Lecture 4

Agenda

- Introduction
- SDN and OpenFlow
- Network Virtualization
- Network Virtualization in OpenStack
- Our Work

The Service Trend

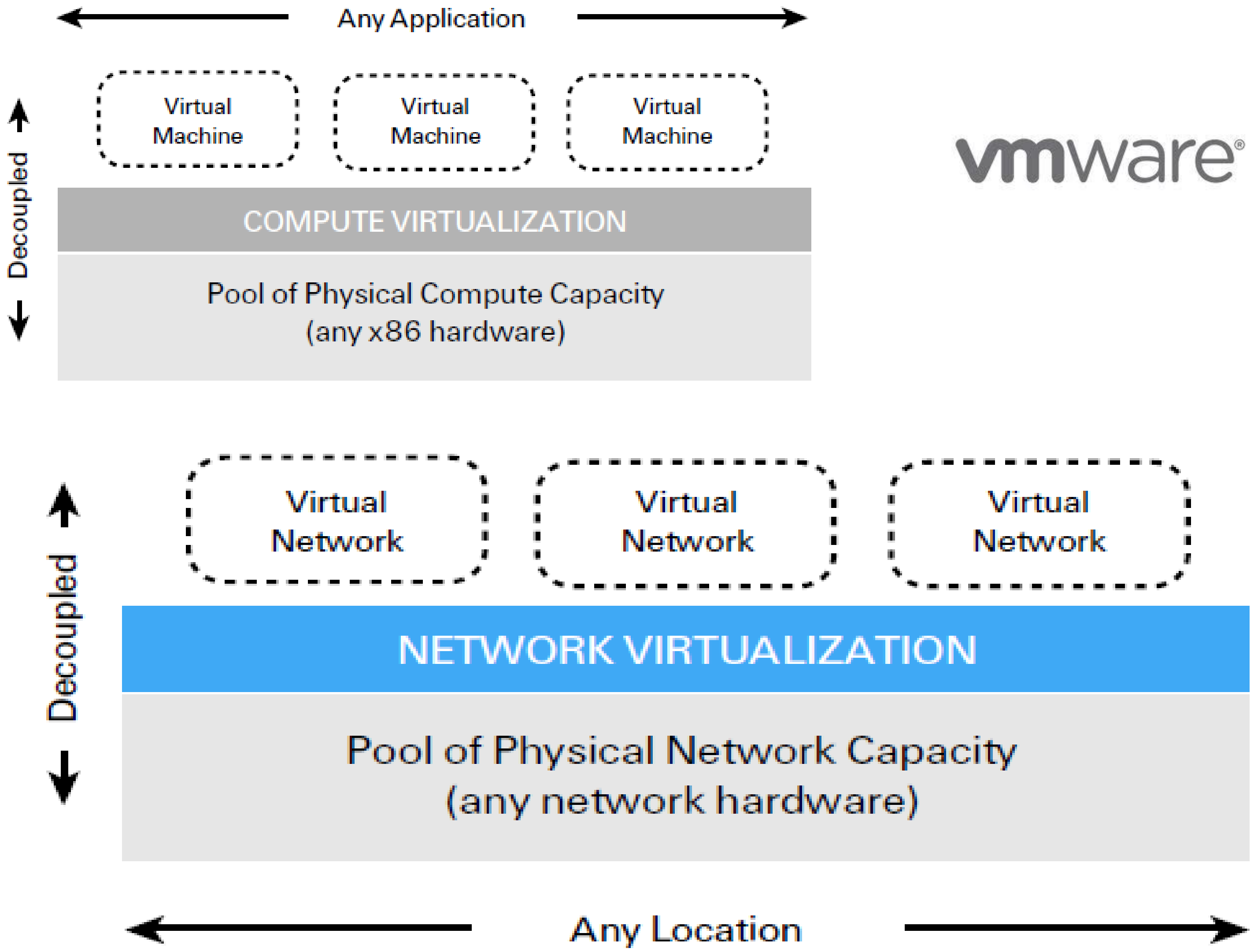
- "Decoupling infrastructure management from service management can lead to innovation, new business models, and a reduction in the complexity of running services. It is happening in the world of computing, and is poised to happen in networking."

Jennifer Rexford

Professor, Princeton University

- Last month, VMware paid $1.2B to acquire Nicira for software defined networking (SDN).

Why is Nicira worth $1.2 billion?



vmware

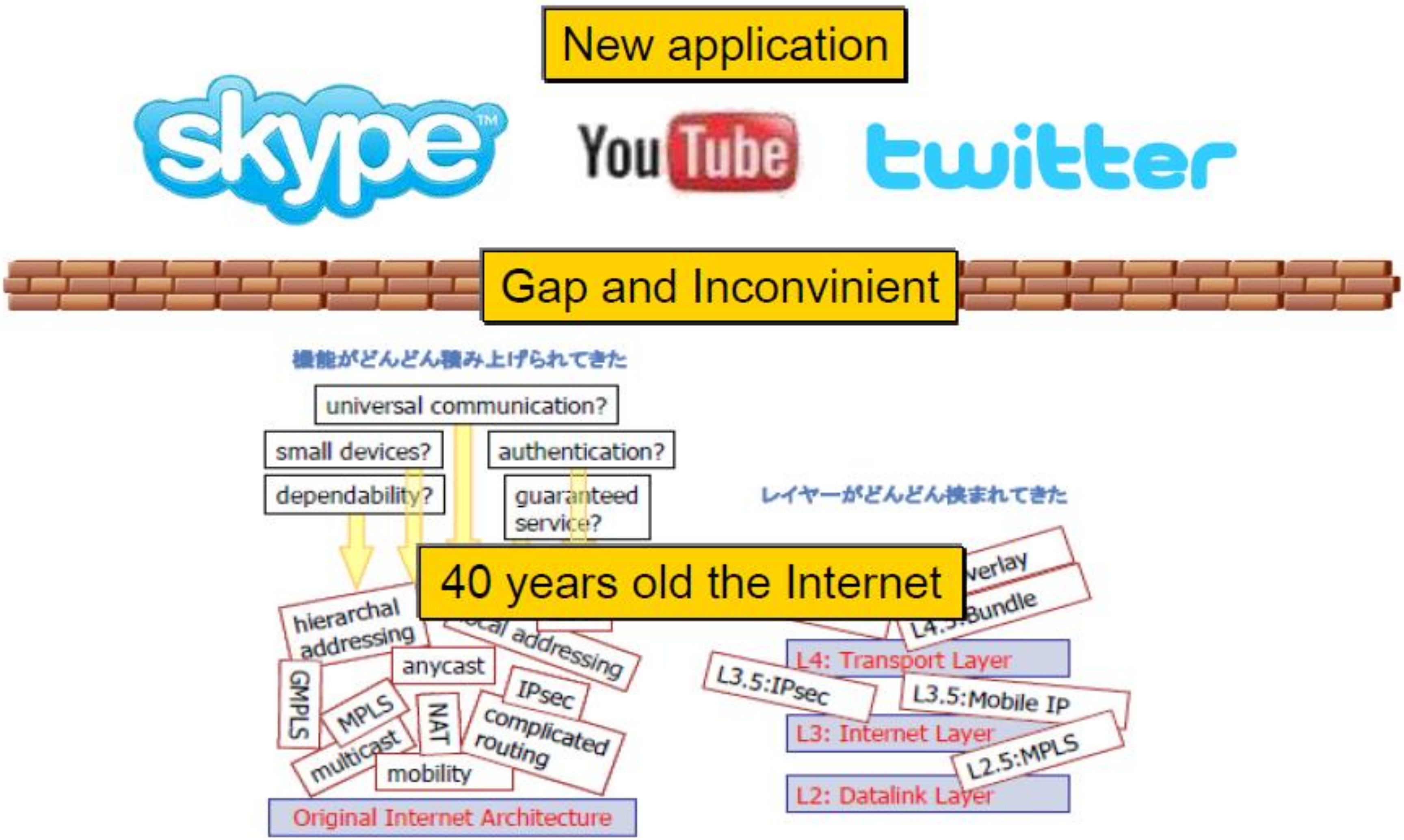
nicira

SDN and OpenFlow

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

40 years old the Internet

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.
  - GENI@USA, FP7@EU, 高可信网络@China...
  - OpenFlow born in Stanford Univ.

Software Defined Network

- 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

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

Stanford University
CLEAN SLATE
An Interdisciplinary Research Program

- Mar.2011: Open Networking Foundation Founded
  - https://www.opennetworking.org/

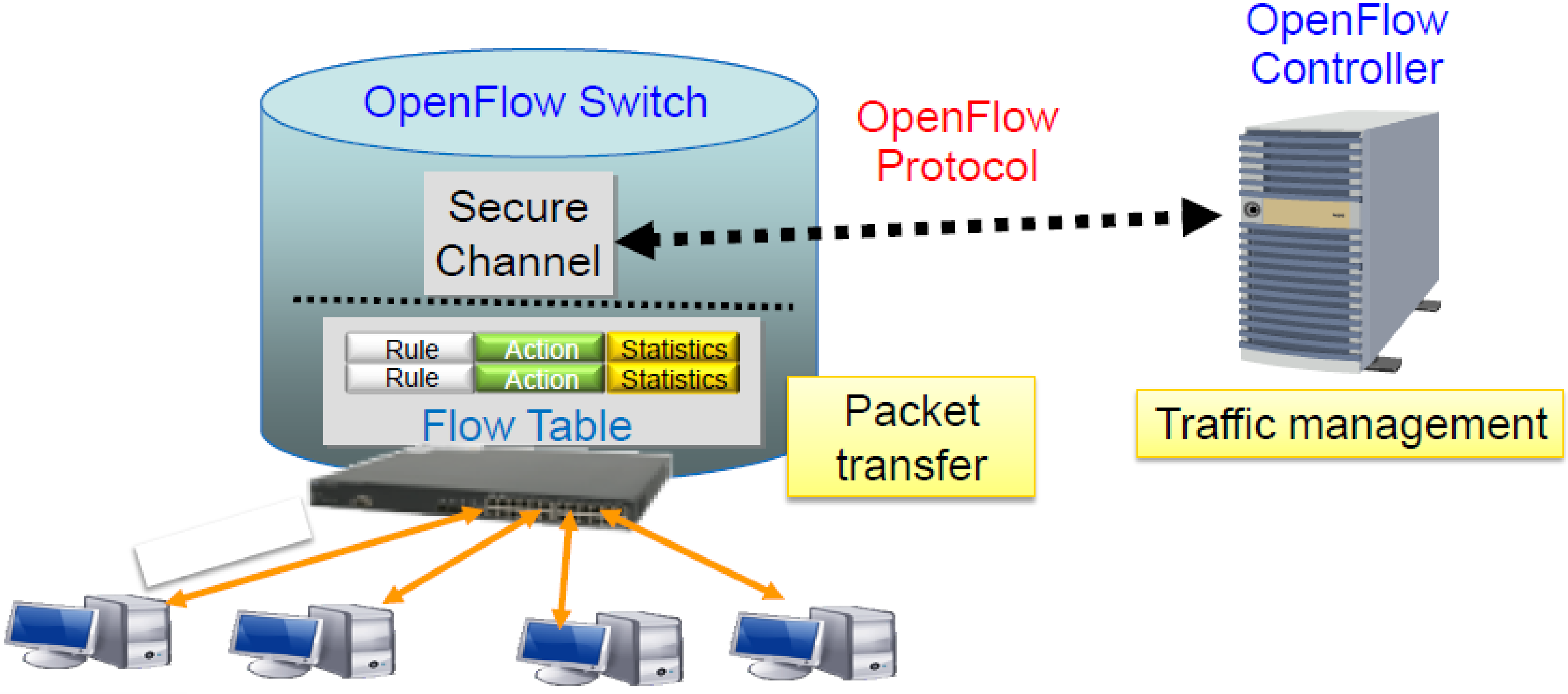
ONF
OPEN NETWORKING
FOUNDATION

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

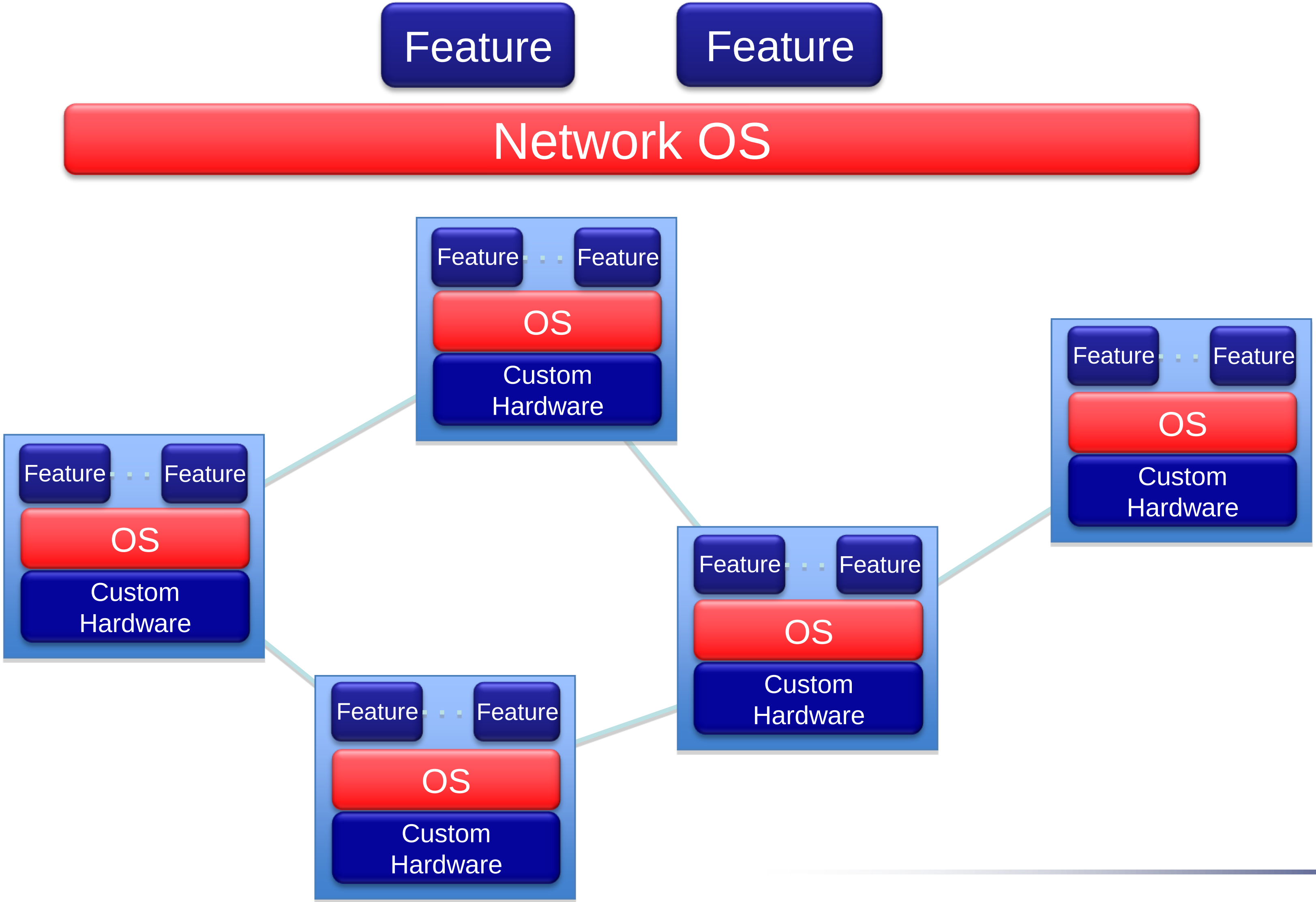
Open Networking Research Center
at Stanford University

OpenFlow Basics: Architecture

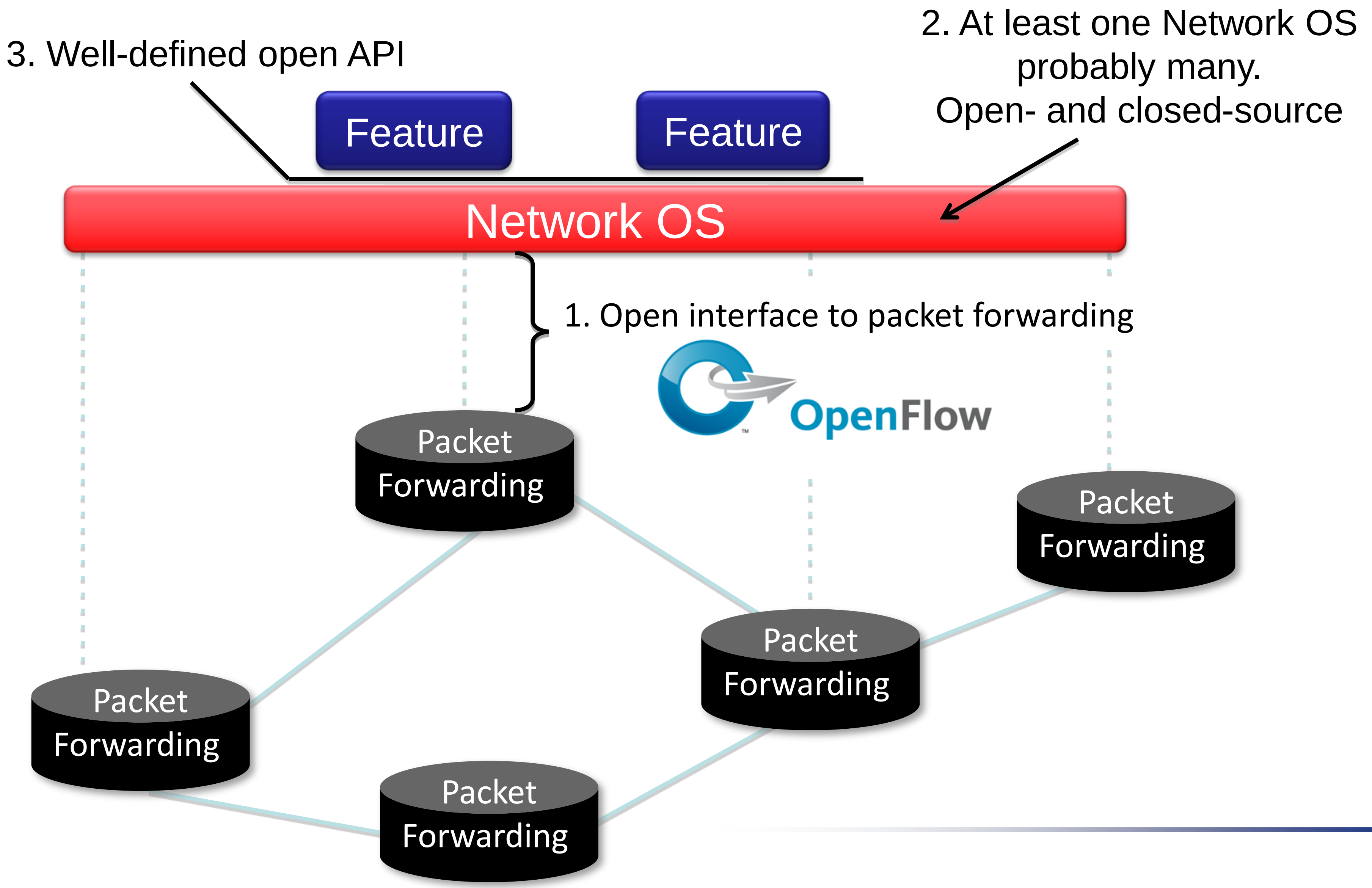
- Separate Data Plane and Control Plane
- OpenFlow is the protocol between switch and controller
- L1-L4 field are used for switching



Network restructuring



Software Defined Network



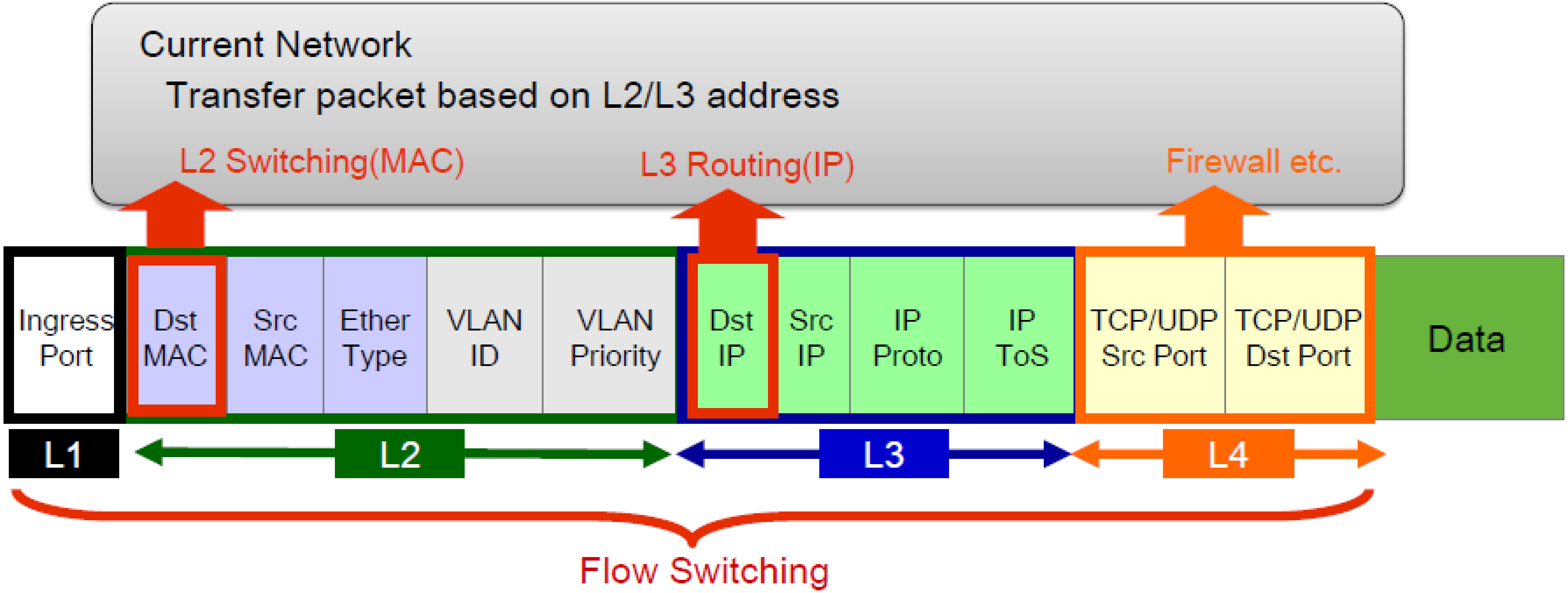
3. Well-defined open API

2. At least one Network OS
probably many.
Open- and closed-source

1. Open interface to packet forwarding

OpenFlow Basics: Flow Switching

Transfer packet based on “Flow”



Current Network

Transfer packet based on L2/L3 address

L2 Switching(MAC)

L3 Routing(IP)

Firewall etc.

Ingress Port

Dst MAC

Src MAC

Ether Type

VLAN ID

VLAN Priority

Dst IP

Src IP

IP Proto

IP ToS

TCP/UDP Src Port

TCP/UDP Dst Port

Data

L1

L2

L3

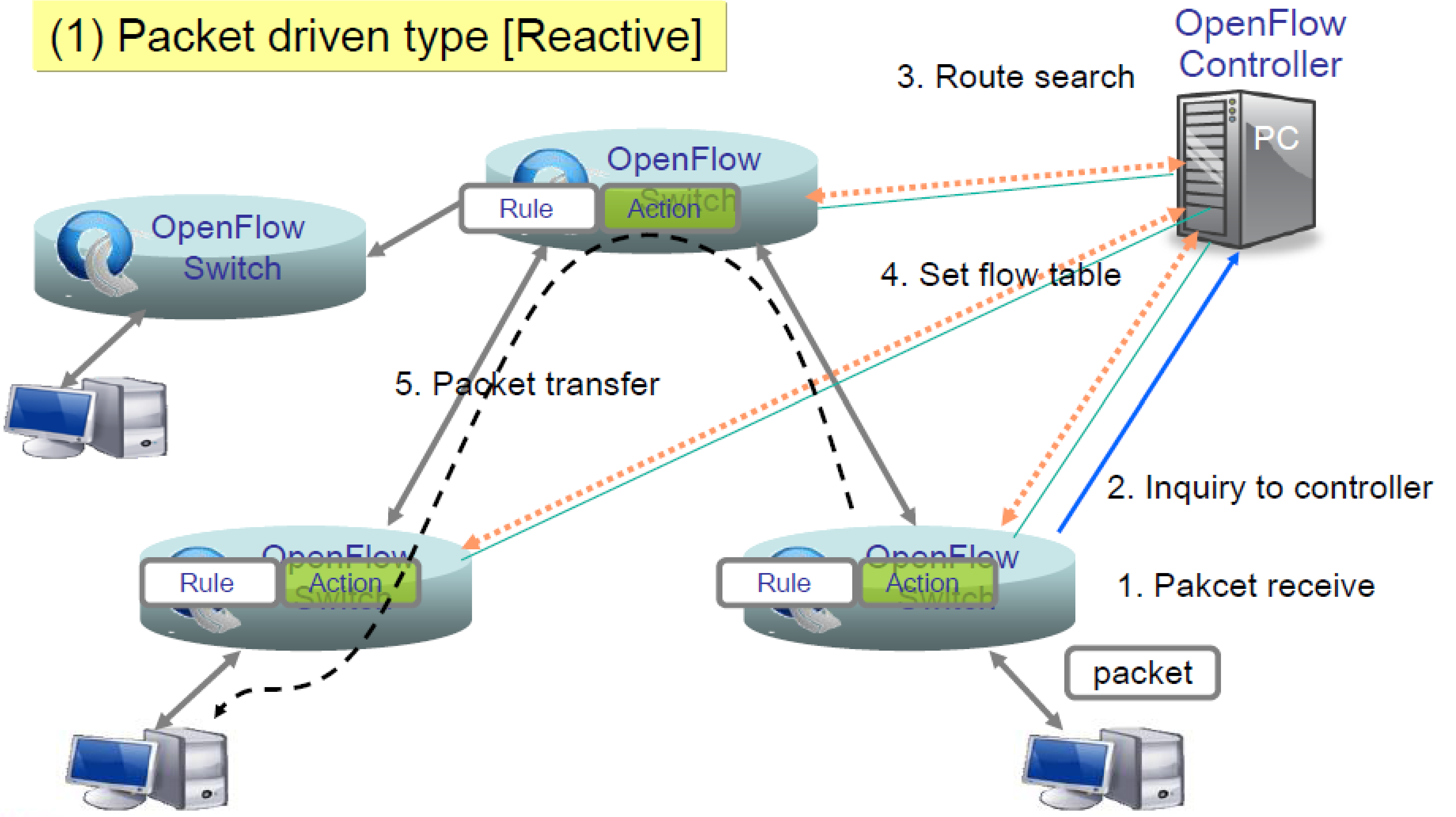
L4

Flow Switching

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.

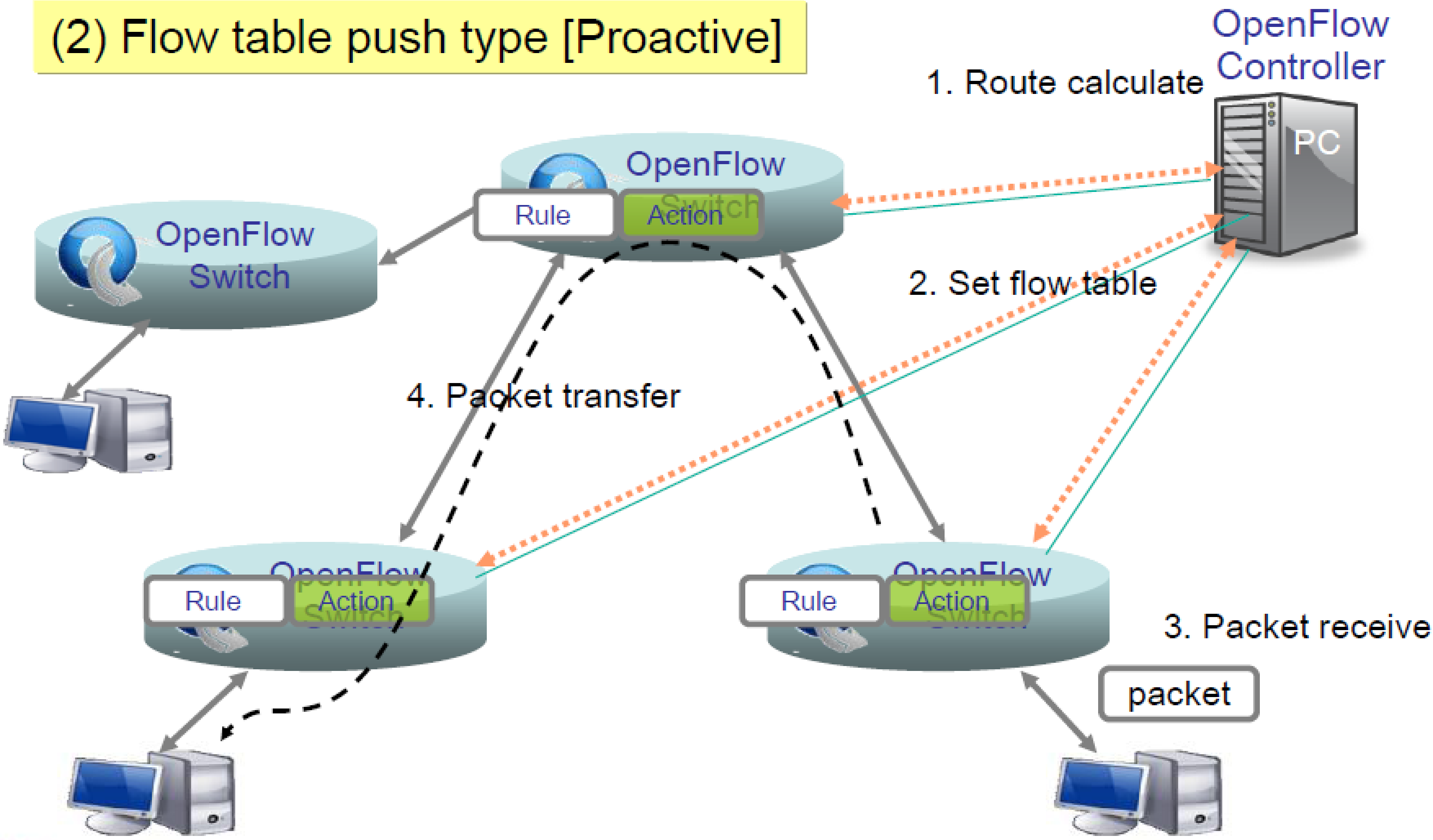
How OpenFlow works

(1) Packet driven type [Reactive]

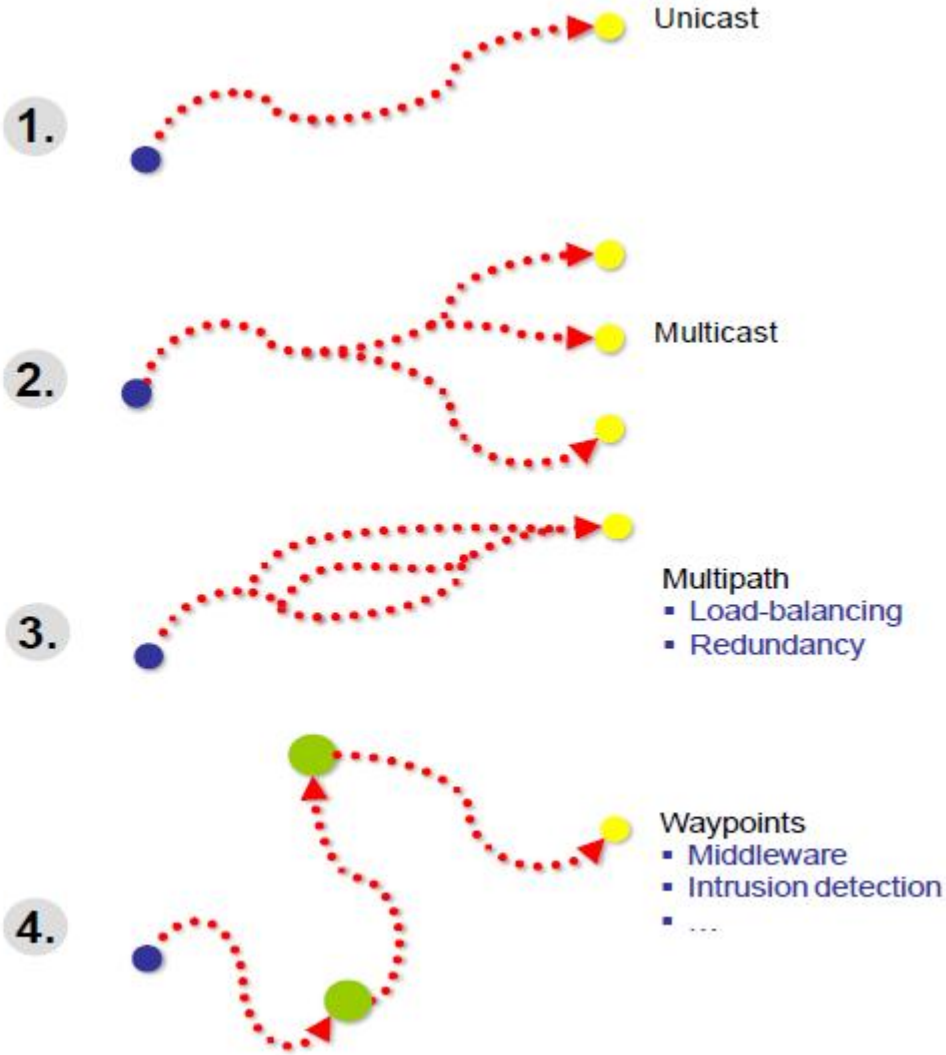


How OpenFlow works

(2) Flow table push type [Proactive]



Flow examples



1. Unicast

2. Multicast

3. Multipath
- Load-balancing
- Redundancy

4. Waypoints
- Middleware
- Intrusion detection
- …

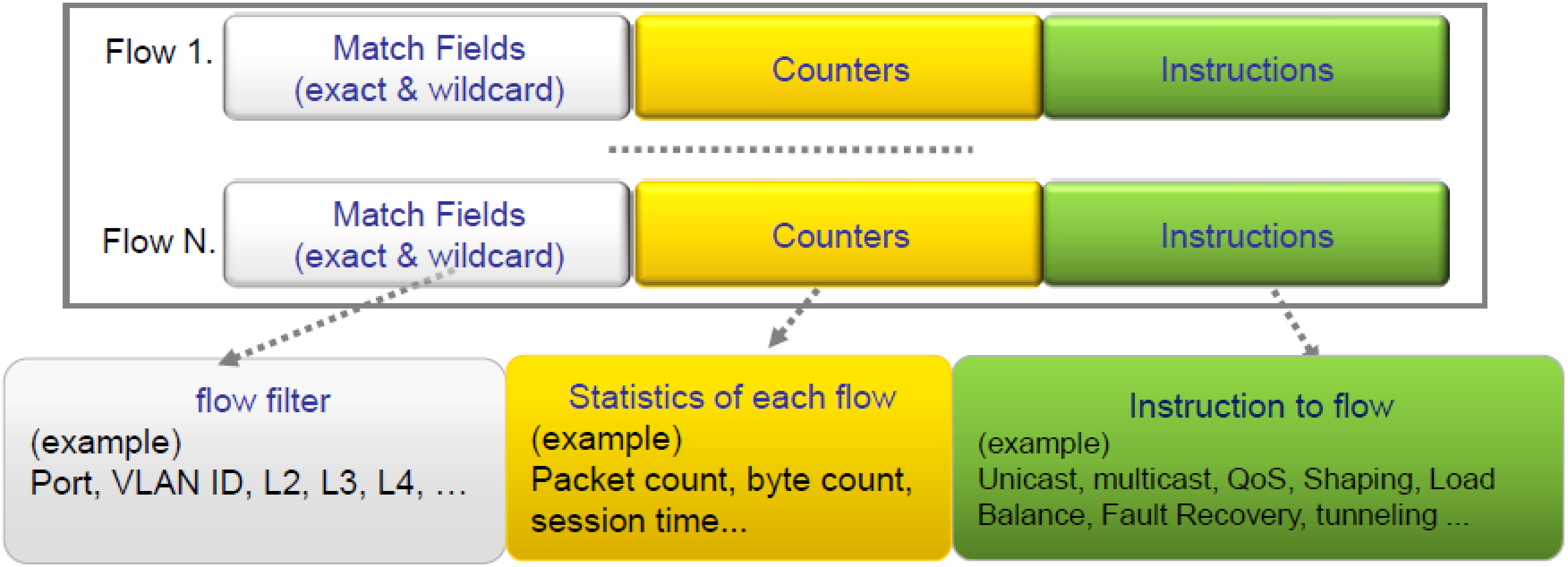
OpenFlow Protocol Detail

- 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 Definition



Flow 1. Match Fields (exact & wildcard) | Counters | Instructions

Flow N. Match Fields (exact & wildcard) | Counters | Instructions

flow filter
(example)
Port, VLAN ID, L2, L3, L4, …

Statistics of each flow
(example)
Packet count, byte count, session time…

Instruction to flow
(example)
Unicast, multicast, QoS, Shaping, Load Balance, Fault Recovery, tunneling …

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

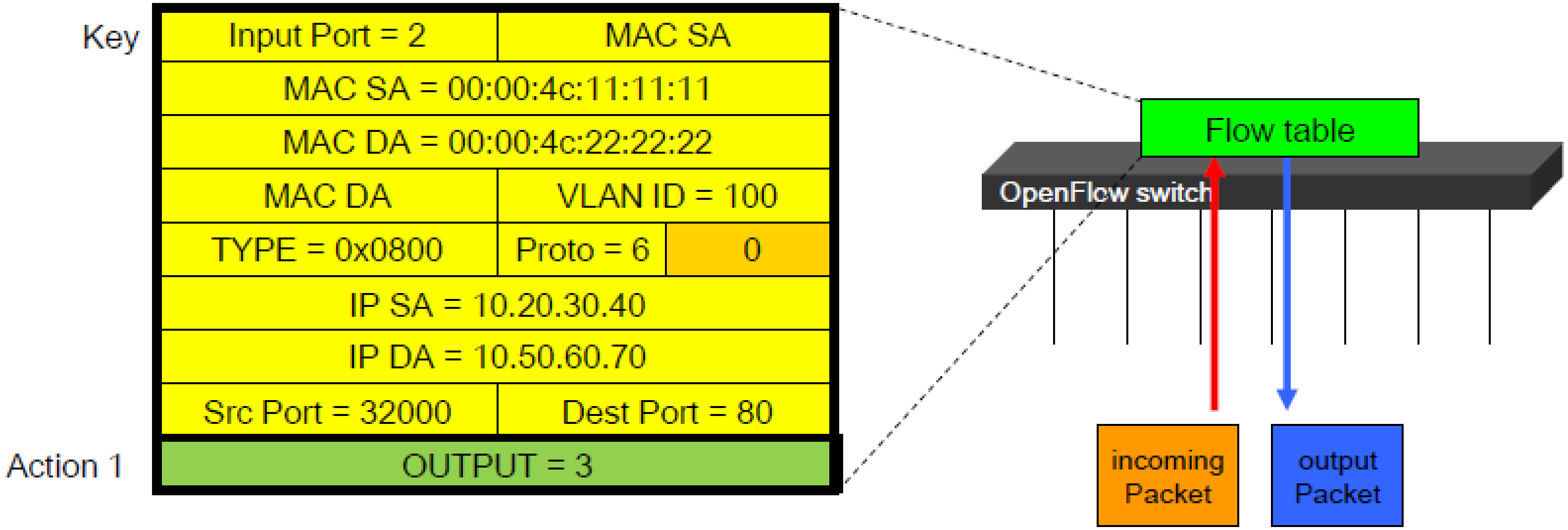
Various type of transferring rules

Possible to modify header

Possible to set multi actions

Example of flow table

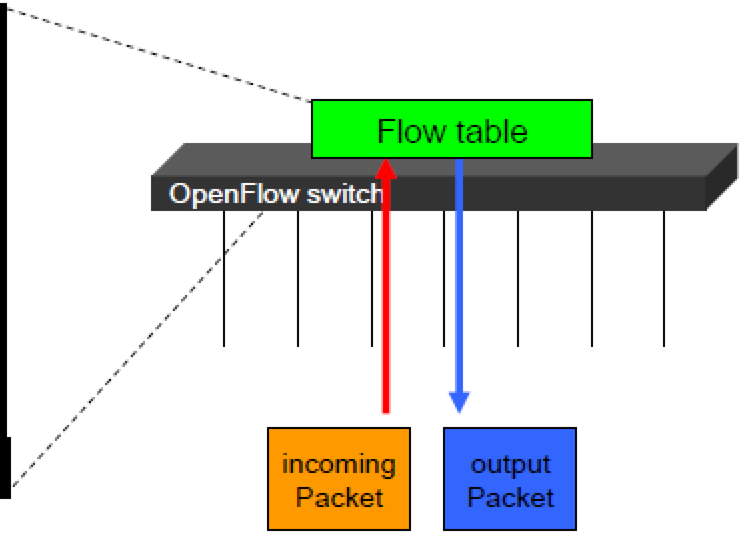
Flow switching



Key	Input Port = 2	MAC SA	
	MAC SA = 00:00:4c:11:11:11		
	MAC DA = 00:00:4c:22:22:22		
	MAC DA	VLAN ID = 100	
	TYPE = 0x0800	Proto = 6	0
	IP SA = 10.20.30.40		
	IP DA = 10.50.60.70		
	Src Port = 32000	Dest Port = 80	
Action 1	OUTPUT = 3		

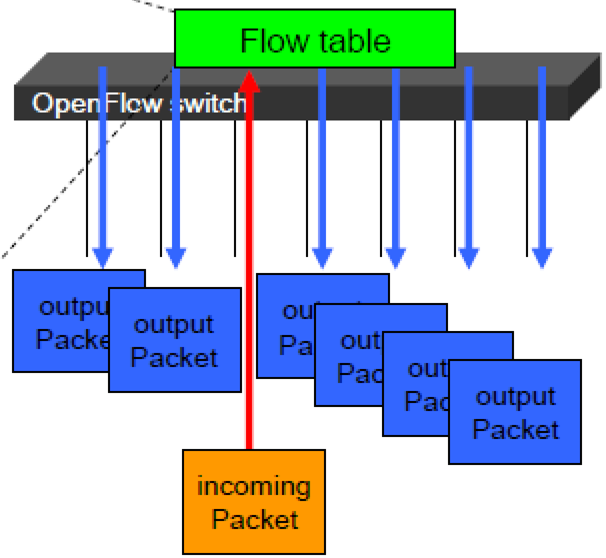
L2 switching

Key	Input Port = *	MAC SA	
	MAC SA = *		
	MAC DA = 00:00:4c:22:22:22		
	MAC DA	VLAN ID = *	
	TYPE = *	Proto = *	0
	IP SA = *		
	IP DA = *		
	Src Port = *	Dest Port = *	
Action 1	OUTPUT = 3		



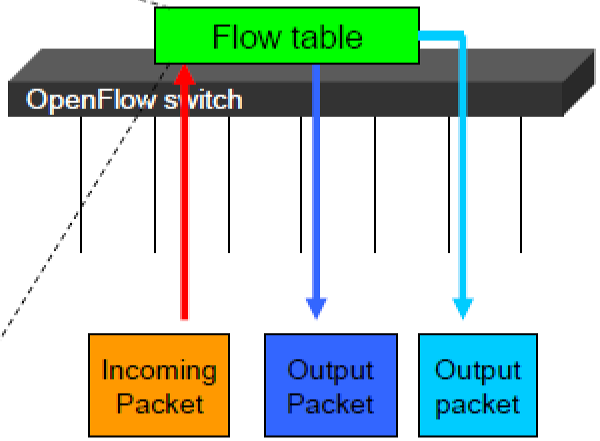
Broadcast

Key	Input Port = *	MAC SA	
	MAC SA = *		
	MAC DA = FF:FF:FF:FF:FF:FF		
	MAC DA	VLAN ID = *	
	TYPE = *	Proto = *	0
	IP SA = *		
	IP DA = *		
	Src Port = *	Dest Port = *	
Action 1	OUTPUT = FLOOD		



multicast

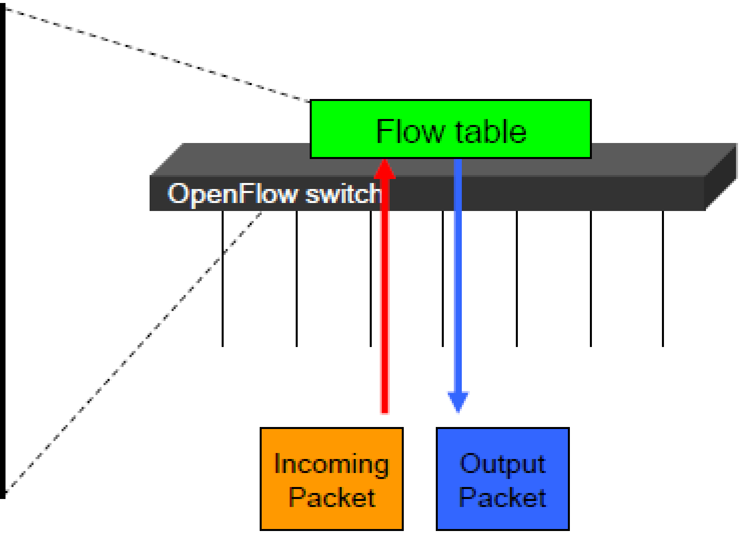
Key	Input Port = 1	MAC SA	
	MAC SA = *		
	MAC DA = *		
	MAC DA	VLAN ID = *	
	TYPE = 0x0800	Proto = *	0
	IP SA = *		
	IP DA = 224.10.10.1		
	Src Port = *	Dest Port = *	
Action 1	SET_DL_SRC = 02:00:4c:00:00:01		
Action 2	OUTPUT = 3		
Action 3	SET_DL_SRC = 02:00:4c:00:00:02		
Action 4	OUTPUT = 5		



SHANGHAI JIAO TONG UNIVERSITY

IP Routing

Key	Input Port = *	MAC SA	
	MAC SA = *		
	MAC DA = *		
	MAC DA	VLAN ID = *	
	TYPE = *	Proto = *	0
	IP SA = *		
	IP DA = 10.20.30.40		
	Src Port = *	Dest Port = *	
Action 1	OUTPUT = 3		



OpenFlow Controller

Open Source

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

Products

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

OpenFlow Implementation

- Hypervisor Mode
  - Open vSwitch (OVS): XEN, KVM, …
  - OVS other features: security, visibility, QoS, automated control
- Hardware Mode
  - OpenFlow Switch
  - Hop by hop configuration

BROCADE
extreme networks
hp
IBM
NEC
NETGEAR
Connect with Innovation

Reality Check

- “OpenFlow doesn’t let you do anything you couldn’t do on a network before” –Scott Shenker (Professor, UC Berkeley, OpenFlow co-inventor)
- Frames are still forwarded, packets are delivered to hosts.
- OpenFlow 1.3 was recently approved.
- Major vendors are participating - Cisco, Juniper, Brocade, Huawei, Ericsson, etc. It’s still early stage technology but commercial products are shipping.
- OpenFlow led by large companies Google/Yahoo/Verizon and lack of focus on practical applications in the enterprise.

OpenFlow Interop

- Fifteen Vendors Demonstrate OpenFlow Switches at Interop (May 8-12, 2011)

big switch networks

BROADCOM.

BROCADE

CiTRiX

DELL

extreme networks

FULCRUM microsystems

hp

IBM

JUNIPer NETWORKS

NEC

NETGEAR Connect with Innovation

NetOptics

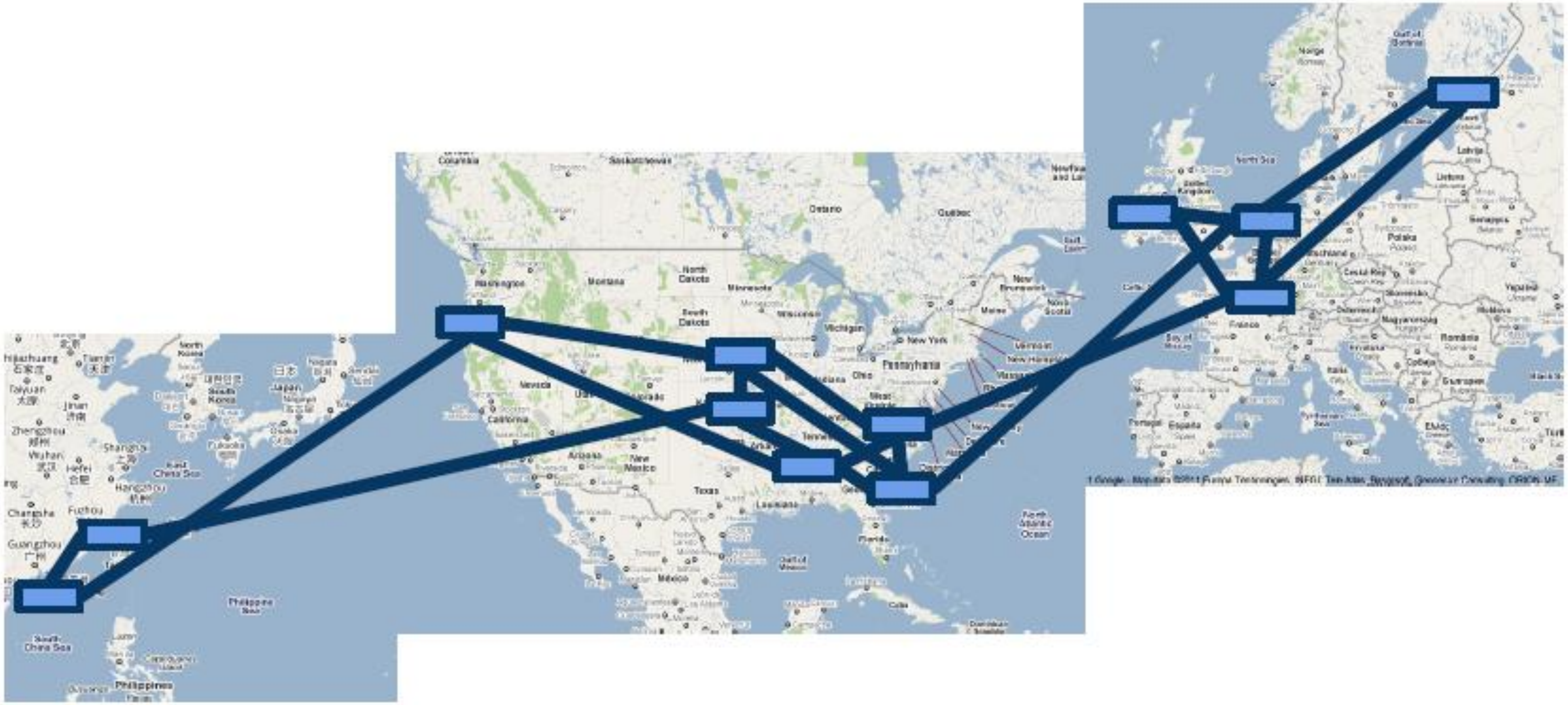
OPNET

pronto

Google's WAN

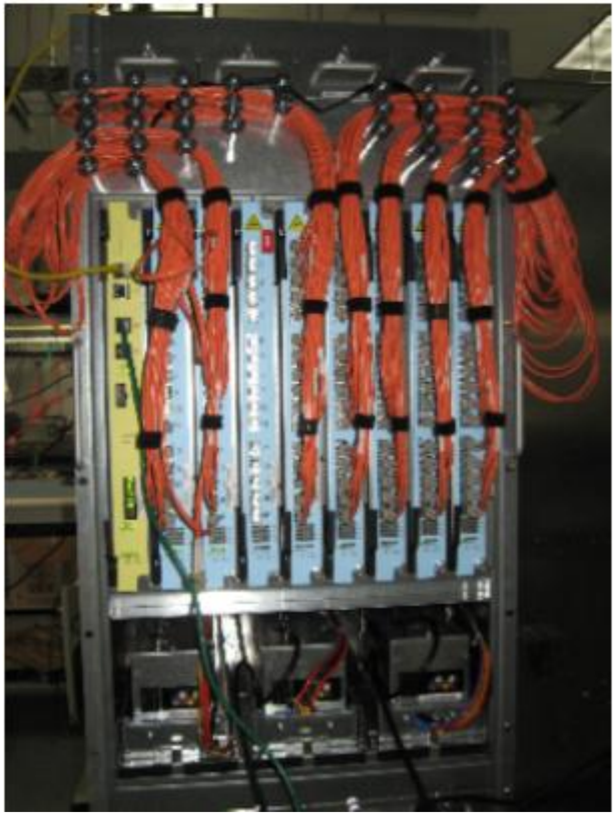
- Two backbones
  - Internet facing (user traffic)
  - Datacenter traffic (internal)
- Widely varying requirements: loss sensitivity, availability, topology, etc.
- Widely varying traffic characteristics: smooth/diurnal vs. bursty/bulk
- Therefore: built two separate logical networks
  - I-Scale (bulletproof)
  - G-Scale (possible to experiment)

Google's OpenFlow WAN

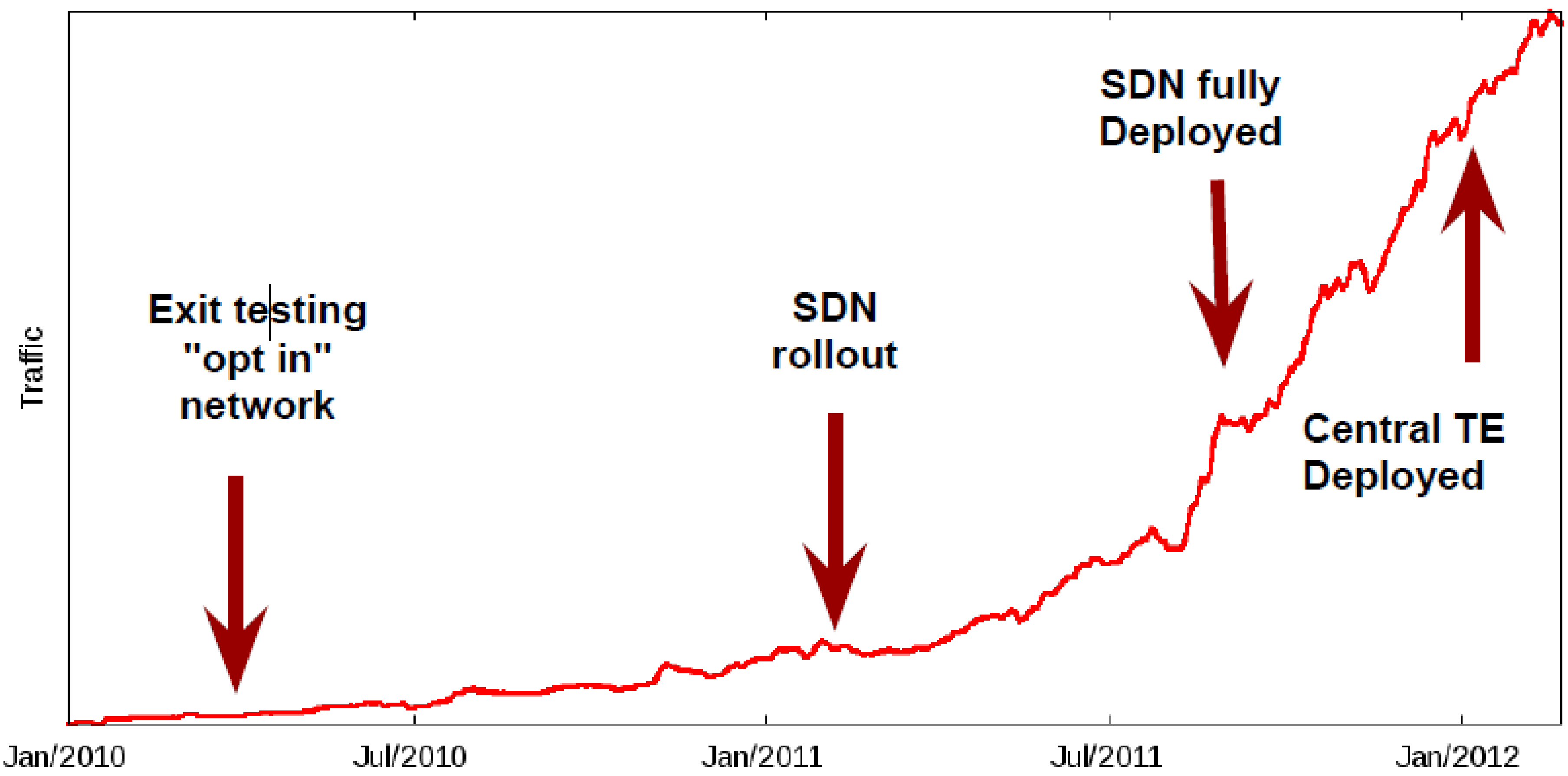


G-Scale Network Hardware

- Built from merchant silicon
  - 100s of ports of nonblocking 10GE
- OpenFlow support
- Open source routing stacks for BGP, ISIS
- Does not have all features
  - No support for AppleTalk...
- Multiple chassis per site
  - Fault tolerance
  - Scale to multiple Tbps

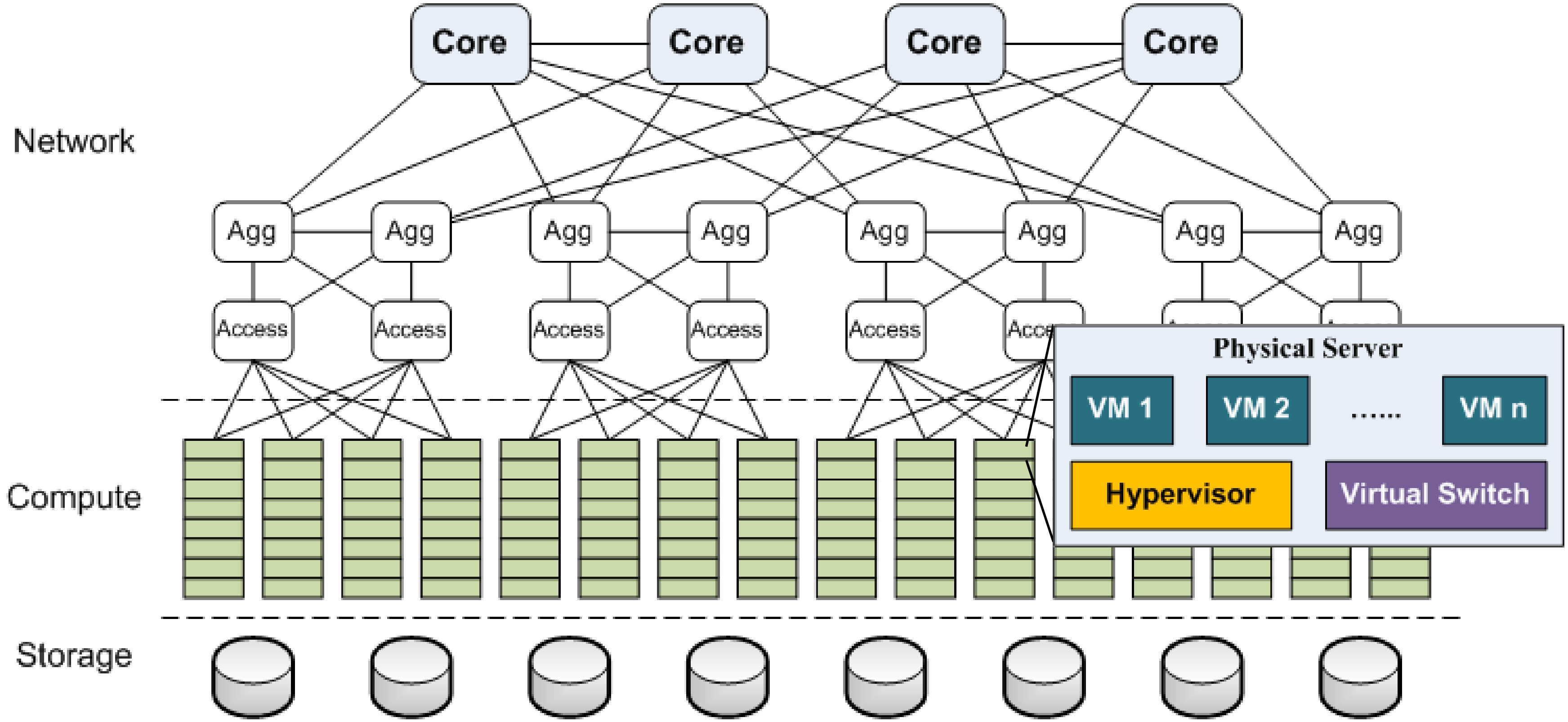


G-Scale WAN Usage



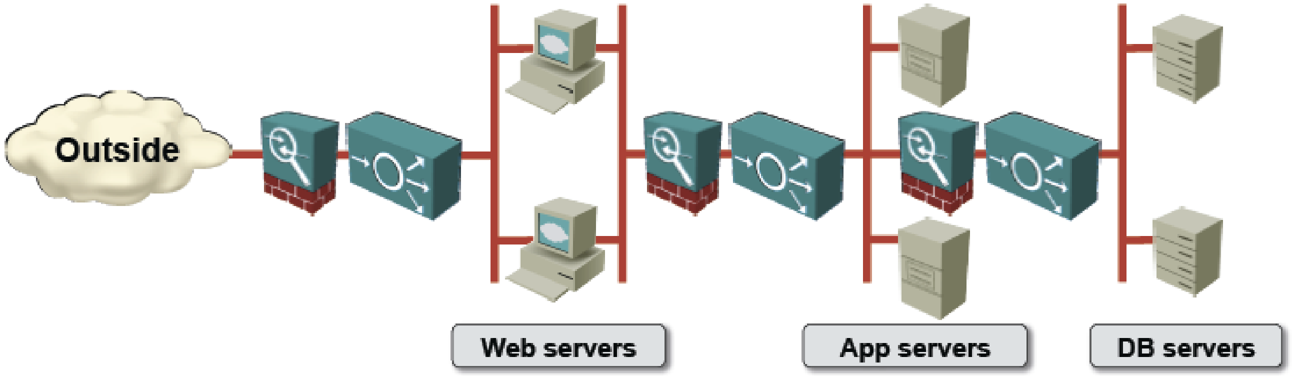
Network Virtualization

General Data Center Architecture

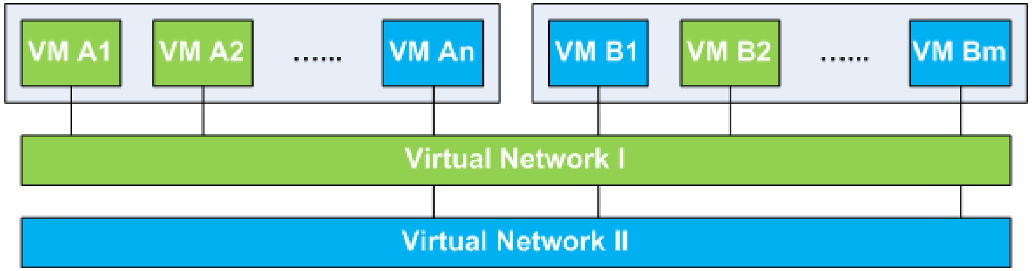


Cloud management system allows us dynamically provisioning VMs and virtual storage.

What customers really want?



Virtual Network



- Requirements
  - Multiple logical segments
  - Multi-tie applications
  - Load balancing and firewalling
  - Unlimited scalability and mobility

Multi-Tenant Isolation

- Making life easier for the cloud provider
  - Customer VMs attached to “random” L3 subnets
  - VM IP addresses allocated by the IaaS provider
  - Predefined configurations or user-controlled firewalls
- Autonomous tenant address space
  - Both MAC and IP addresses could overlap between two tenants, or even within the same tenant
  - Each overlapping address space needs a separate segment

Scalability

- Datacenter networks have got much bigger (and getting bigger still !!)
  - Juniper’s Qfabric ~6000 ports, Cisco’s FabricPath over 10k ports
- Tenant number dramatically increase as the IaaS experiences rapid commoditization
  - Forrester Research forecasts that public cloud today globally valued at $2.9B, projected to grow to $5.85B by 2015.
- Server virtualization increase demand on switch MAC address tables
  - Physical with 2 MACs -> 100 VMs with 2 vNIC need 200+ MACs!

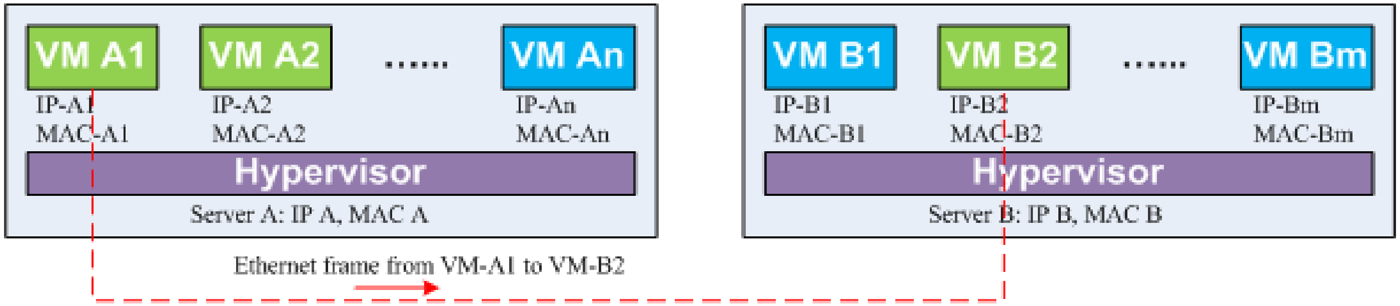
Possible Solutions (1)

- VLANs per tenant
  - limitations of VLAN-id range (Only 12bits ID = 4K)
  - VLAN trunk is manually configured
  - Spanning tree limits the size of the network
- L2 over L2
  - vCDNI(VMware), Provider Bridging(Q-in-Q)
  - Limitations in number of users (limited by VLAN-id range)
  - Proliferation of VM MAC addresses in switches in the network (requiring larger table sizes in switches)
  - Switches must support use of same MAC address in multiple VLANs (independent VLAN learning)

Possible Solutions (2): L2 over IP

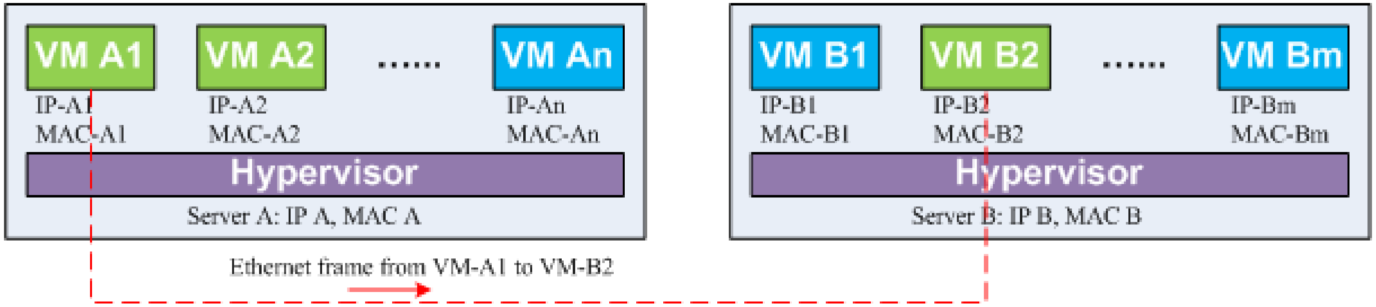
- Virtual eXtensible LAN (VXLAN)
  - VMware, Arista, Broadcom, Cisco, Citrix, Red Hat
  - VXLAN Network Identifier (VNI): 24 bits = 16M
  - UDP encapsulation, new protocol
- Network Virtualization Generic Routing Encapsulation (NVGRE)
  - Microsoft, Arista, Intel, Dell, HP, Broadcom, Emulex
  - Virtual Subnet Identifier (VSID): 24 bits = 16M
  - GRE tunneling, relies on existing protocol
- Stateless Transport Tunneling (STT)
  - Nicira
  - Context ID: 64 bits, TCP-like encapsulation

VXLAN/NVGRE: How it Works?



without overlay

D-MAC: MAC-B2	**S-MAC**: MAC-A1	**D-IP**: IP-B2	**S-IP**: IP-A1	**Payload**
Ethernet Header		IP Header		



using VXLAN

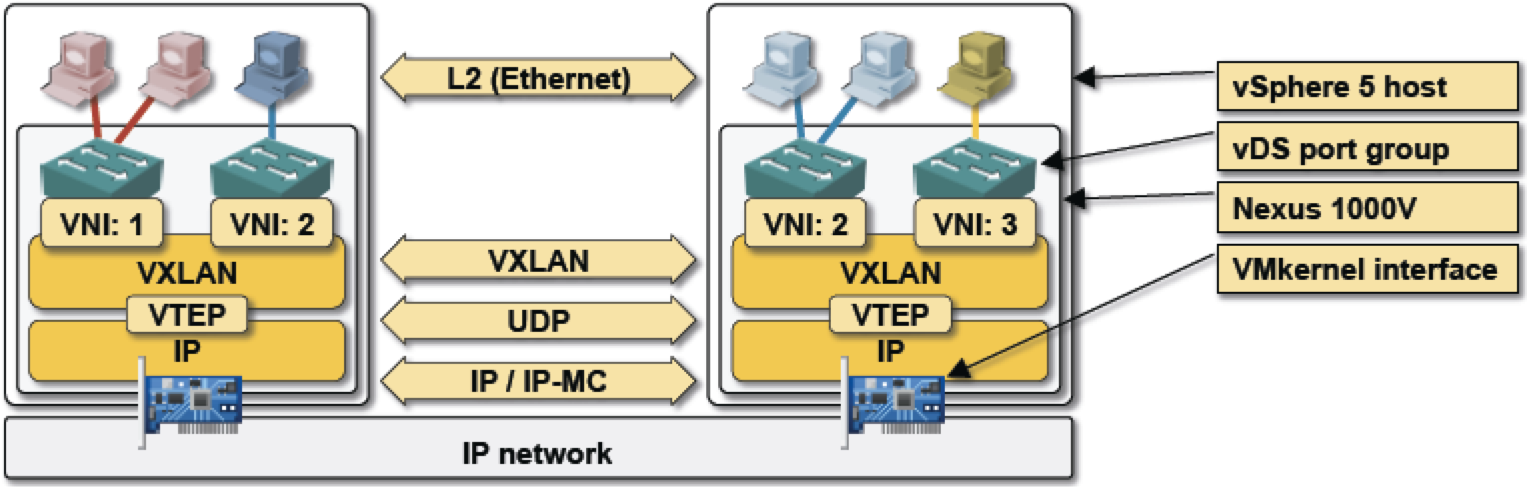
D-MAC: MAC-B	**S-MAC**: MAC-A	**D-IP**: IP-B	**S-IP**: IP-A	**UDP**	**VNI**	**D-MAC**: MAC-B2	**S-MAC**: MAC-A1	**D-IP**: IP-B2	**S-IP**: IP-A1	**Payload**
Outer Ethernet Header		Outer IP Header				Inner Ethernet Header		Inner IP Header		

D-MAC: MAC-B	**S-MAC**: MAC-A	**D-IP**: IP-B	**S-IP**: IP-A	**GRE**	**VSID**	**D-MAC**: MAC-B2	**S-MAC**: MAC-A1	**D-IP**: IP-B2	**S-IP**: IP-A1	**Payload**
Outer Ethernet Header		Outer IP Header				Inner Ethernet Header		Inner IP Header		

using NVGRE

Dynamic MAC learning

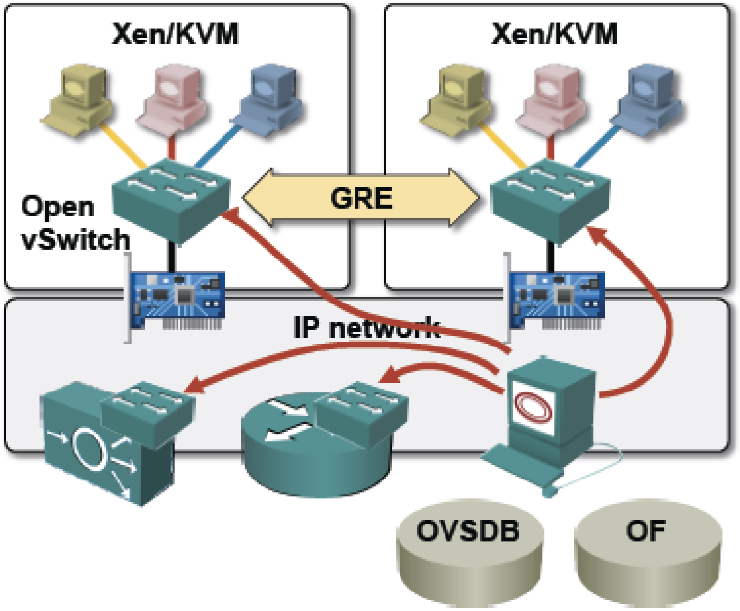
- Dynamic MAC learning with L2 flooding over IP multicasting



Flooding does not scale when fabric gets bigger.

Control Plane (Nicira)

- L2-over-IP with control plane
  - OpenFlow-capable vSwitches
  - IP tunnels (GRE, STT ...)
  - MAC-to-IP mappings by OpenFlow
  - Third-party physical devices
- Benefits
  - No reliance on flooding
  - No IP multicast in the core



Transitional Strategy Depends on Your Business

- 100s tenants, 100s servers: VLANs
- 1000s tenants, 100s servers: vCDNI or Q-in-Q
- Few 1000s servers, many tenants: VXLAN/NVGRE/STT
- More than that: L2 over IP with control plane

Open question: How to solve the co-existing scenarios in one cloud?