IP), L4(port).
Transferring method that use flow is called flow switching.

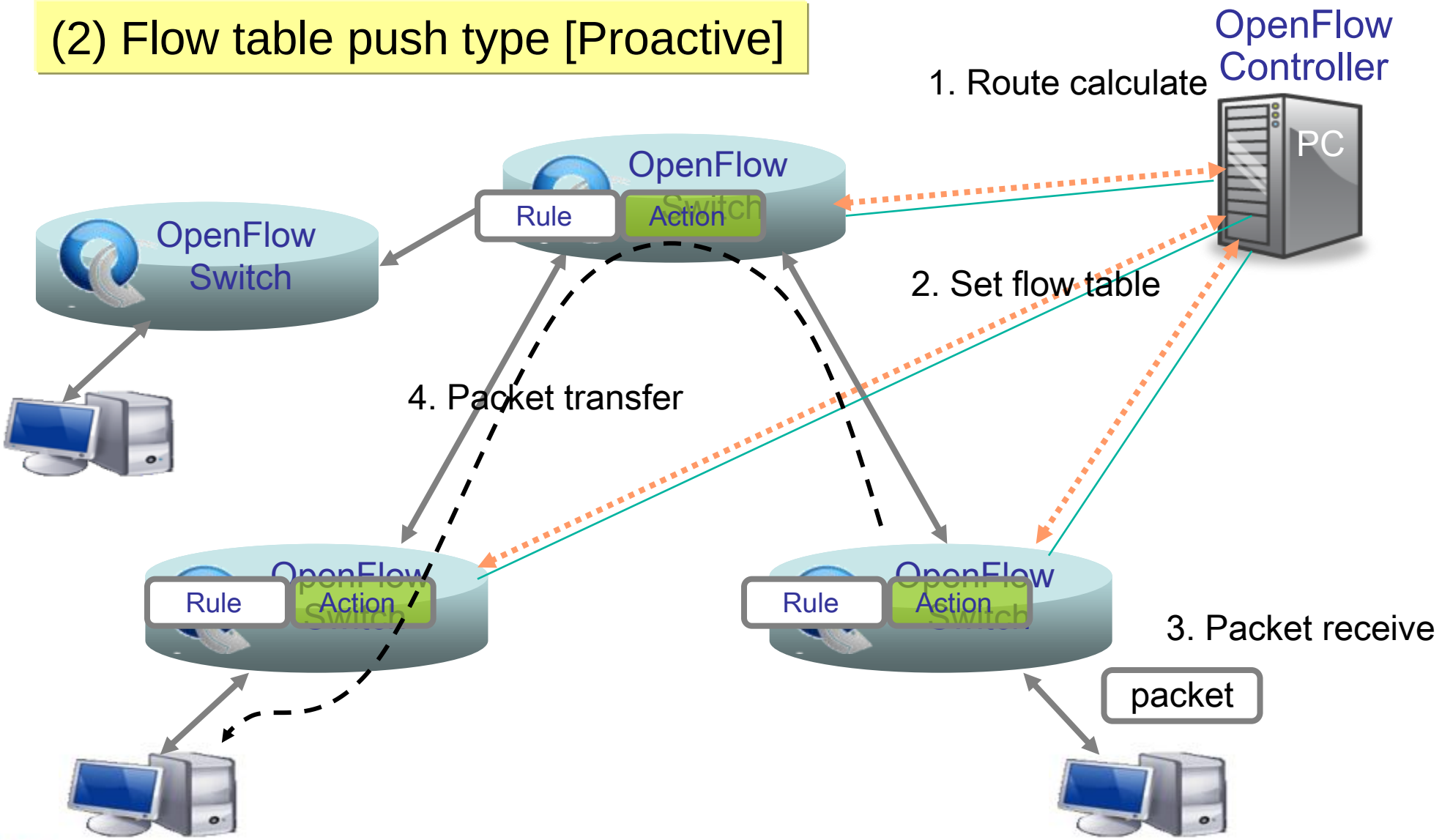
How OpenFlow works

(1) Packet driven type [Reactive]

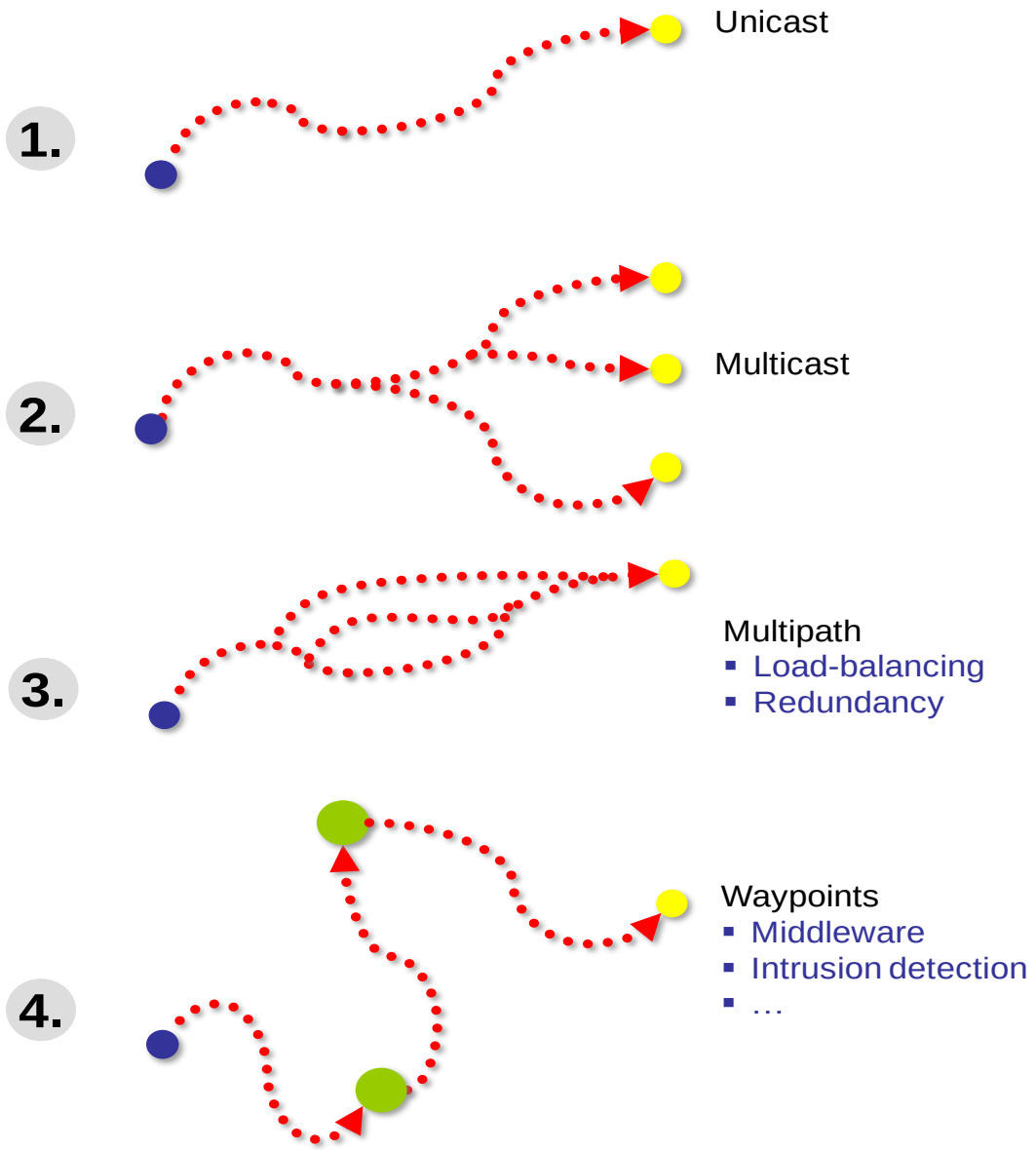


How OpenFlow works

(2) Flow table push type [Proactive]



Flow examples



OpenFlow Protocol detail

- Protocol between OpenFlow Switch and OpenFlow Controller

- Messages
- Flow table
- Match
- Action

OpenFlow Messages

Packet

- Packet in : switch to controller
- Packet out : controller to switch

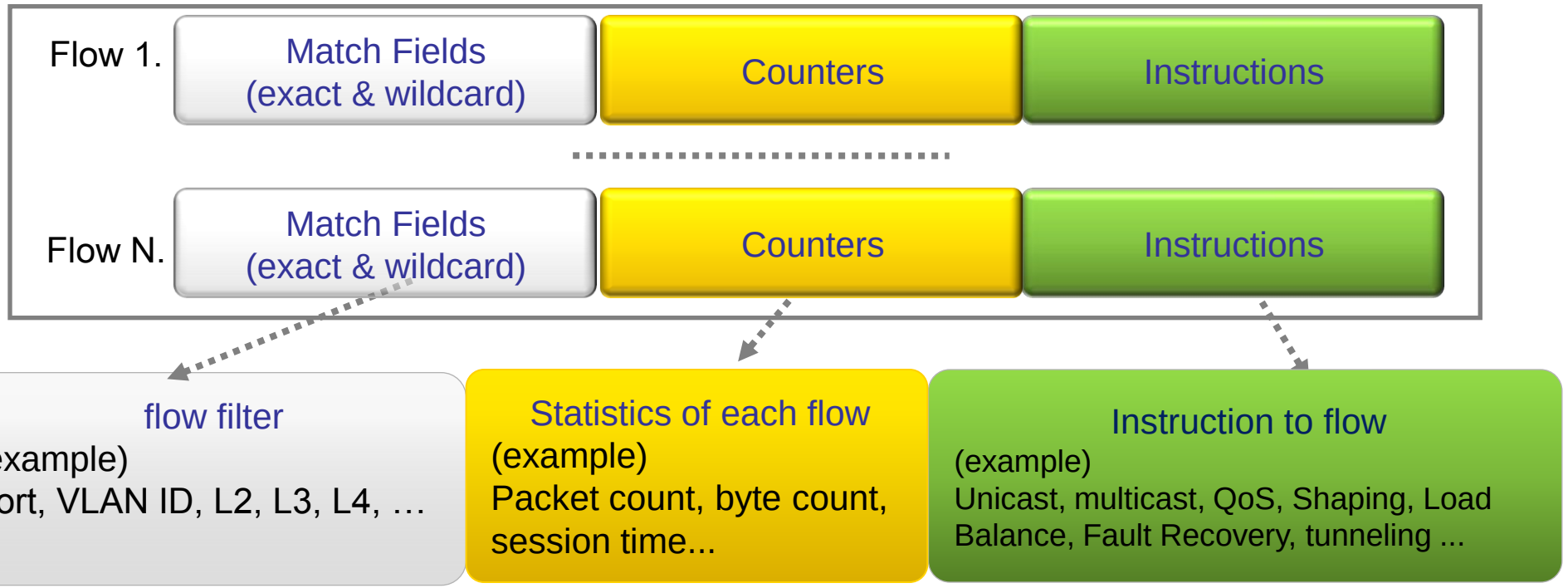
Flow entry

- Flow mod : controller to switch
- Flow removed : switch to controller (expire)

Management

- Port status : switch to controller (port status change notify)
- Echo request/reply
- Features request/reply
- …

Flow Table Difinition



Matching Filter

- Ingress port
- Ethernet source/destination address
- Ethernet type
- VLAN ID
- VLAN priority
- IPv4 source/destination address
- IPv4 protocol number
- IPv4 type of service
- TCP/UDP source/destination port
- ICMP type/code

12 tuple through L1 to L4 header field can be used

Action

- Forward
  - Physical ports (Required)
  - Virtual ports : All, Controller, Local, Table, IN_PORT (Required)
  - Virtual ports : Normal, Flood (Required)
- Enqueue (Optional)
- Drop (Required)
- Modify Field (Optional)
  - Set/Add VLAN ID
  - Set VLAN priority
  - Strip VLAN Header
  - Modify Ethernet source/destination address
  - Modify IPv4 source/destionation address
  - Modify IPv4 type of service bits
  - Modify IPv4 TCP/UDP source/destination port

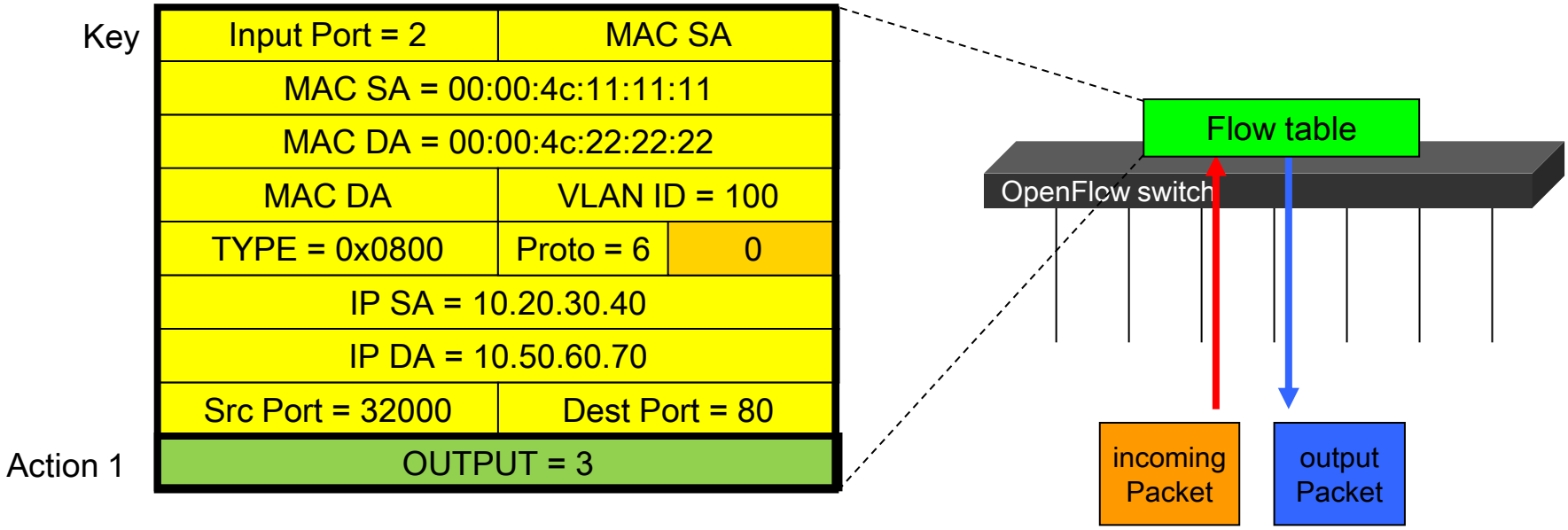
Various type of transfering rules

Possible to modify header

Possible to set multi actions

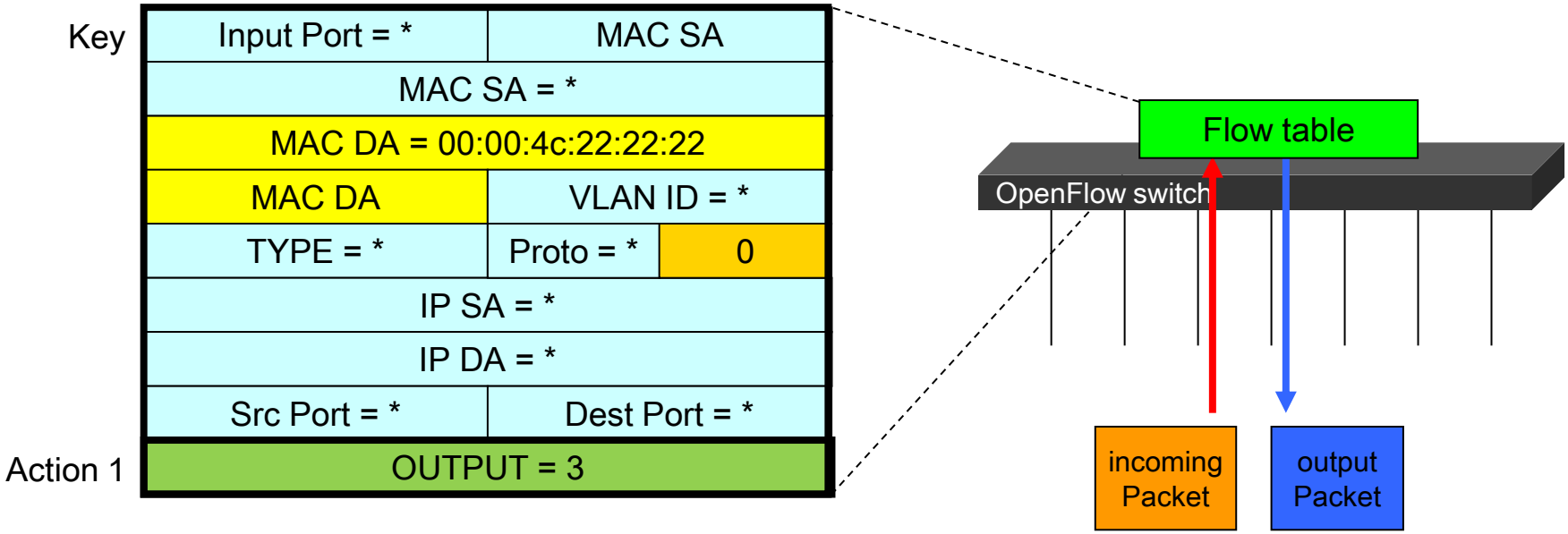
Example of flow table

Flow switching



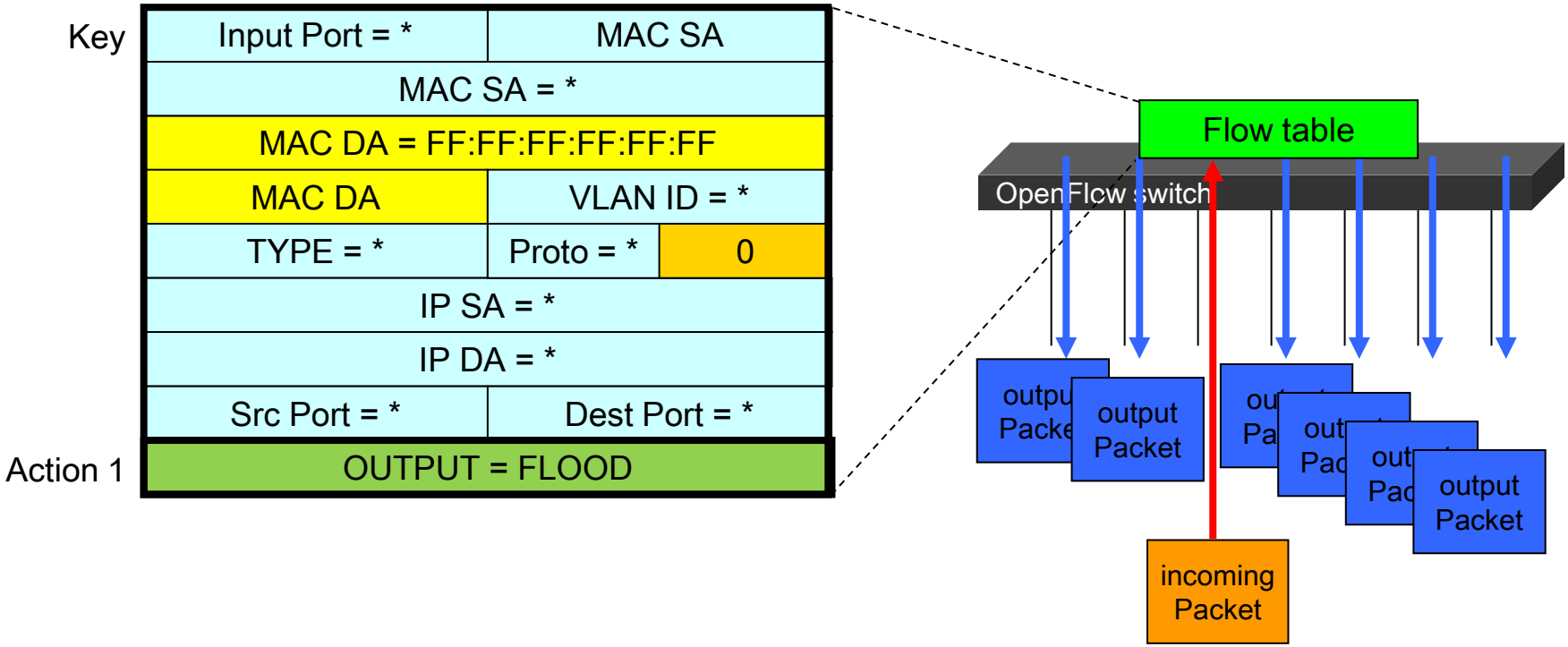
Example of flow table

L2 switching



Example of flow table

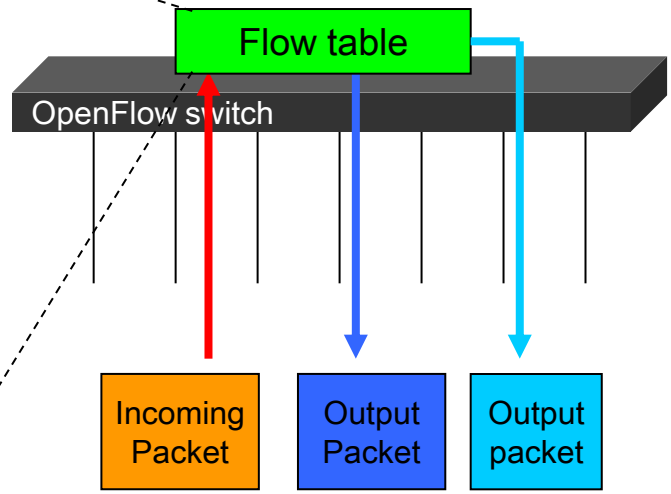
Broadcast



Example of flow table

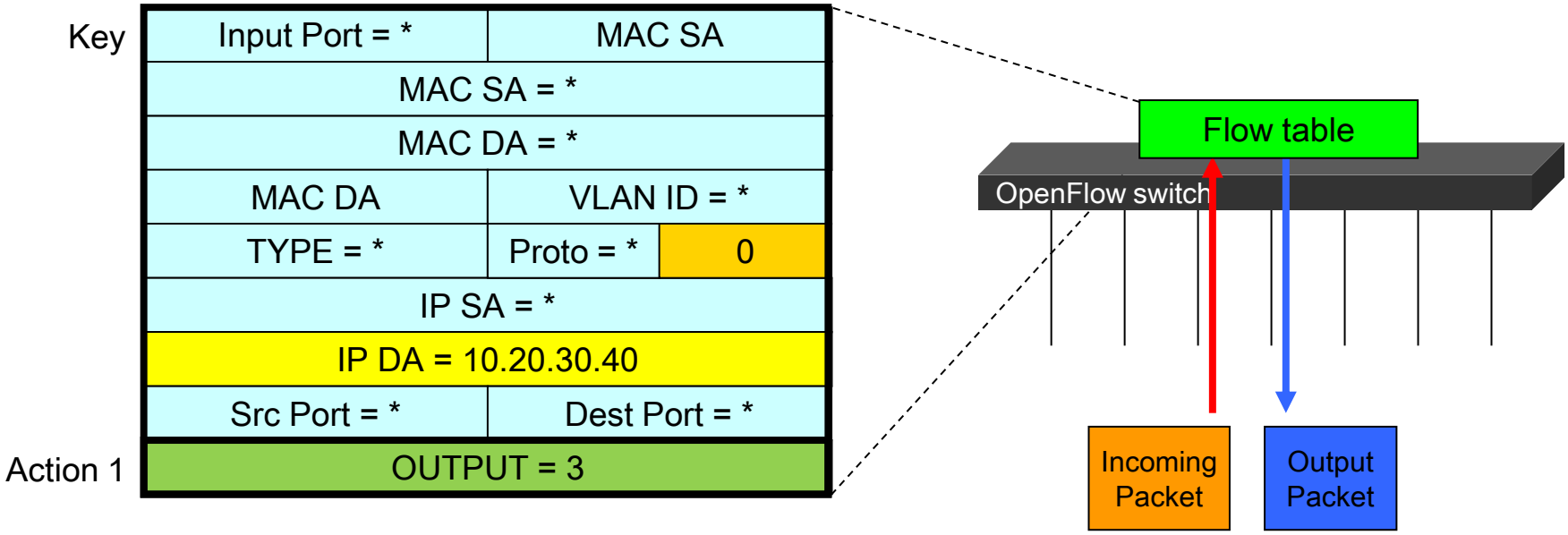
multicast

Key	Input Port = 1	MAC SA	
	MAC SA = *		
	MAC DA = *		
	MAC DA	VLAN ID = *	
	TYPE = 0x0800	Proto = *	0
	IP SA = *		
	IP DA = 224.10.10.1		
	Src Port = *	Dest Port = *	
Action 1	SET_DL_SRC = 02:00:4c:00:00:01		
Action 2	OUTPUT = 3		
Action 3	SET_DL_SRC = 02:00:4c:00:00:02		
Action 4	OUTPUT = 5		

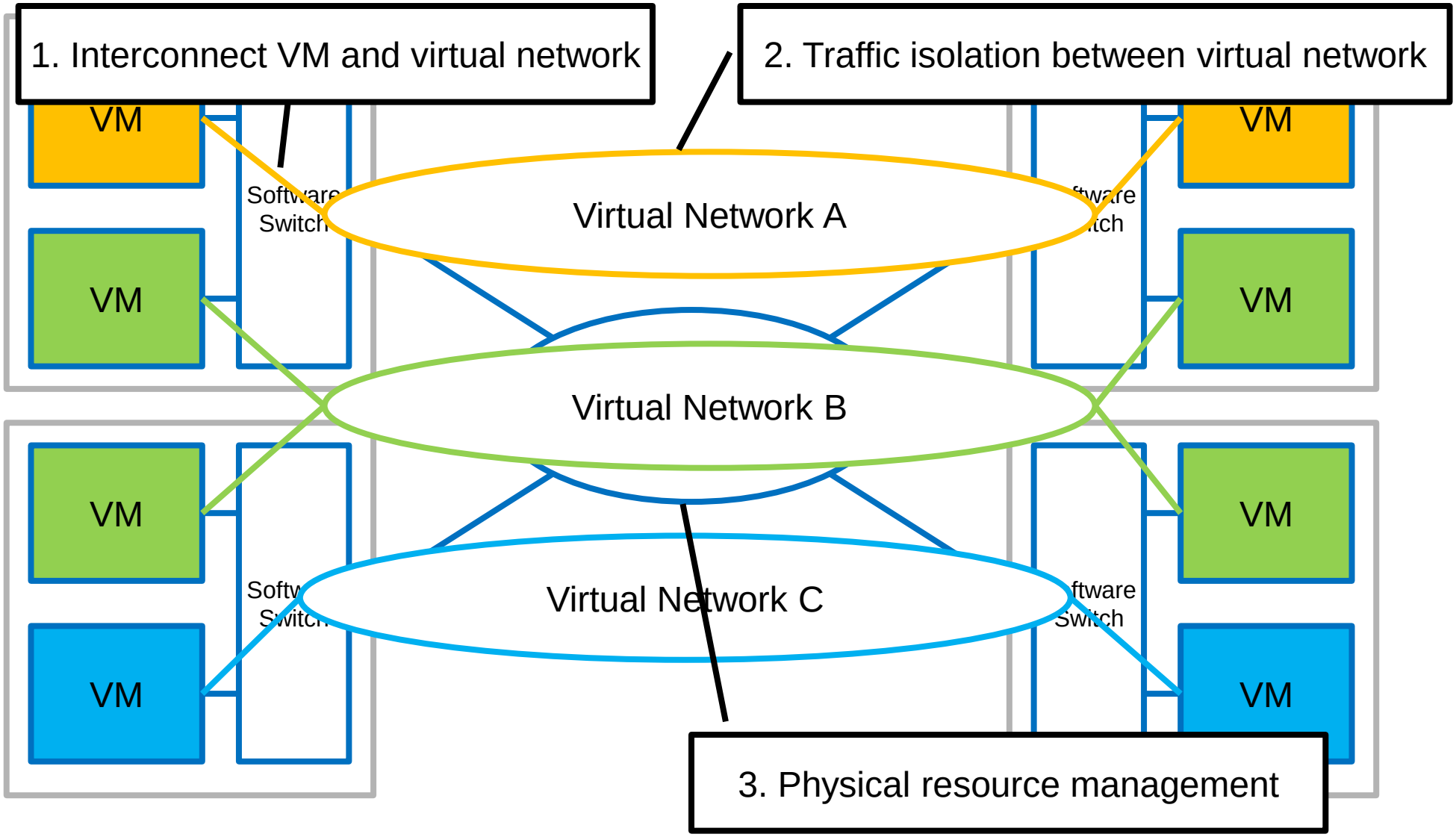


Example of flow table

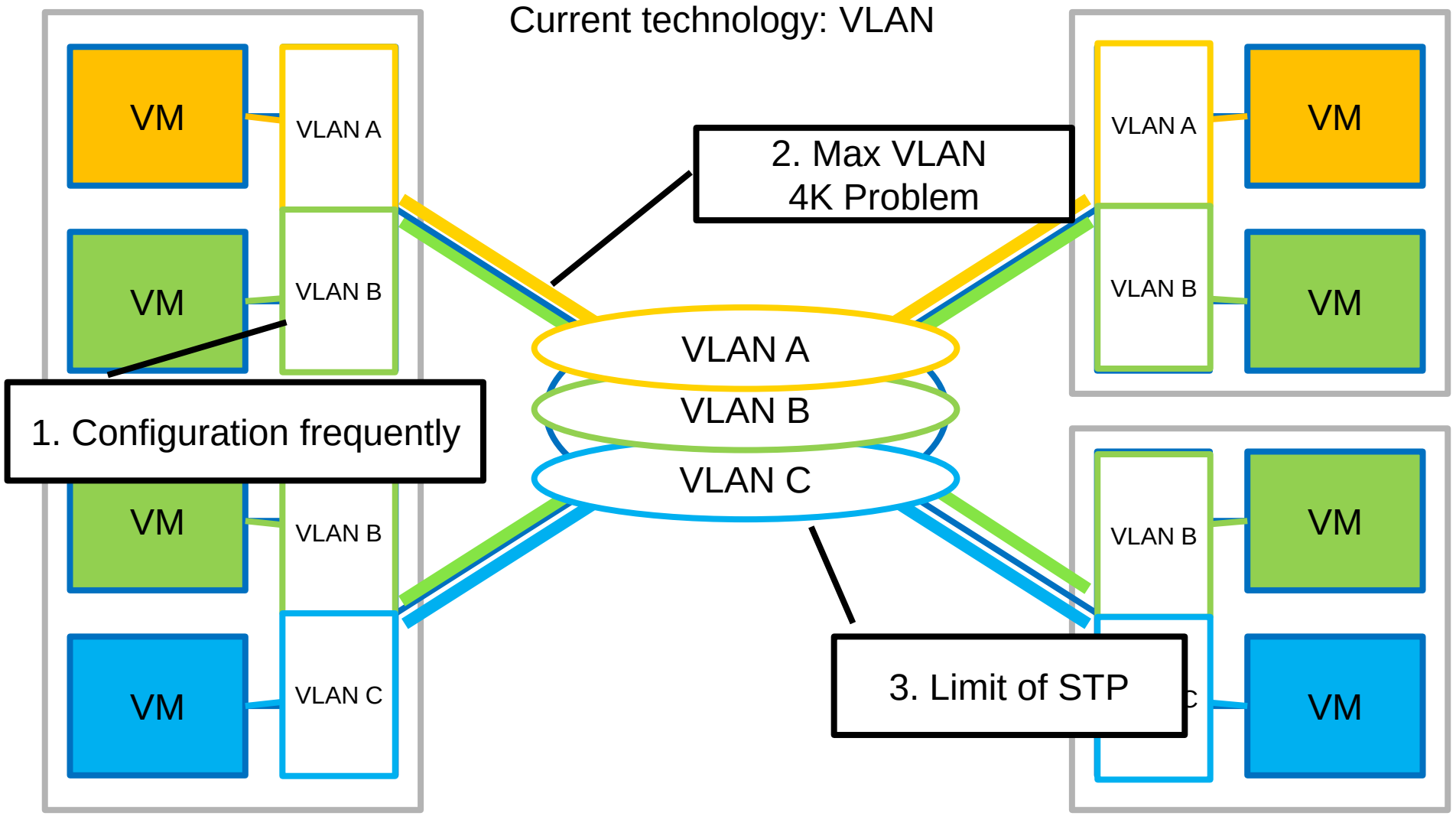
IP Routing



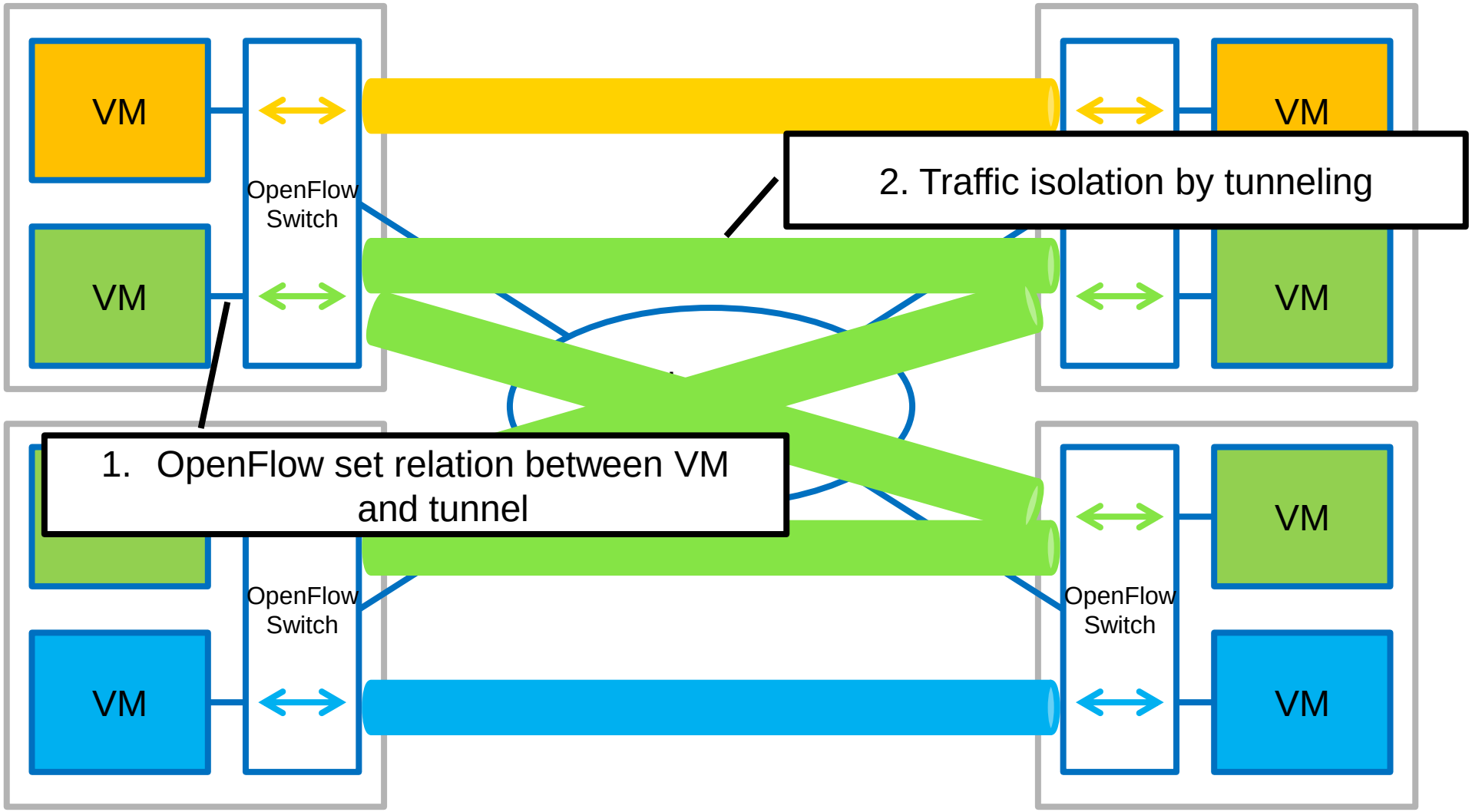
Network virtualization in Data Center



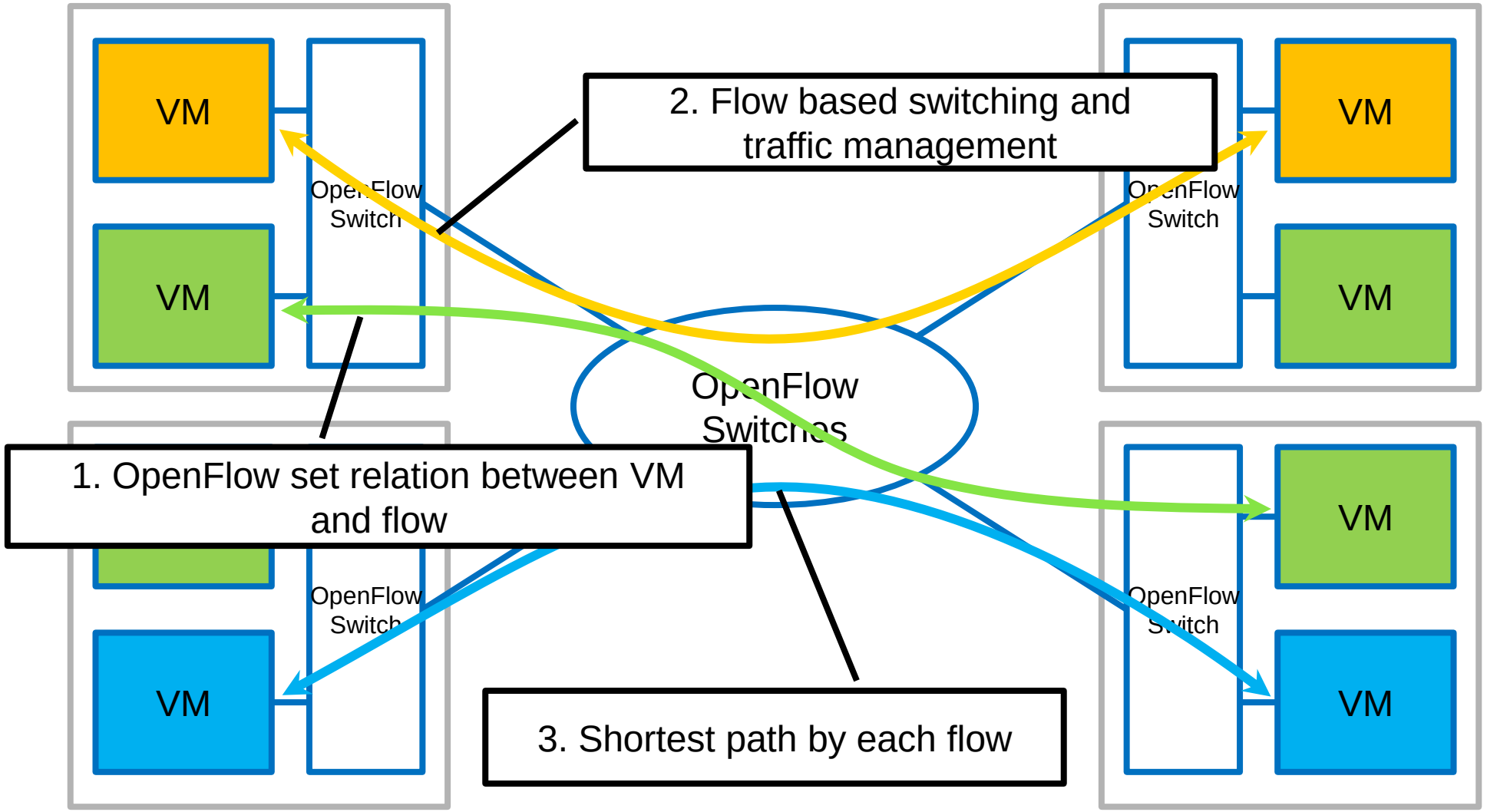
VLAN


Current technology: VLAN
VM
VLAN A
VM
VLAN B
VM
VLAN B
VM
VLAN C
VLAN A
VM
VLAN B
VM
VLAN B
VM
VM
VLAN A
VLAN B
VLAN C
2. Max VLAN
4K Problem
1. Configuration frequently
3. Limit of STP

OpenFlow with Overlay type



OpenFlow with Hop-by-Hop type



Hop-by-Hop implement example – Trema/SliceableSwitch

3. Check the ingress and egress belong to same slice.

slice

2. Search flow table by DMAC to get egress switch and port

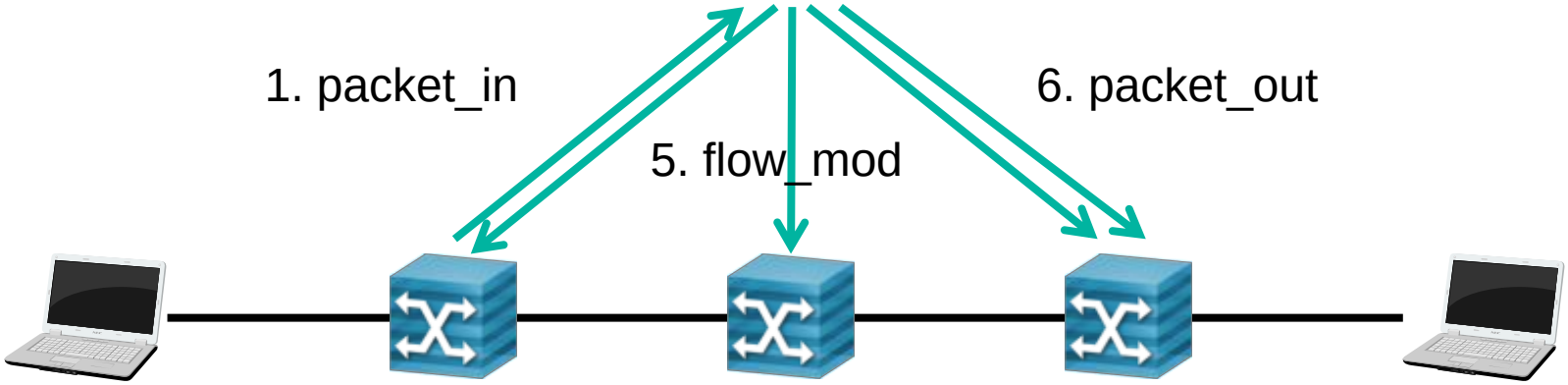
4. Calculate the shortest path

fdb

Sliceable Switch

Trema

topology



Controller dicide the shortest path by each flow and construct the total path.

Variation of OpenFlow switch implementation

L2 switch base

- Firmware update with commodity L2 switch
- Instruct L2 through L4 header
- NEC, HP, Juniper, Quanta, Arista ...

Software switch

- Implement OpenFlow switch inside host
- Open vSwitch, Stanford Reference Switch ...

Transport node base

- Path can be configured by OpenFlow protocol
- Cienna, Fujitsu America ...

Wireless

- WiFi-AP or WiMAX-BS/ASNGW that can handle OpenFlow

OpenFlow Controller (1/2)

OSS

- NOX
- POX
- SNAC
- Trema
- Beacon,
- Floodlight
- Ryu, Node Flow, Flow ER, Nettle, Mirage, ovs-controller, Maestro

https://events.linuxfoundation.org/images/stories/pdf/lcjp2012_yamahata_openflow.pdf

OpenFlow Controller (2/2)

Products available

- Nicira: NVP Network Virtualization Platform
- BigSwitch: Floodlight based?
- Midokura: Midonet
- NTT Data:
- Travelping: FlowER based
- NEC: ProgrammableFlow

WHY OPENFLOW/SDN?

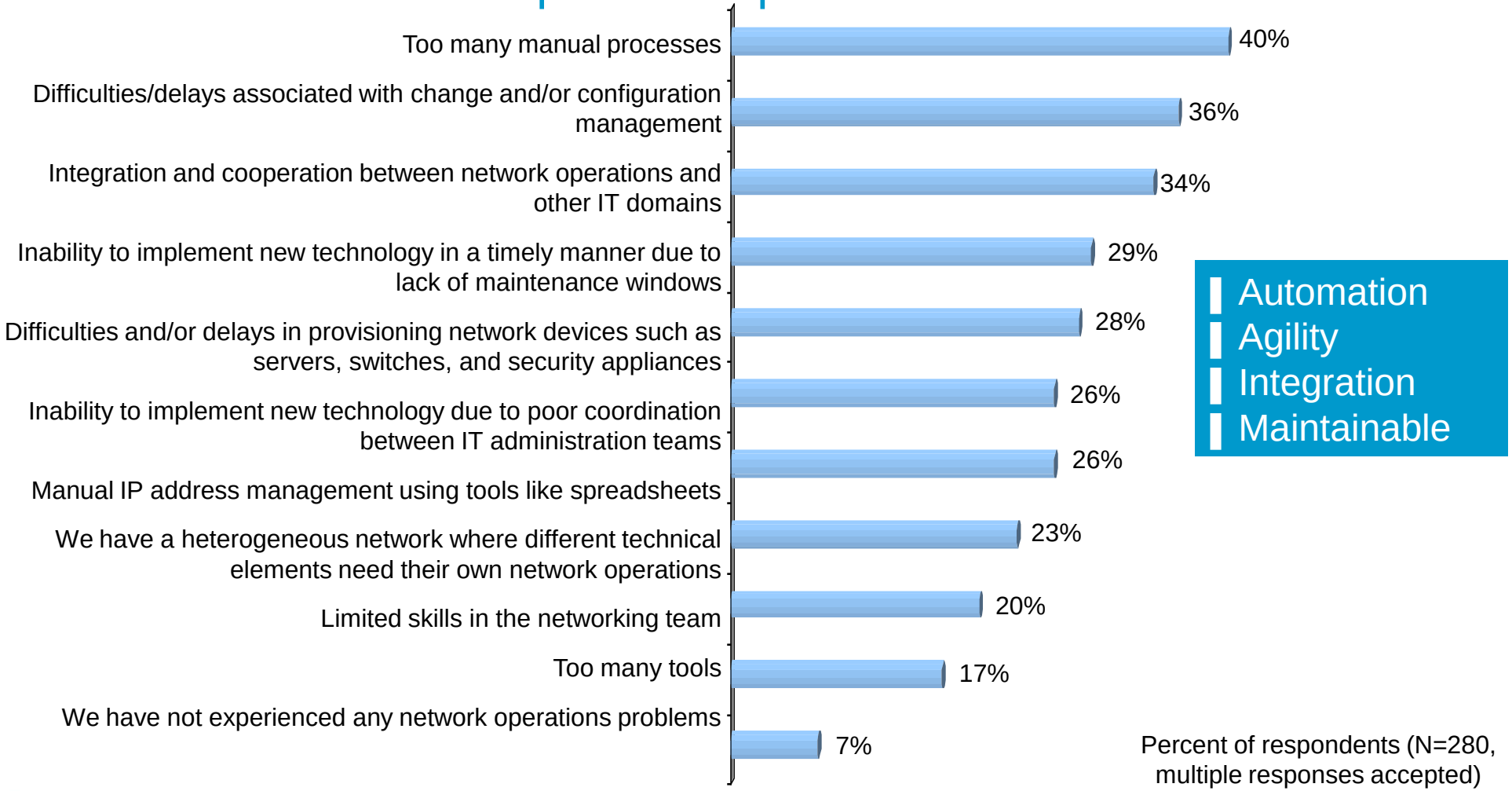
Network is complicated!

"To support a cloud infrastructure, in reality all aspects of the IT infrastructure must be cloud-enabled, including storage, software, and – importantly – networking. Legacy network architectures are designed to support static network configurations and often do not provide the flexibility required to support cloud and data center deployments."

Rohit Mehra, IDC
January 2012

Challenges of Managing a Data Center Network

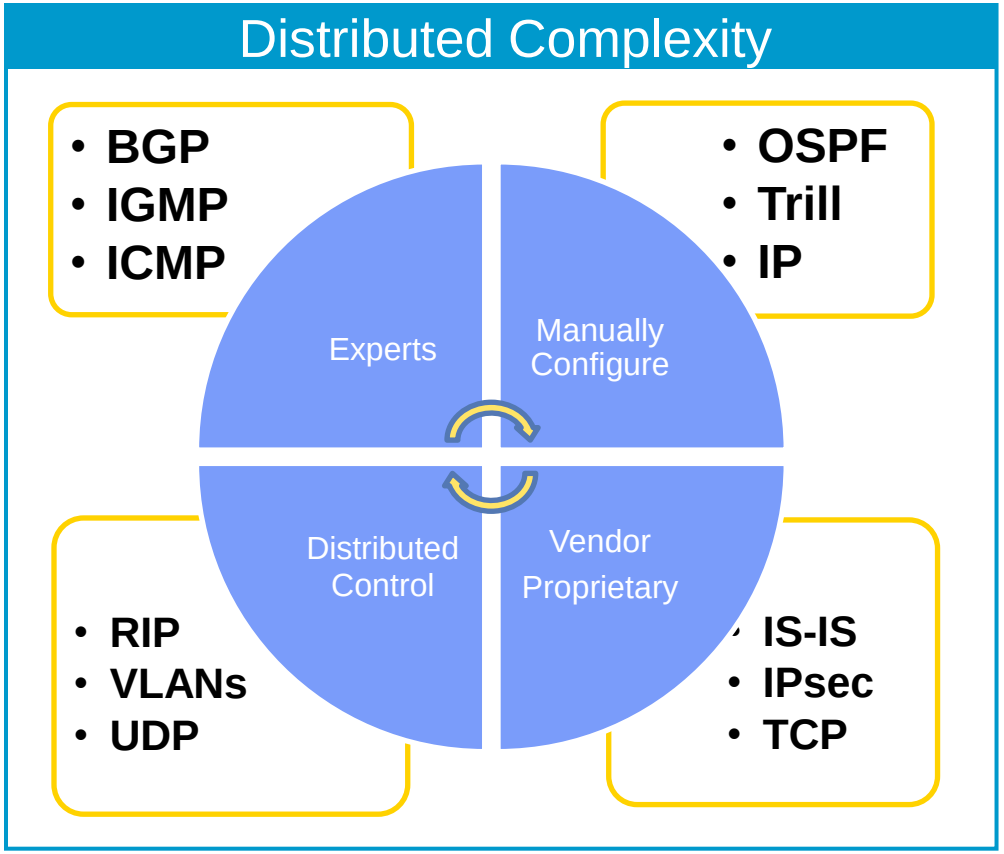
Has your organization experienced any of the following network operations problems?



Percent of respondents (N=280, multiple responses accepted)

Network Not Keeping Pace with Server Virtualization

- Many protocols to solve performance and scalability requirements
- Vendor-specific configurations
- Software rollouts frequent
- Switch-by-switch configuration management



Server virtualization needs an open, agile network

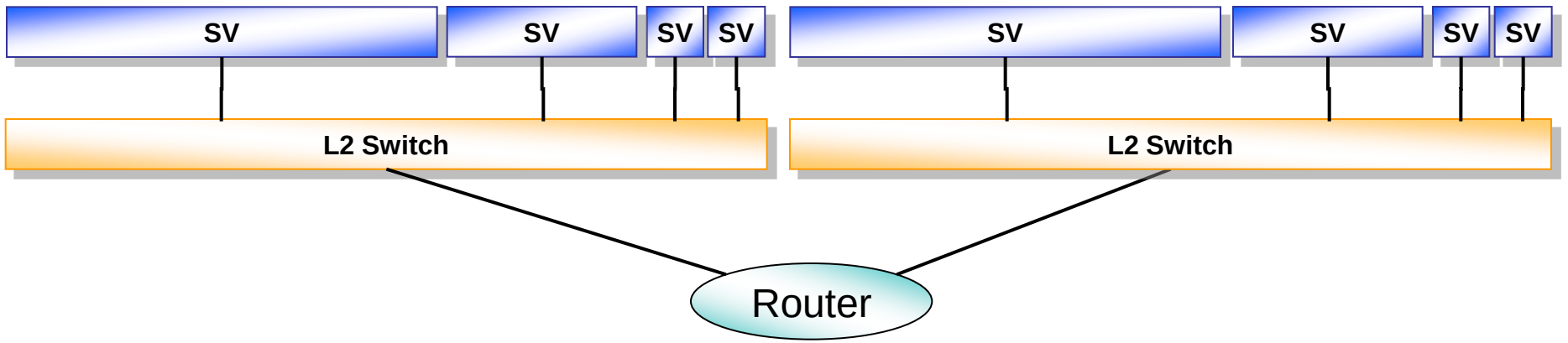
FAQ: Why OpenFlow?

- Benefit ?
- Risk?
- New technology?
- Applications?
- Potential?
- Open?

- Let’s see the history of IT system.

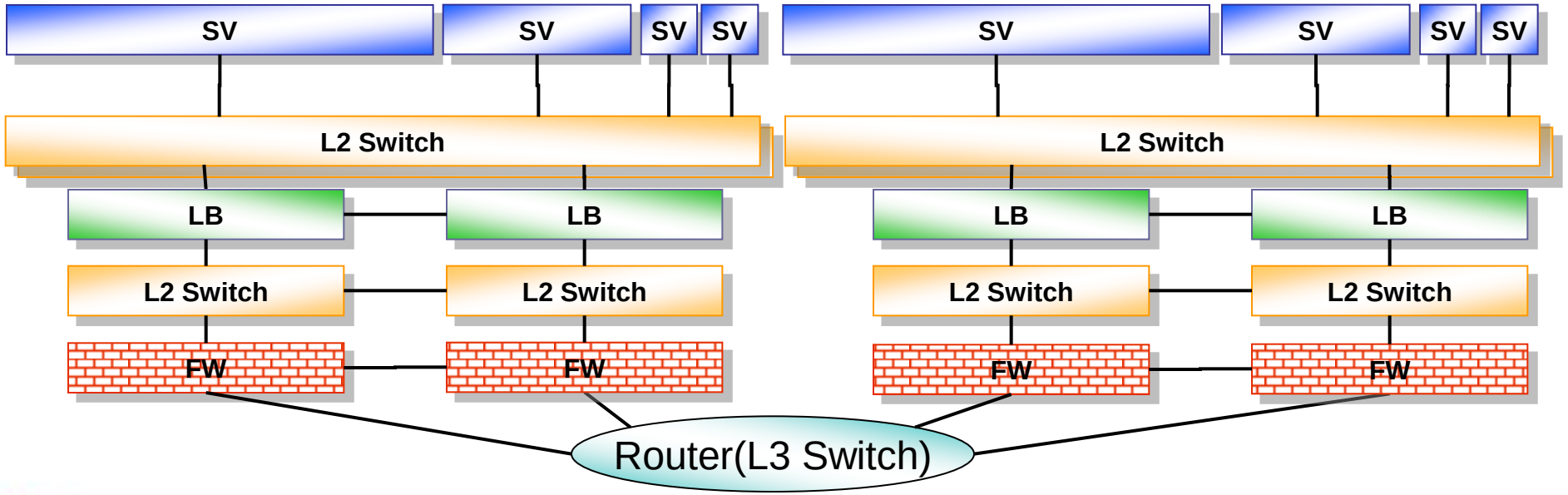
Before Virtualization: BV

- Management of servers and networks are fully separated.
- There are some difficult problem, for example virtual IP address management



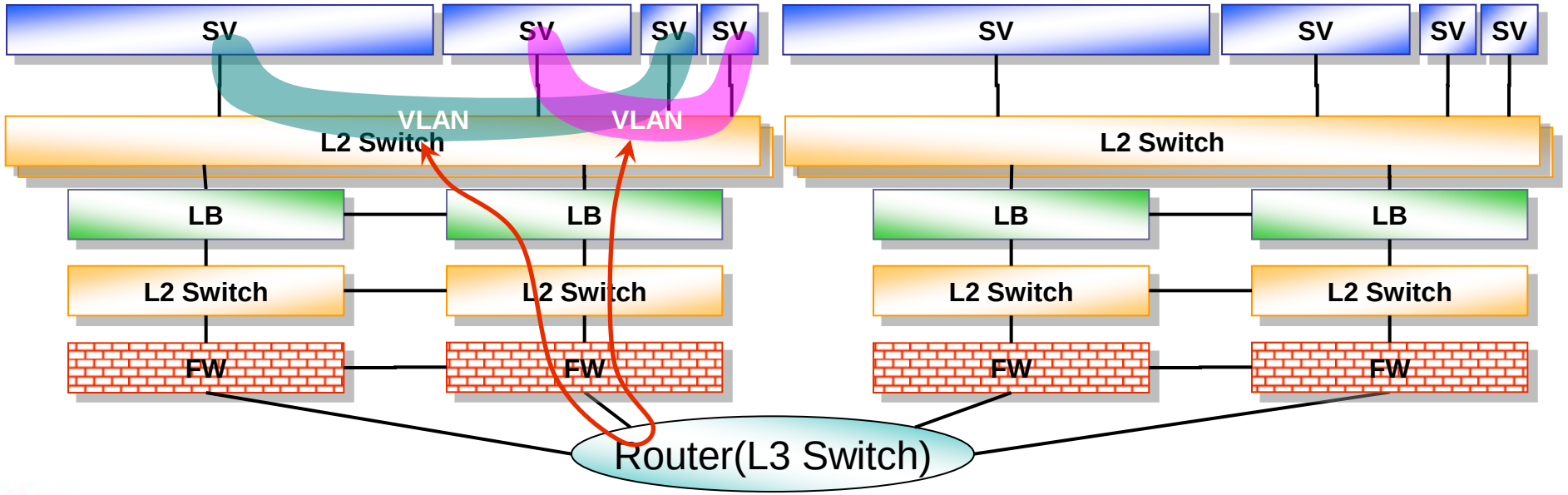
BV: Firewall, LoadBalance

- Switch and routers are redundant
- Firewall or loadbalancer are specialized hardware and clusterd



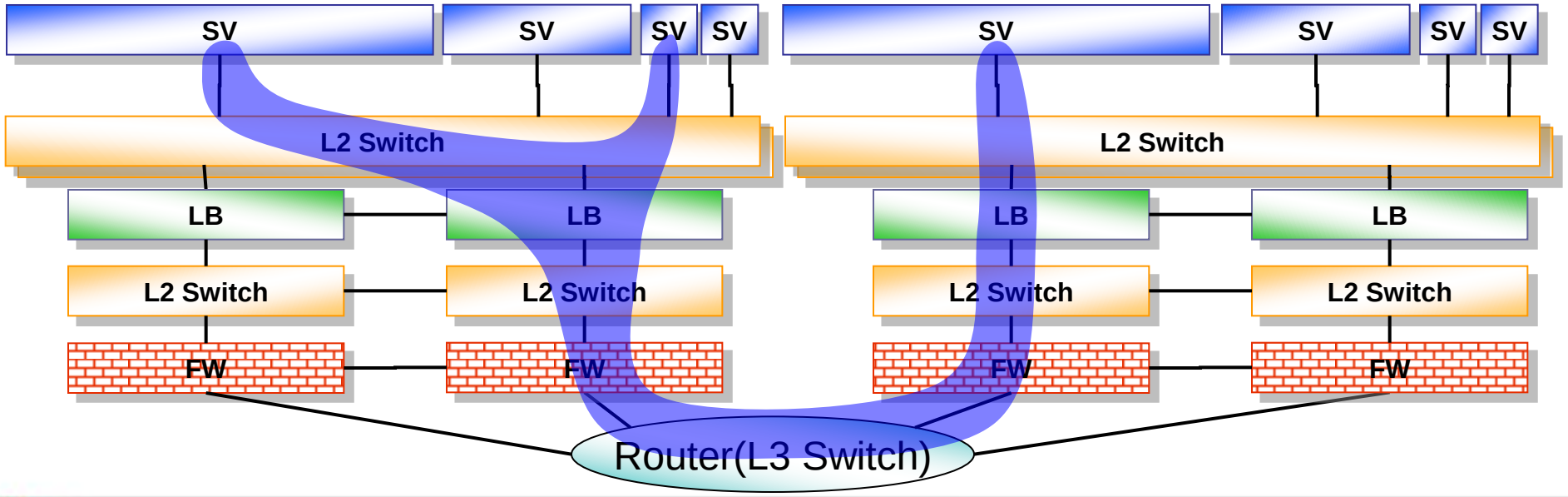
VLAN appears

- VLAN can separate subnet
- number of switch port grows, resouce utilization is improved
- Multiple subnets can consolidate on same switch.
- But routing is interchanged by router.



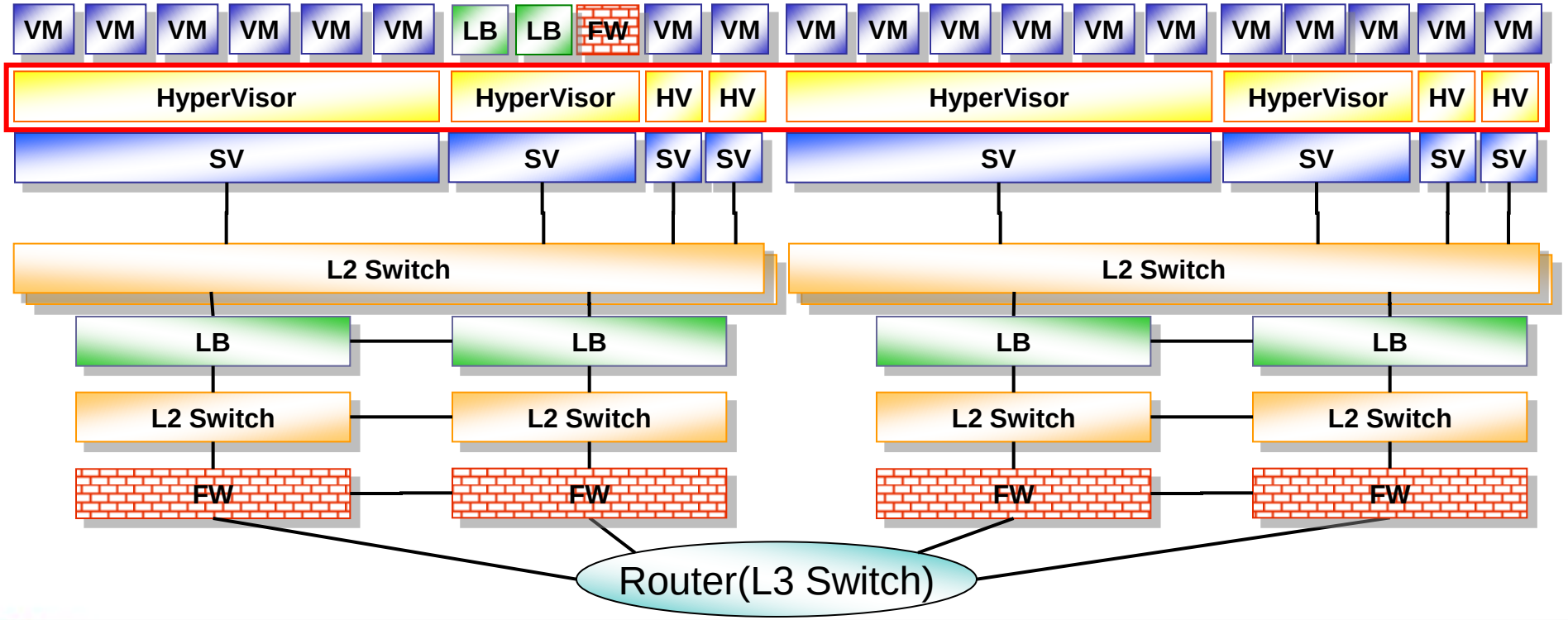
Extension of VLAN

- VLAN can be created crossing router.
- Because subnet can cross router, consistent configurations are necessary



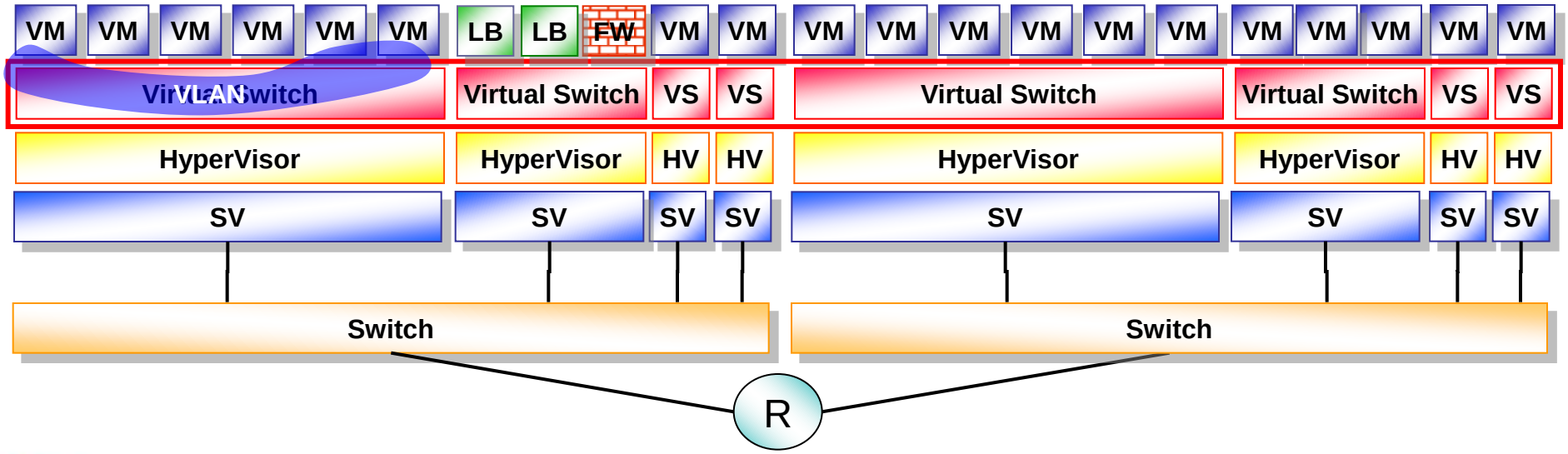
Hypervisor appears

- Virtual machine appears
- It is impossible to manage by human!
- LB or FW can be Virtual Appliance



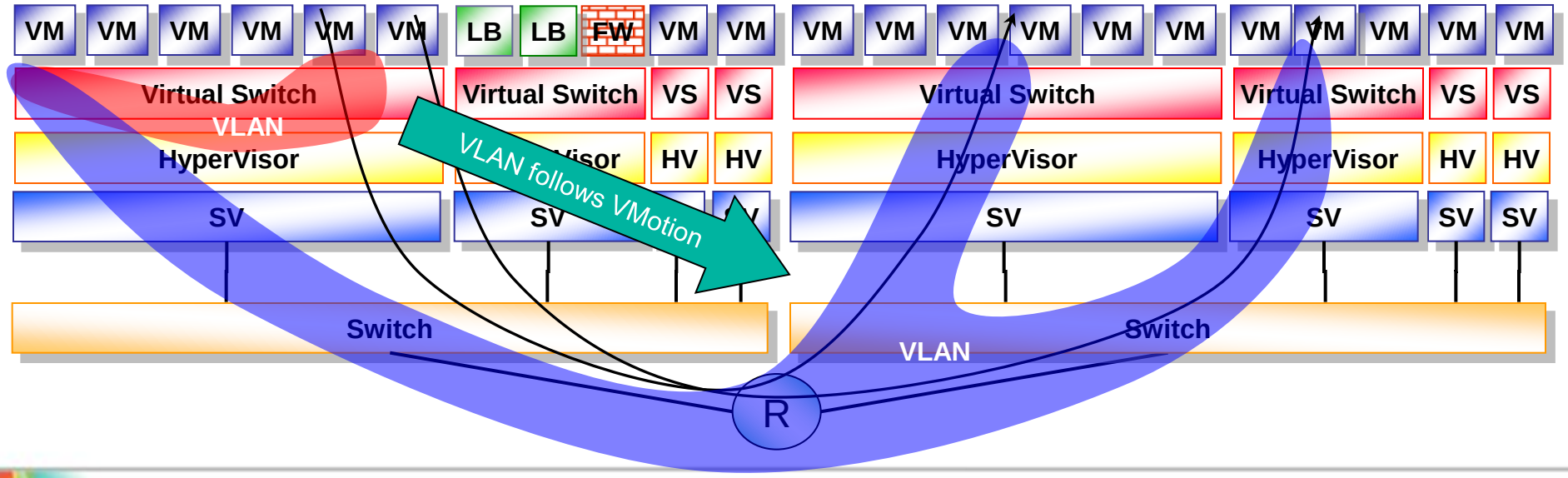
vswitch appears

- trafic inside server can not manageable and complicated
- separation between server and network become cloudy



Live Migration

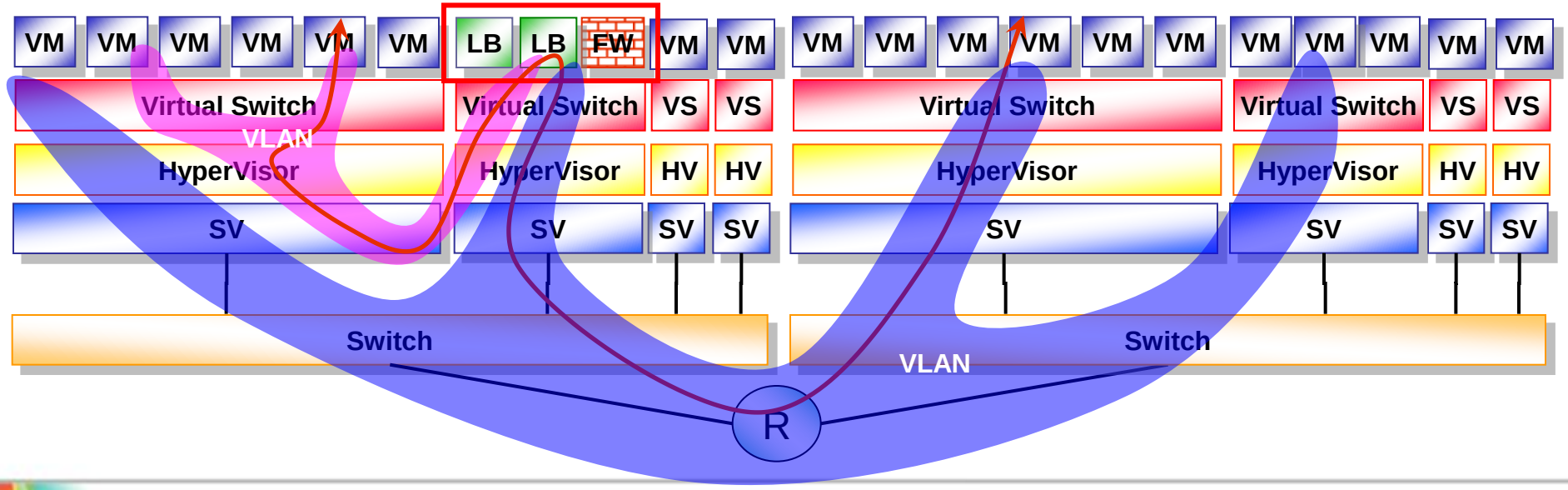
- Live Migration appears, VM can migrate between servers without stop or suspend.
- VLAN must follow VM migration
- Resource management became on-demand, VLAN configuration change is more frequently, inventory control of resources is more important.
- Automation of resource management is desired, not only server resources, but also network resources.



Virtual appliances

- Virtual appliance appears, traffic route is more complicated.
- In past days, Firewall and Load Balancer are managed by network side, but virtual appliances belong to server side.

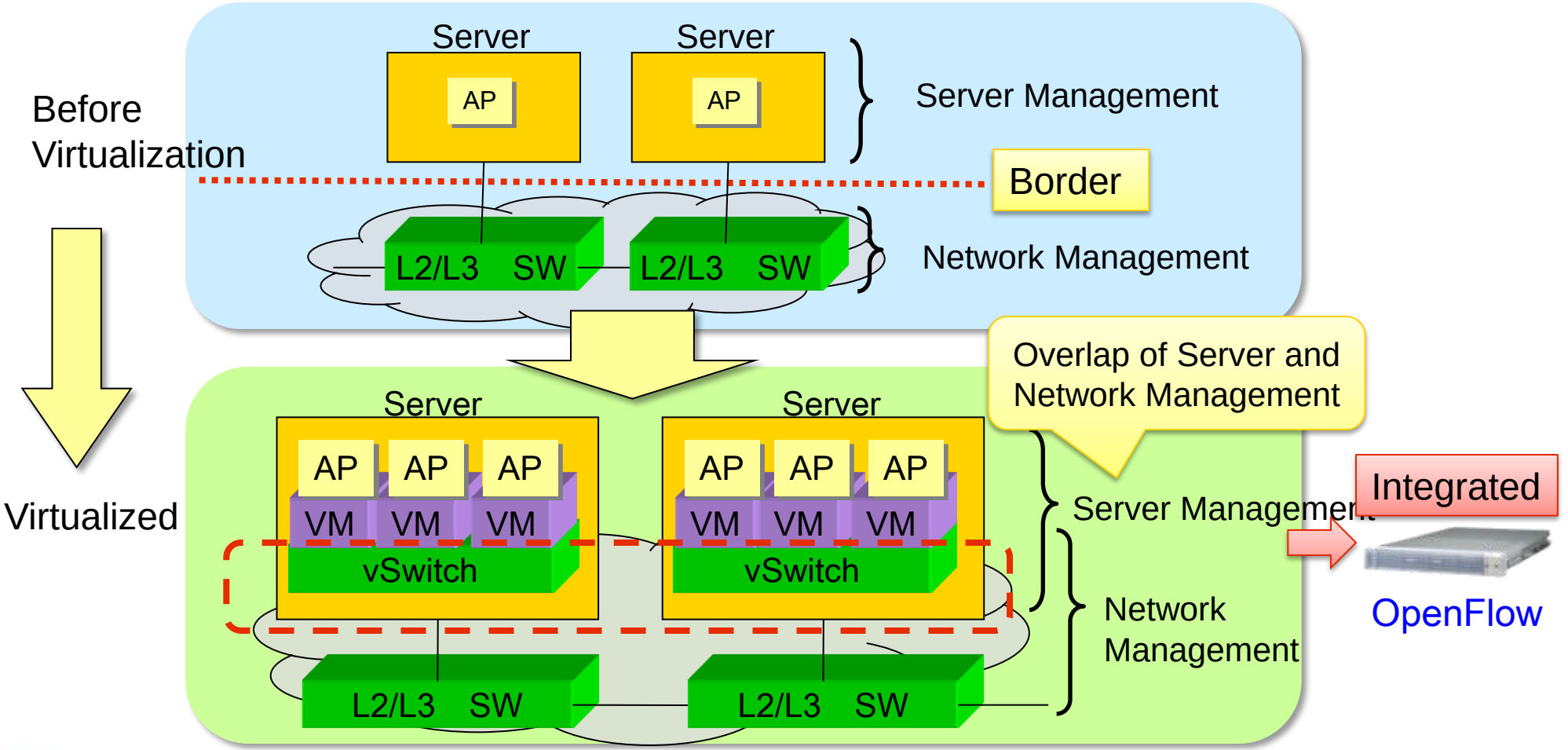
In virtualization era, data center operators face various problems not only server/storage side but also network side.



Now: Server and Network integrated management

Border of Server and Network is not clear.

OpenFlow/SDN can be lead of Server and Network integrated management.



OpenFlow Controller Framework

TREMA

OpenFlow Programmability

- OpenFlow Controller software can manage networks
- There are various control methods
  - HPC
    - Like traditional interconnect
  - Distributed computing
    - Try to use real network instead of overlay
  - IT and Network integrated management
    - Authentication, security, operation..

"*OpenFlow is just* one of the *tools*" - Ivan Pepelnjak

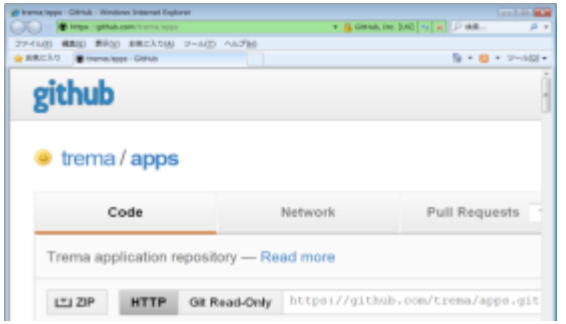
Trema: Full-Stack OpenFlow Framework for Ruby/C

- A software platform for OpenFlow Controller developers

- EASY
  - All-in-one package
  - Integrated developing environment
  - Sophisticated APIs for Ruby and C
- Many sample controllers/parts
  - Useful samples @/src/examples/
  - Practical samples @TremaApps
- Fully tested and supported
- Open community
  - Free software (GPLv2)
  - Community participation (even for commercial product)

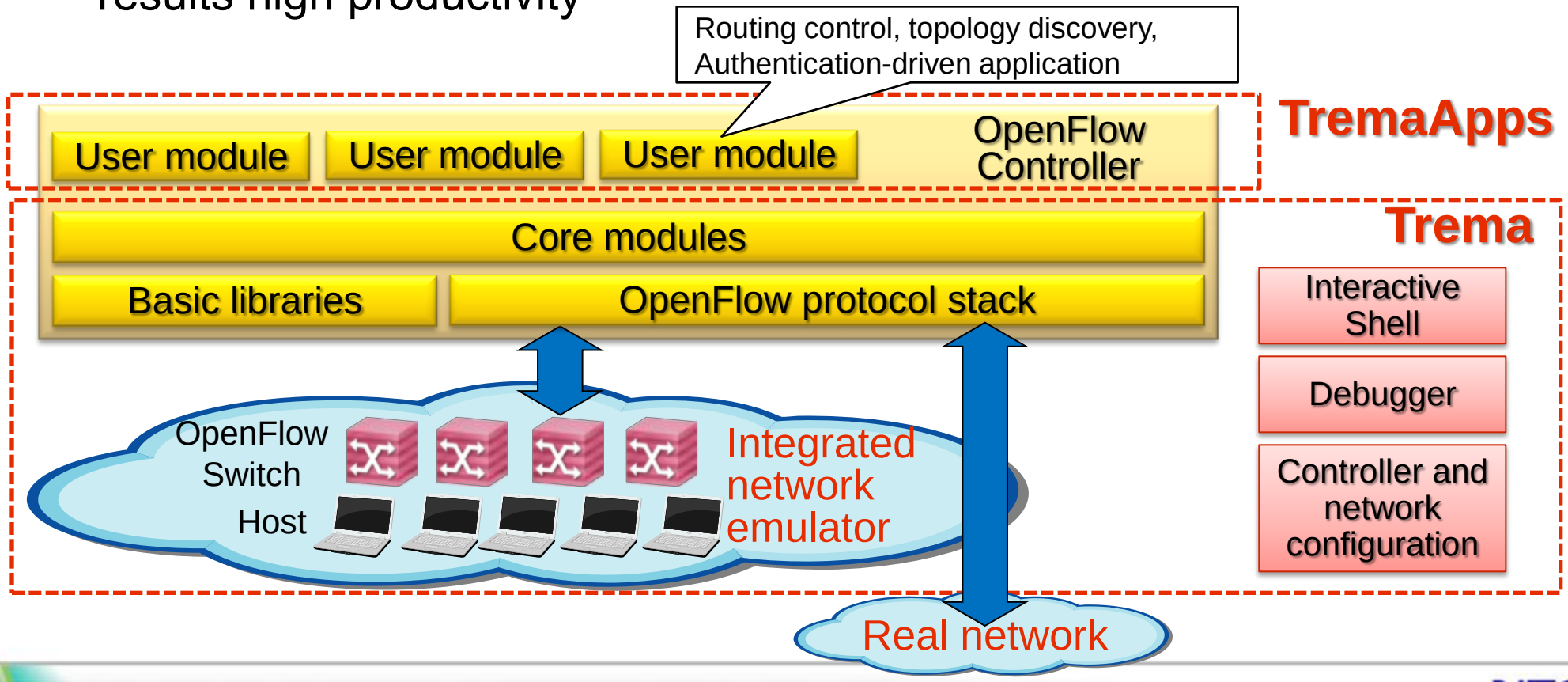
Trema @http://trema.github.com/trema/

TremaApps
@https://github.com/trema/apps



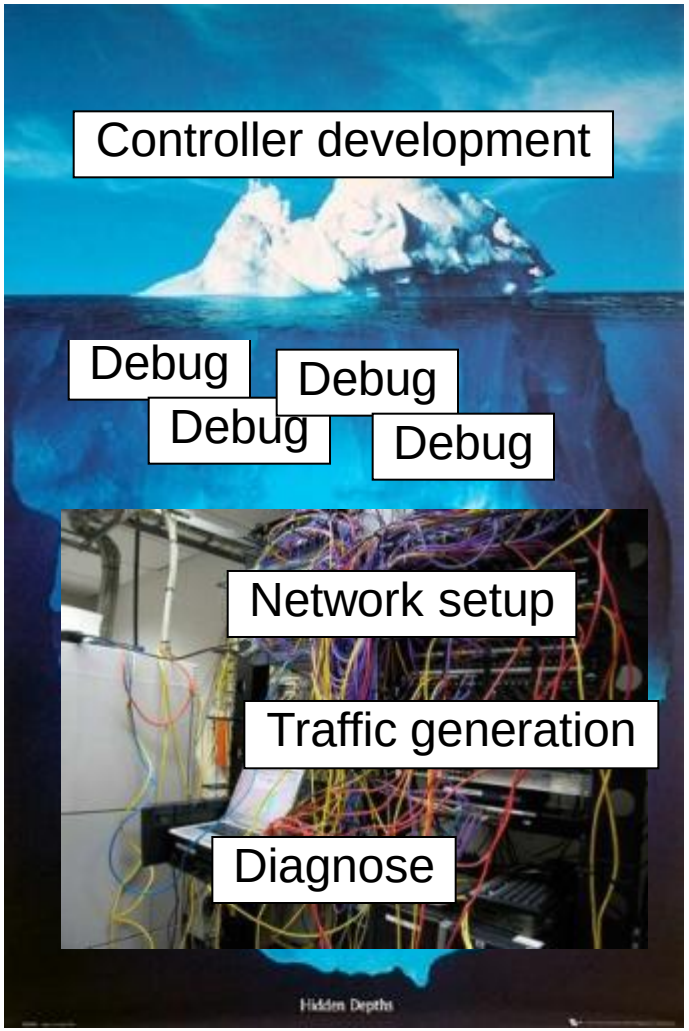
Scope of Trema

- Trema = OpenFlow framework
  = controller platform + integrated network emulator + debugger + etc…
- Why framework? - Tight loop of "coding, testing, and debugging“ results high productivity



Lessons learned

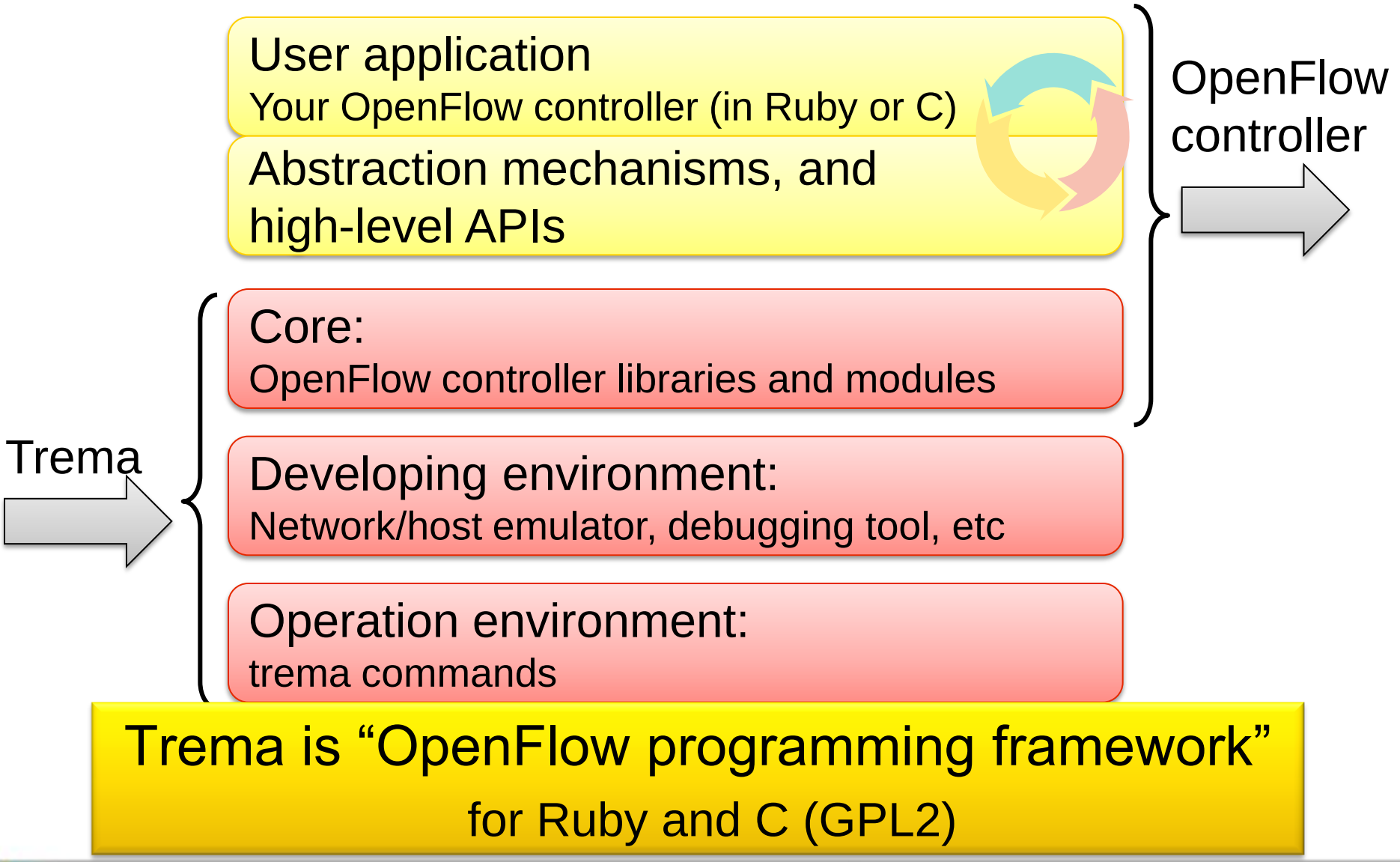
OpenFlow iceberg



Scope

Seamless integration of operations and state monitoring among controller applications, switches, hosts, etc.

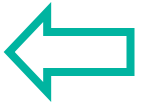
What is Trema



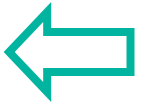
Feature 1: easy coding

Repeater hub written by Ruby

```
class RepeaterHub < Controller
  def packet_in datapath_id, message
    send_flow_mod_add(
      datapath_id,
      :match => ExactMatch.from( message ),
      :actions => ActionOutput.new( OFPP_FLOOD )
    )
    send_packet_out(
      datapath_id,
      :packet_in => message,
      :actions => ActionOutput.new( OFPP_FLOOD )
    )
  end
end
```



Send flow entry add



Packet out

Feature 2: network emulator

- “code, test, debug” in one machine

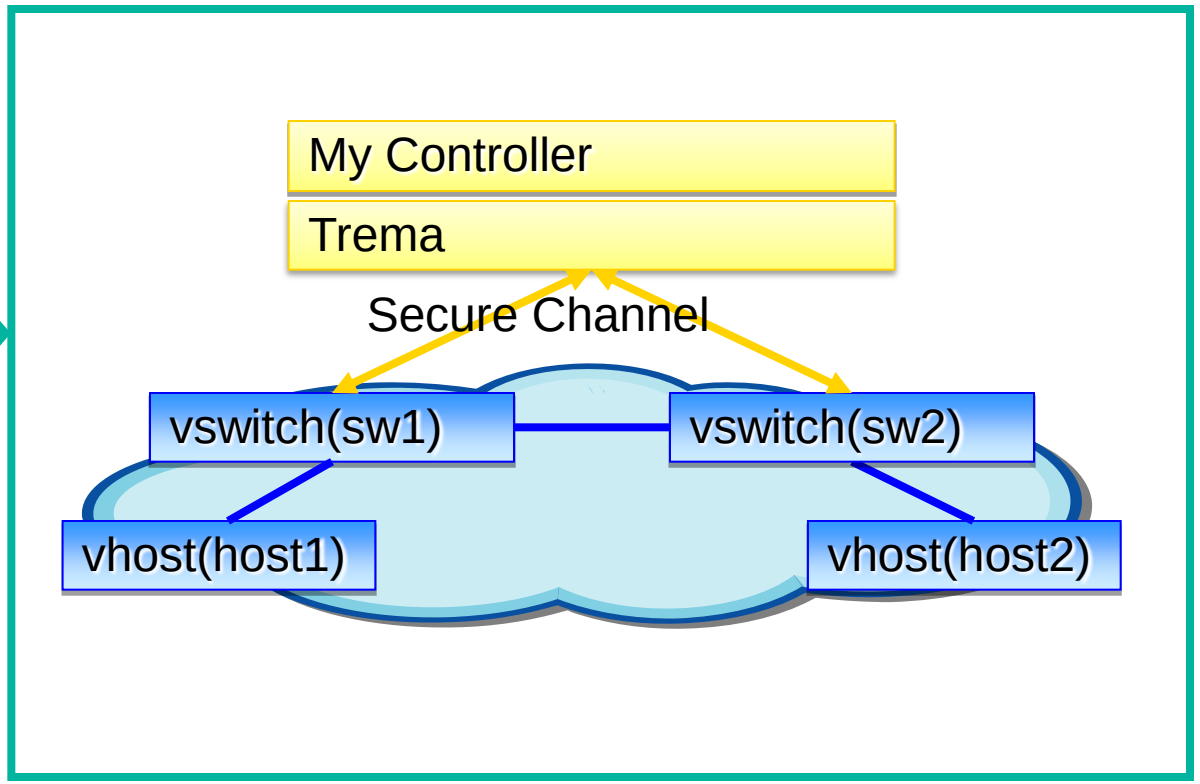
Emulation script

```
vswitch("sw1") {
  datapath_id "0x1"
}
vswitch("sw2") {
  datapath_id "0x2"
}

vhost ("host1")
vhost ("host2")

link "host1", "sw1"
link "sw1", "sw2"
link "host2", "sw2"
```

Development machine



Feature 3: many sample applications

Trema source tree

- Repeater hub
- Learning switch
- Traffic monitor

Trema application repository https://github.com/trema/apps

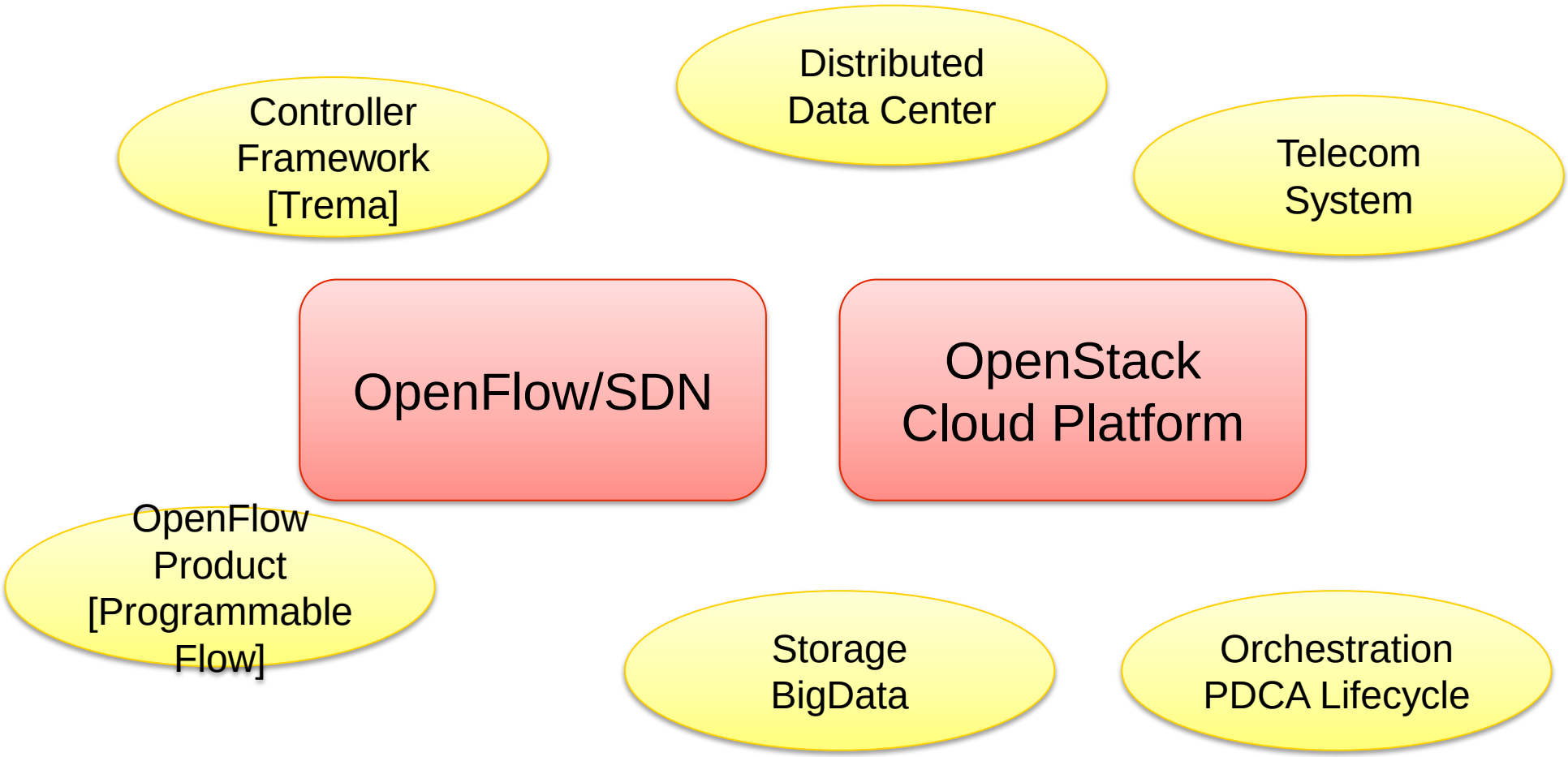
- Topology discovery/management
- Routing switch
- Sliceable Switch

Documents published

- http://gihyo.jp/dev/serial/01/openflow_sd/0010

SOLUTIONS

Our activities around OpenFlow/SDN and OpenStack

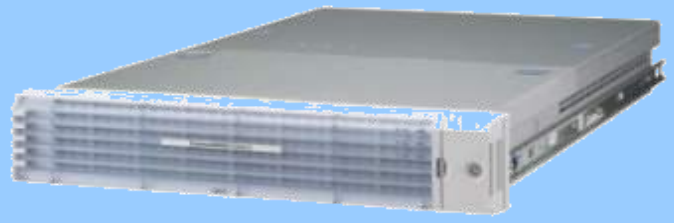


Award-winning ProgrammableFlow

Simple, scalable, secure, open networking

- First OpenFlow-enabled fabric
- Secure network-wide virtualization
- Drag and drop, programmable networking

ProgrammableFlow Controller



ProgrammableFlow Switch Family

Best of Microsoft TechEd North America 2012 — FINALIST — Microsoft Partner Product Awards Produced By: Windows IT Pro | SQL Server Pro | Dev Pro | SharePoint Pro

best of INTEROP Awards 2012 — GRAND PRIZE WINNER — Grand Prize — PRESENTED BY InformationWeek reports

best of INTEROP Awards 2012 — WINNER — Management, Monitoring & Testing — PRESENTED BY InformationWeek reports

BEST OF INTEROP 2011 — Infrastructure — PRESENTED BY InformationWeek analytics

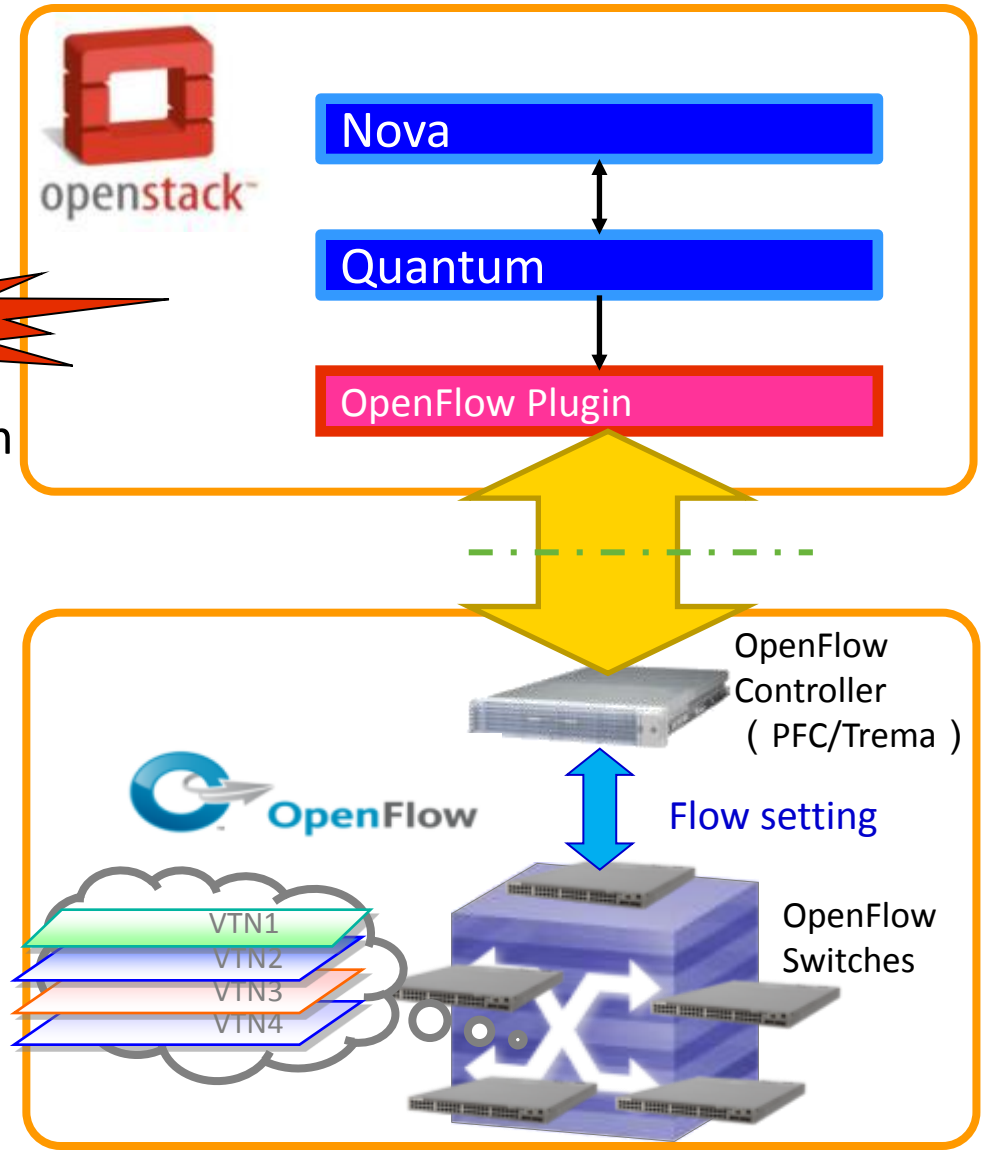
BEST OF SHOW AWARD INTEROP — Grand Prix

BEST OF SHOW AWARD INTEROP — Special Prize

Quantum OpenFlow Plugin

- OpenStack Quantum
  - OpenStack sub-project
  - Managing virtual network
- OpenFlow Plugin
  - The plugin to use OpenFlow from Quantum
  - Download from https://github.com/nec-openstack/quantum-openflow-plugin
- Supported OpenFlow Controller platform
  - Trema http://trema.github.com/trema/
  - ProgrammableFlow PF6800

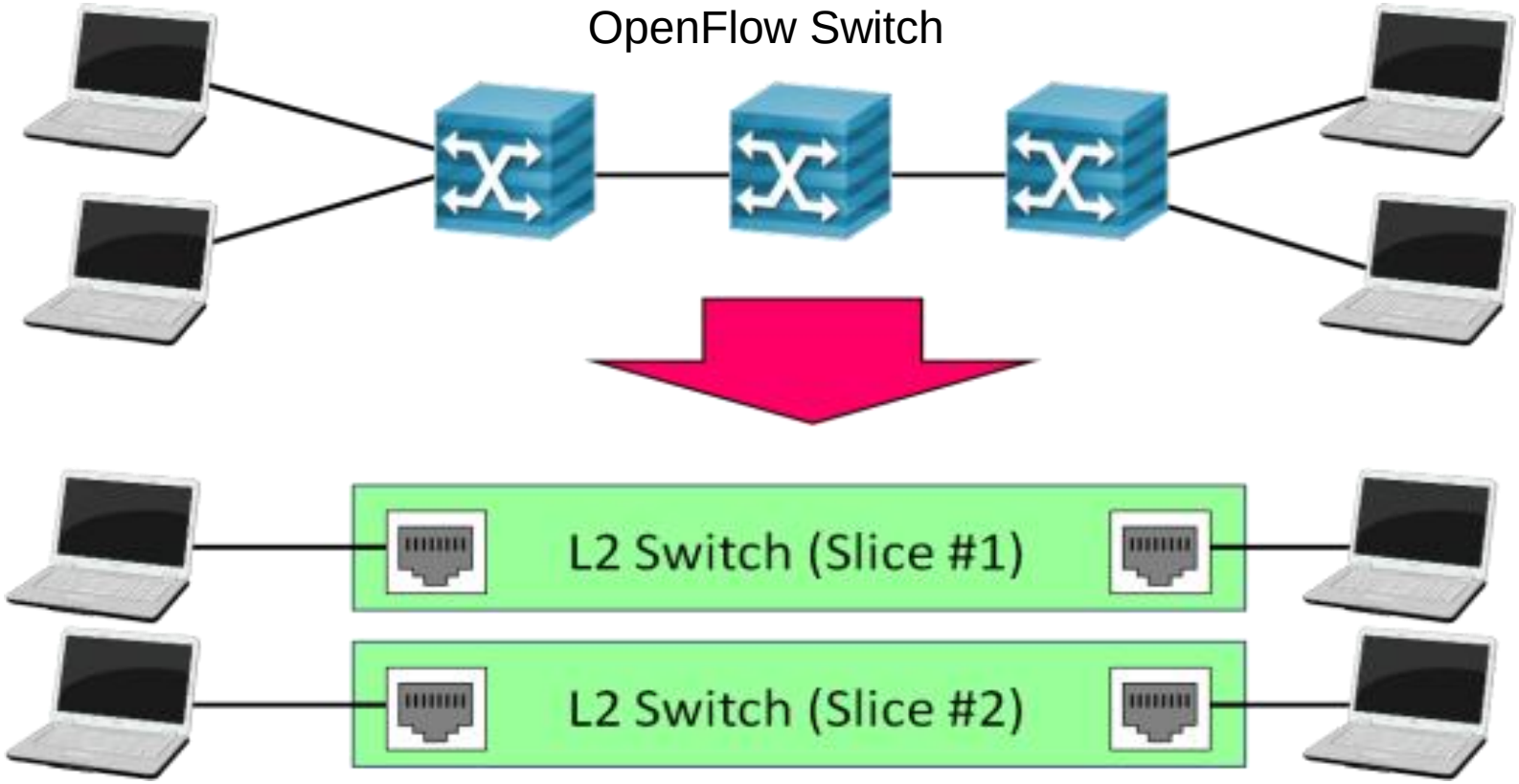
OSS Released



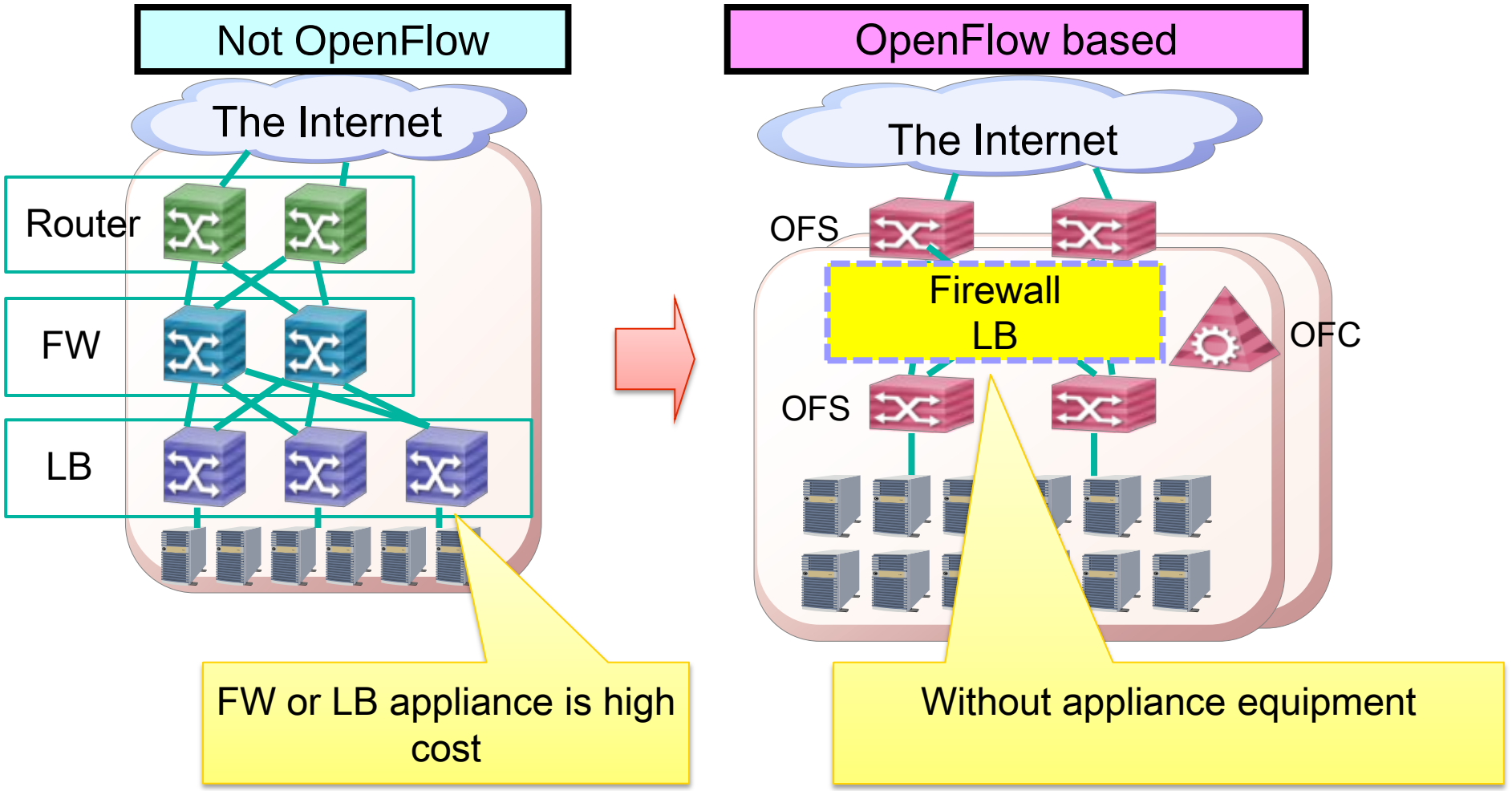
Isolation of networks

SliceableSwitch (trema application)

- Create virtual L2 slice on OpenFlow NW
- L1-L4 filter function enabled



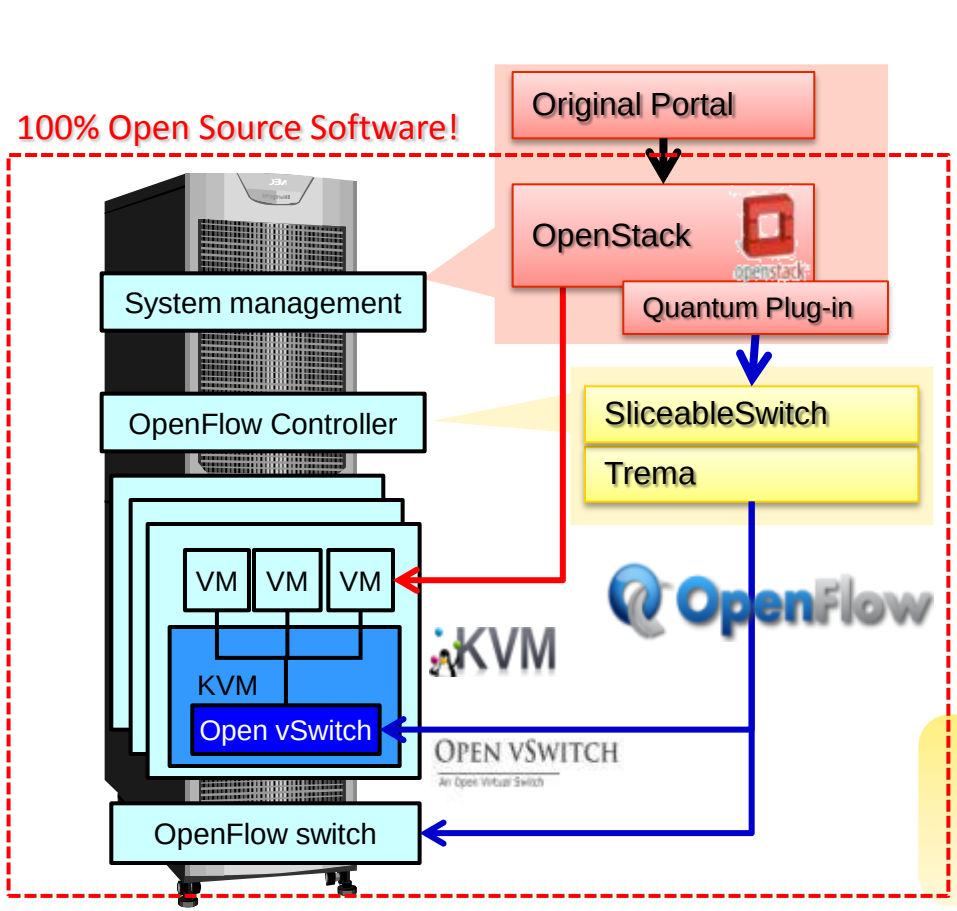
OpenFlow based Firewall / Load Balancer

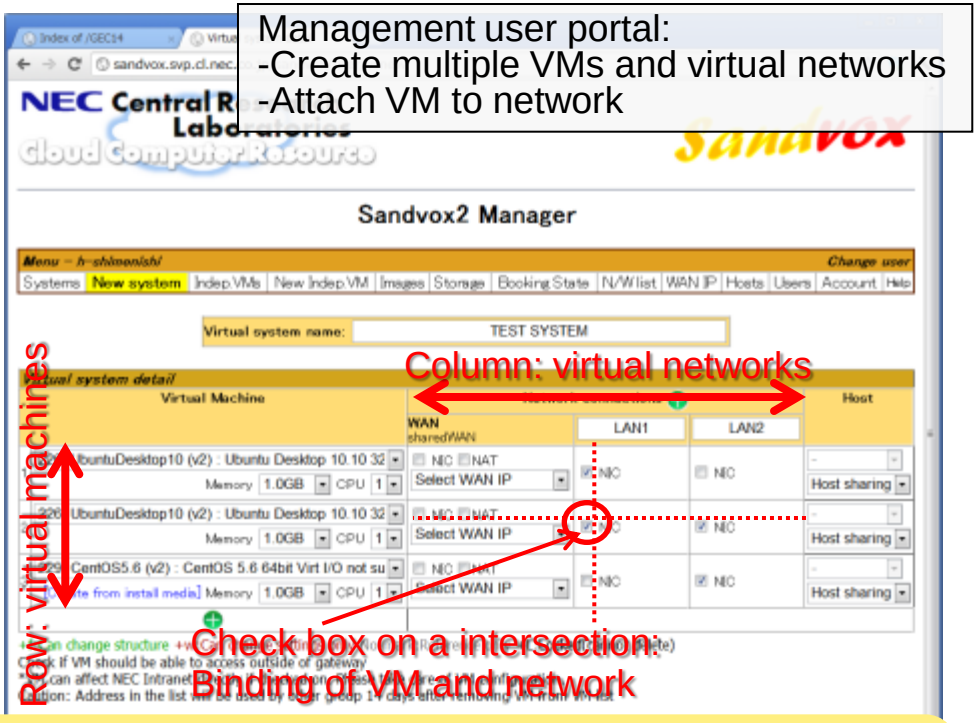


Integrated resource management

IT and network resource integrated management

- OpenFlow for network resource virtualization
- OpenStack for computing resource virtualization



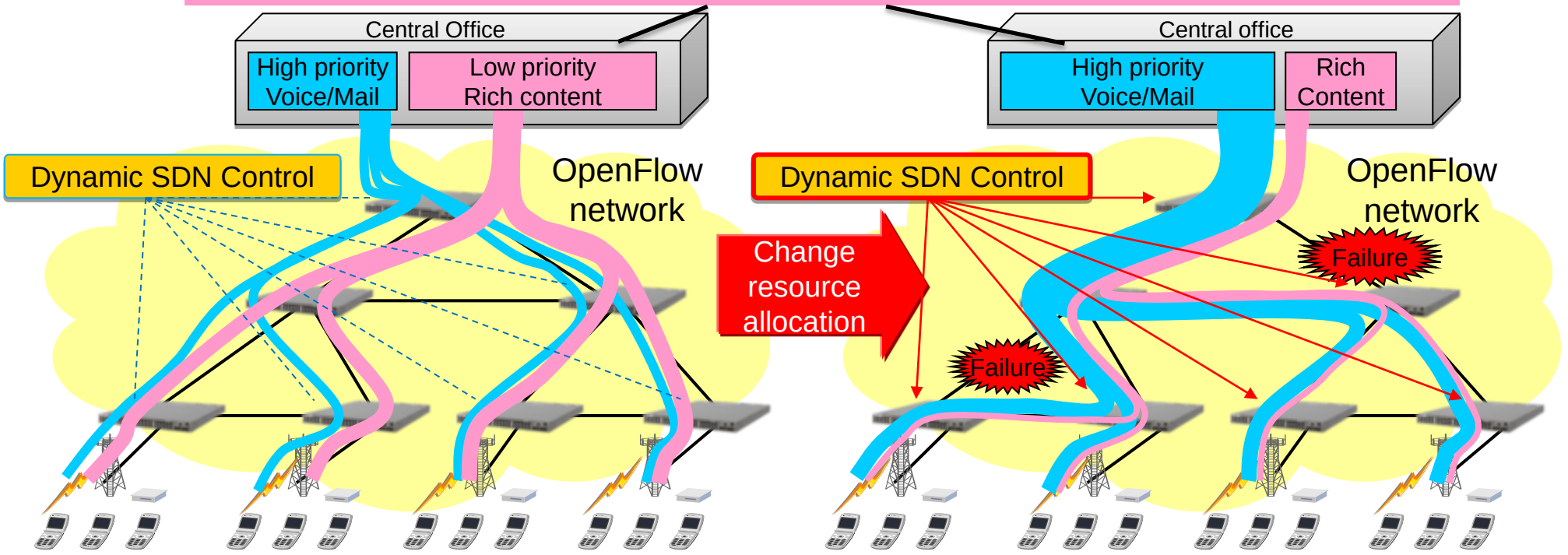


Enabling flexible service provisioning and agile service deployment

Virtualization of mobile core and dynamic SDN control

New project with NTT DoCoMo to make a mobile core network flexible and scalable against network congestions due to disaster

Virtualization of mobile core network services and dynamic server resource re-allocation for high priority service (Voice/Mail).



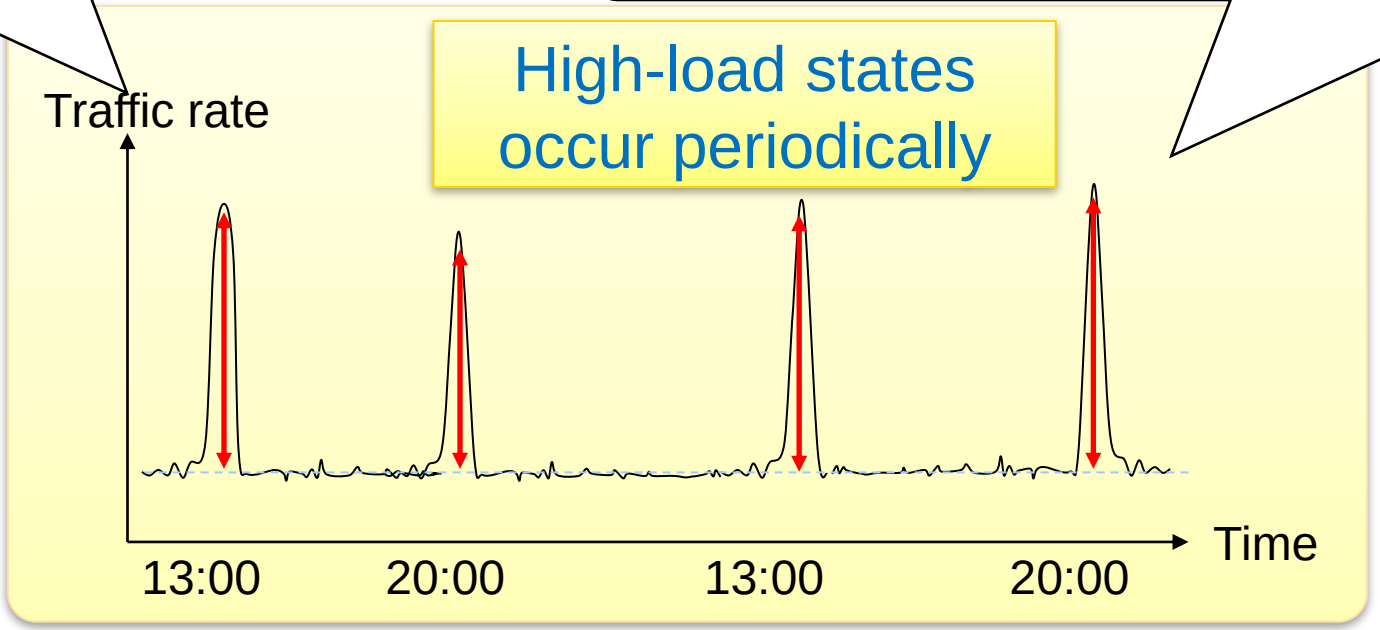
According to network resource re-allocation for high priority service (Voice/Mail), mobile access network path can be dynamically adjusted.

Issues in Mobile Networks

- Voice traffic congestion during 3.11 disaster
- Congestion due to popularity of smart phones
- Now, control plane signaling will also increase

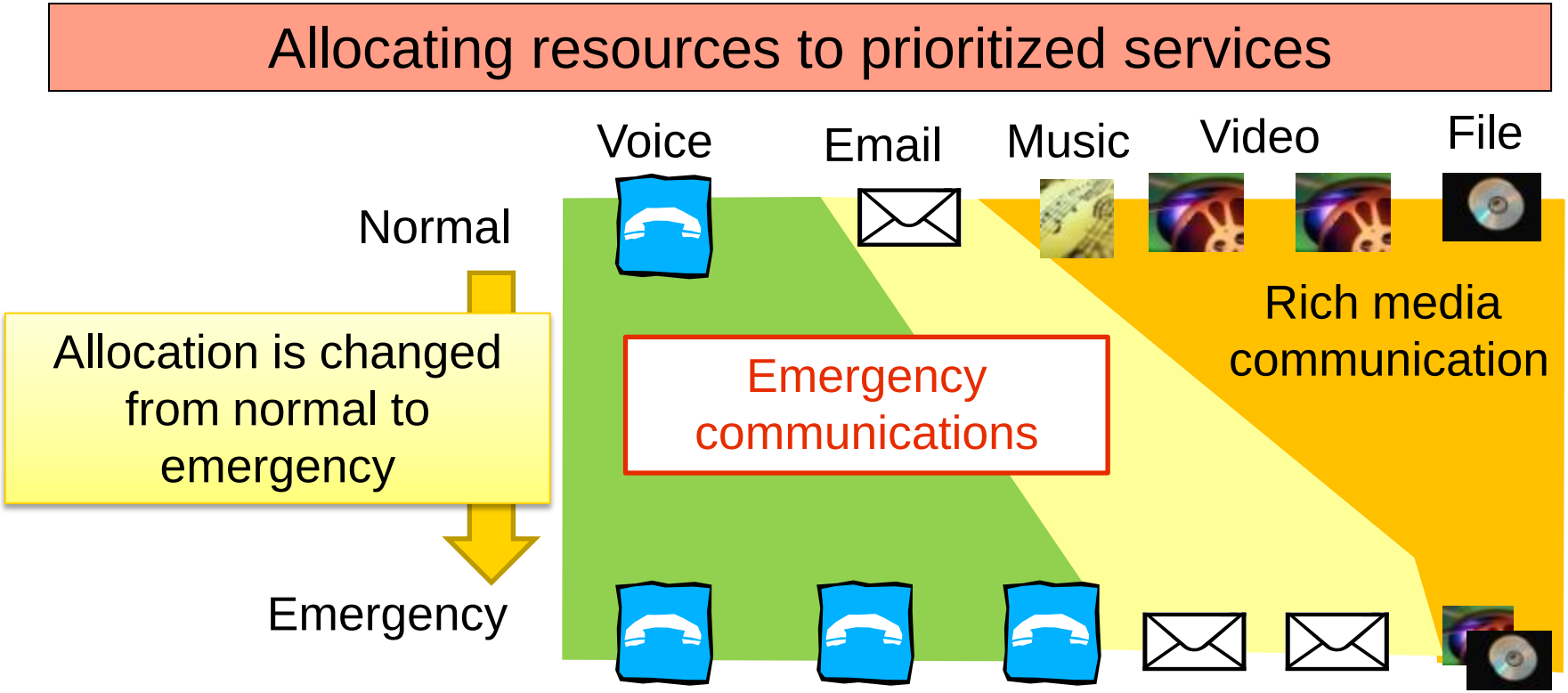
Smart phones cause spike in network traffic

Design to allocate fixed resources each peak is a waste of money



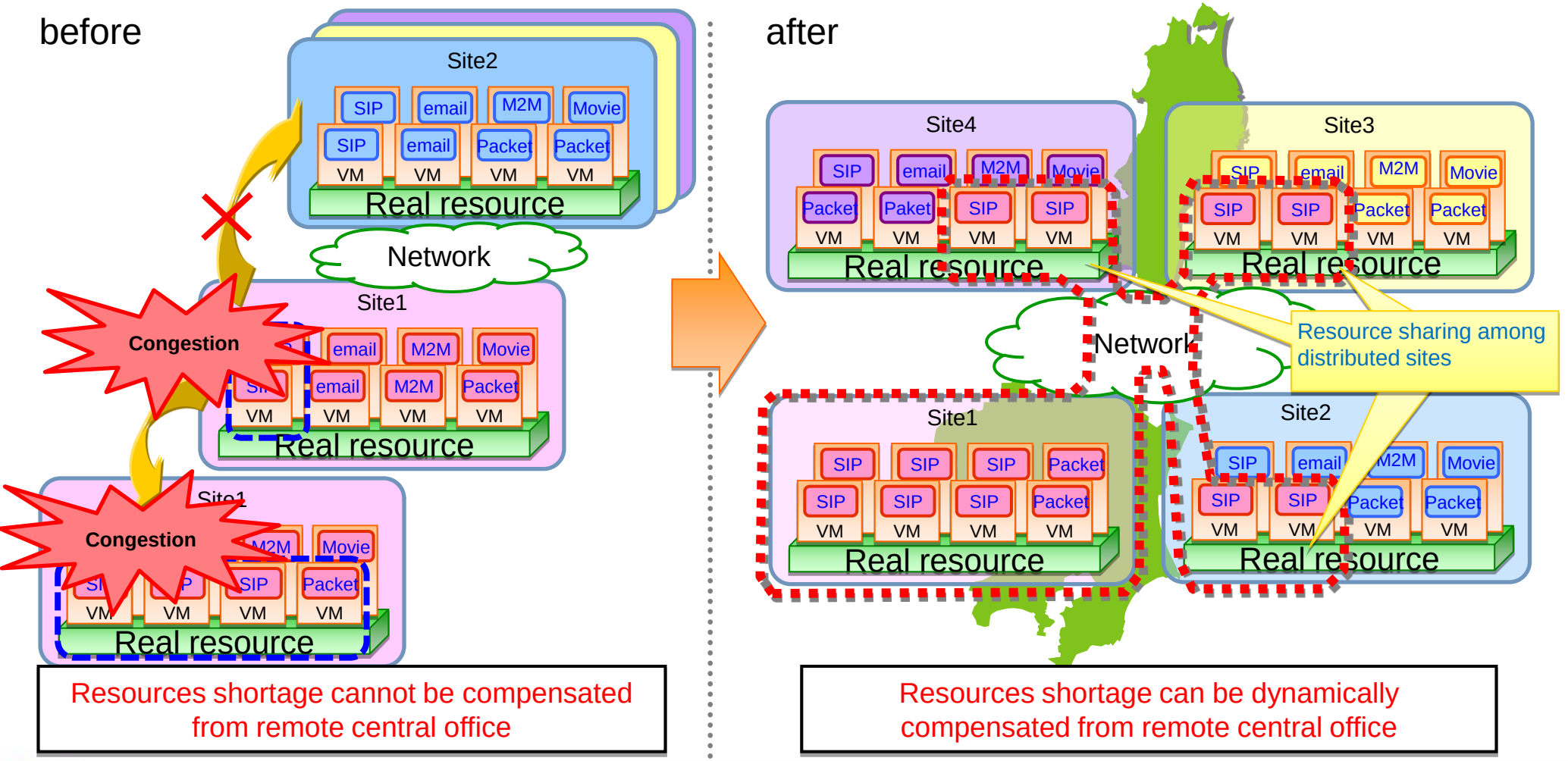
Solution

- Dynamic allocation of mobile network resources to solve various network congestions due to
  - bursty traffic patterns of smart phones
  - immediate increase of traffic patterns due to disaster



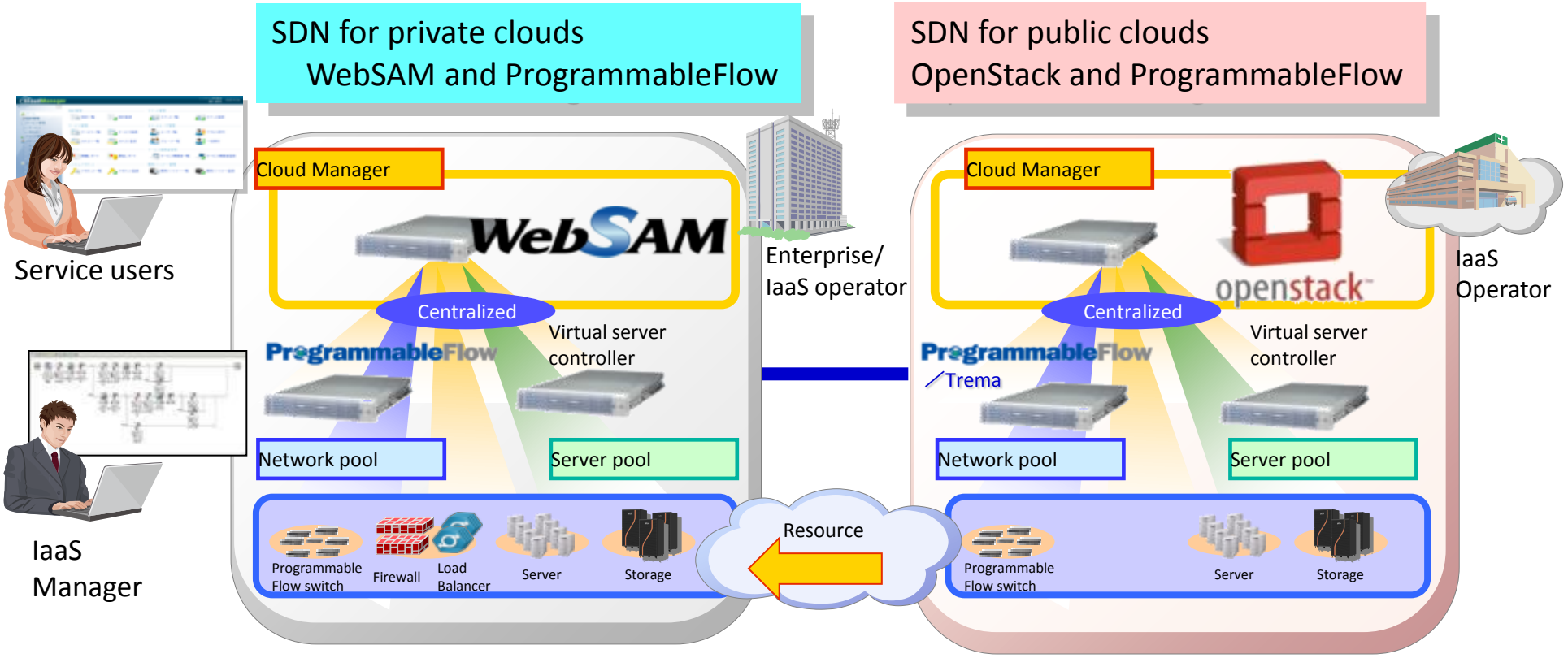
Resource Allocation Control for Inter Central Office

Dynamic resource allocation of virtualized software within Inter central office



Hybrid Cloud

Demonstration at the Interop TOKYO, June 2012

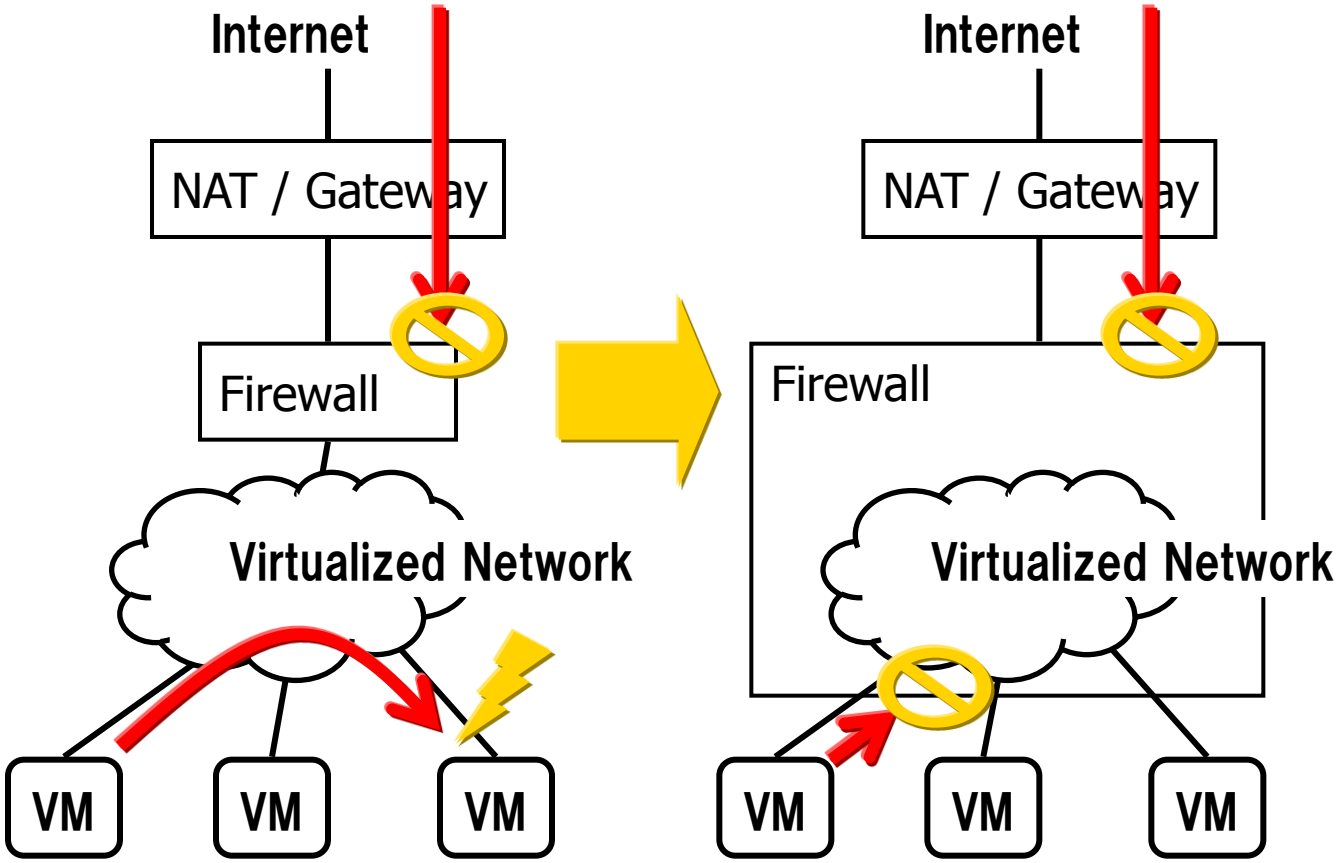


NEC original solutions for sophisticated Enterprise users and IaaS operators

IaaS operators with their customizations

Security

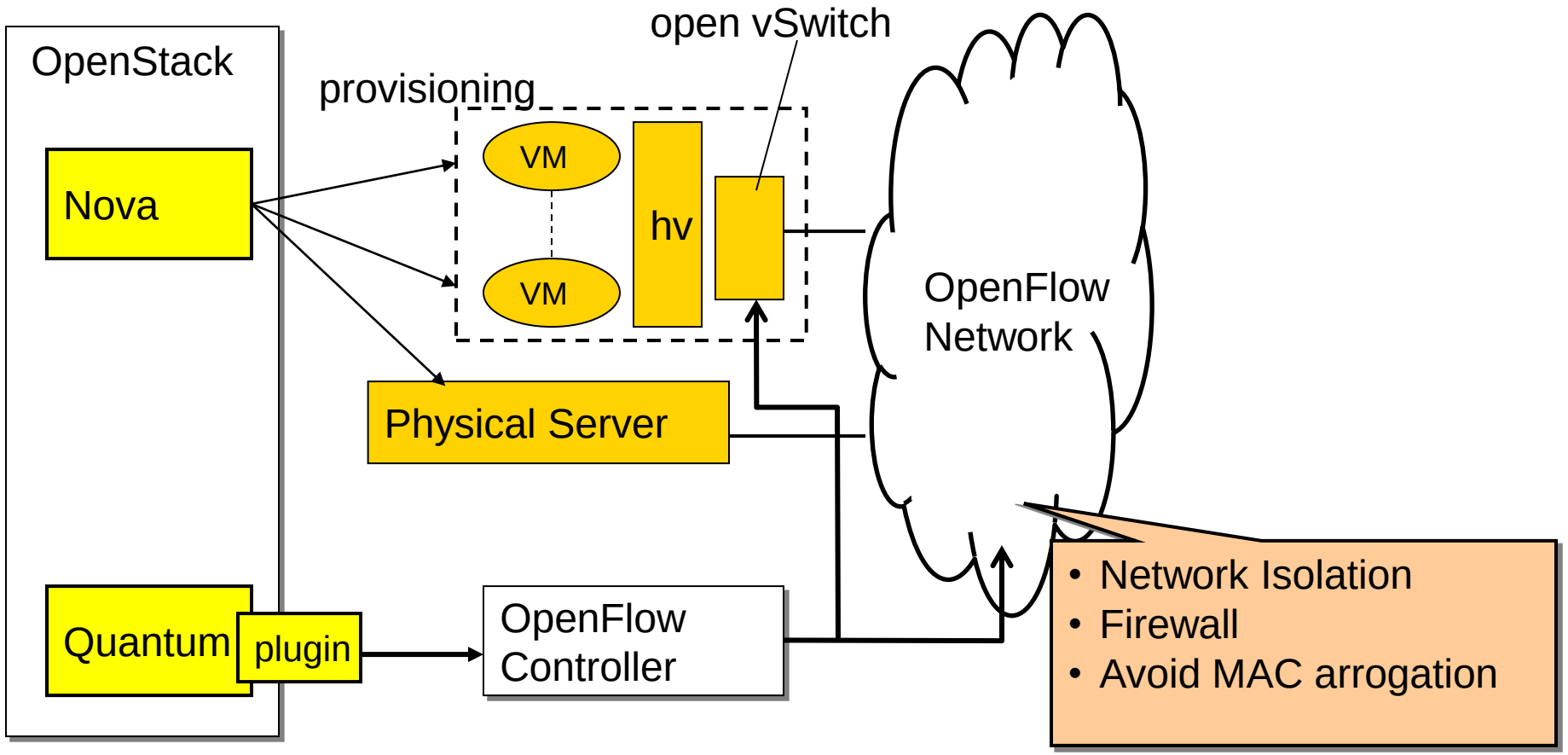
- Quantum extension for network filtering
- Filtering and detecting packet between VMs



Bare metal support

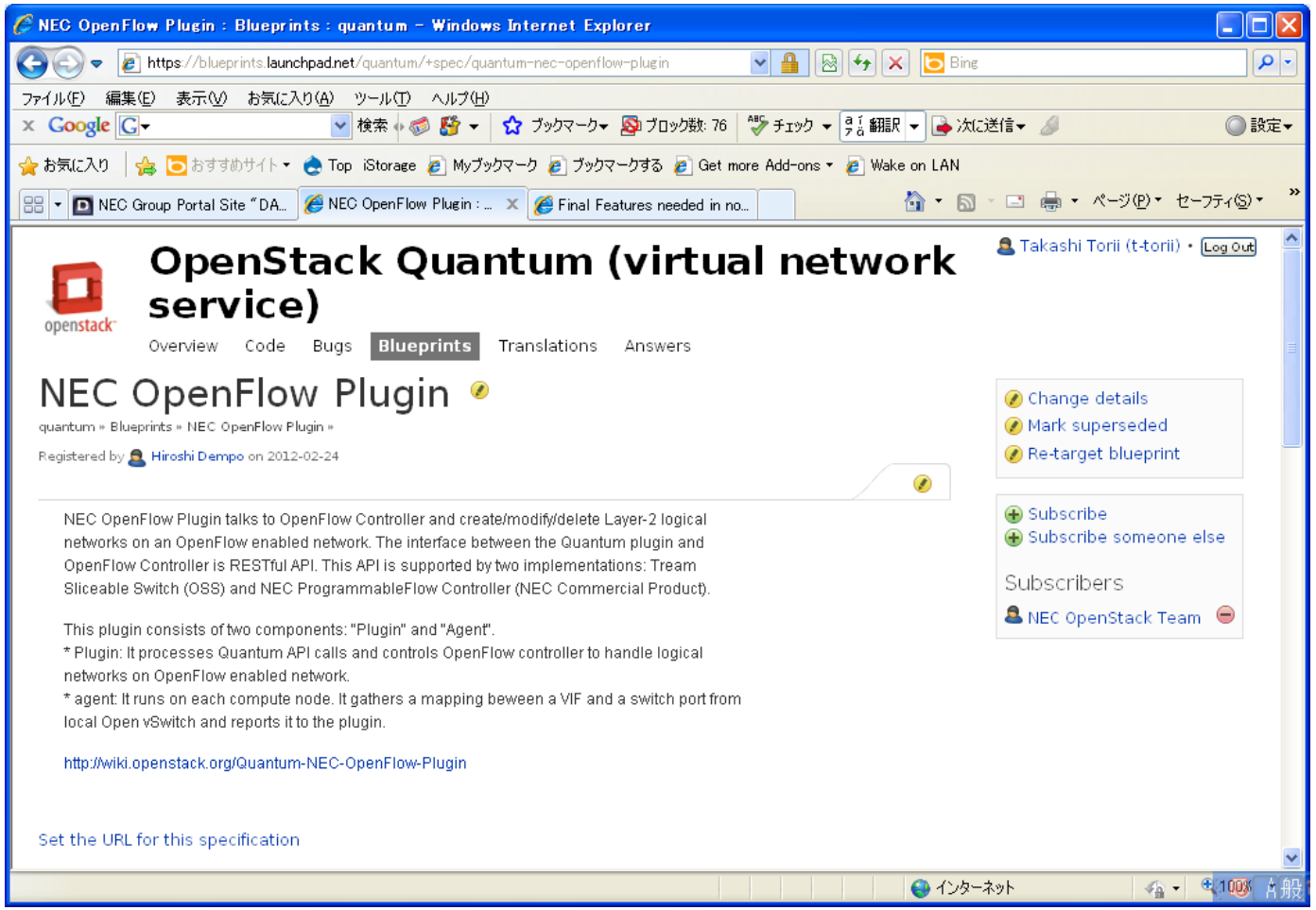
OpenStack bare metal with network isolation

- OpenFlow switch can isolate network
- security function by OpenFlow



Contribution to OpenStack community

✓OpenFlow Plugin target Folsom-3
✓Quantum other features



Why OSS?

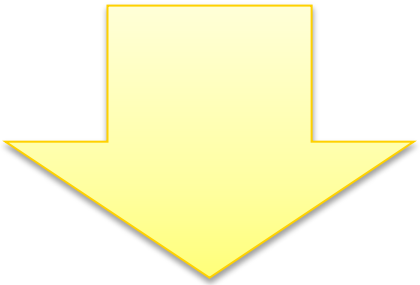
- Vender's products are like "Tower"

Tokyo SkyTree, 634m

- OSS is like "City"

How to make Innovations?

- Use potentially technology/platform
- Open
- Collaboration



OpenStack and OpenFlow

Thank you!

Contact : t-torii@ce.jp.nec.com

Empowered by Innovation