

Ethernet-based Software Defined Network (SDN)

**Cloud Computing Research Center for
Mobile Applications (CCMA), ITRI**

雲端運算行動應用研究中心



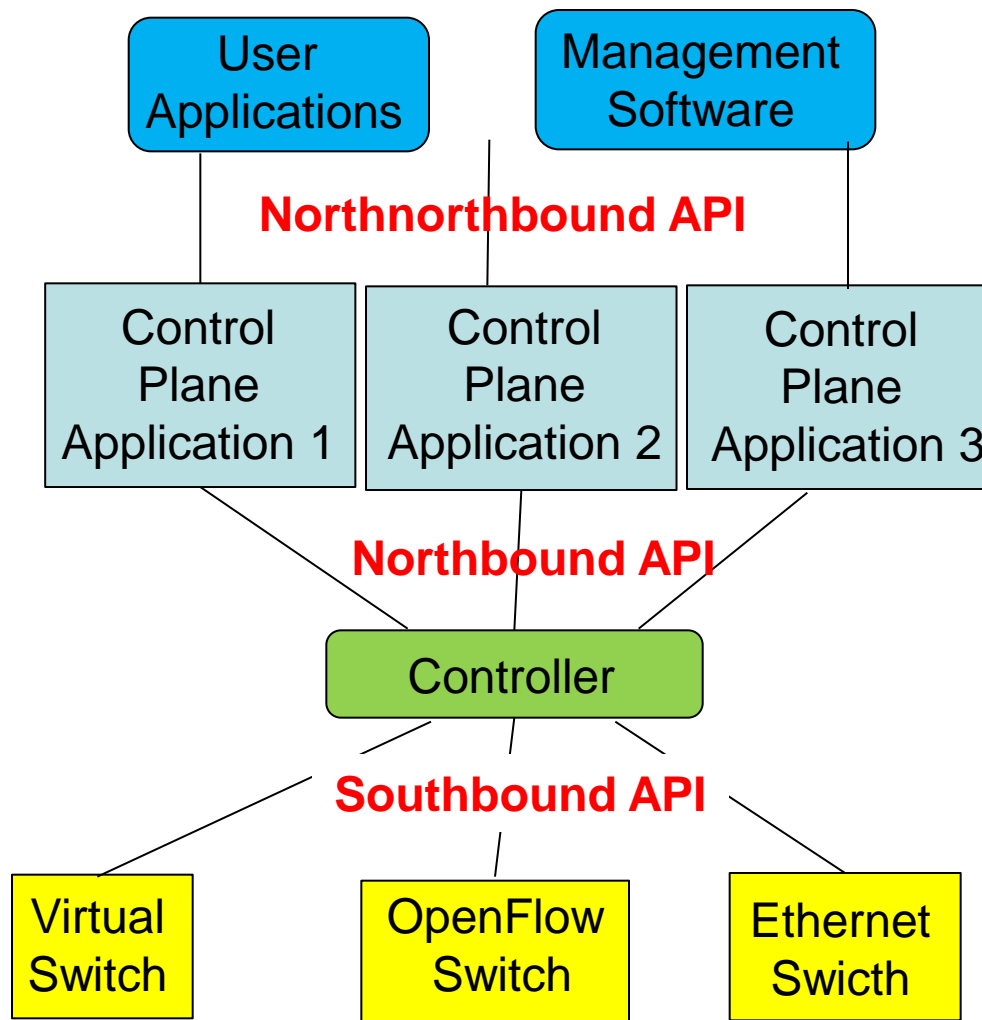
SDN Introduction

- Decoupling of control plane from data plane
 - Breaks the existing vertical integration model in network equipment industries
 - Enables quick development and deployment of innovative third-party control plane applications
- Centralization of control plane
 - Management vs. Control vs. Data plane
 - Facilitates fast deployment and upgrade of control protocols
 - Enables global allocation of a network's resources
- Software in software-defined network
 - User applications
 - Control protocols



SDN Software Architecture

- Data plane
- Controller
- Control plane applications
- Southbound AP
- Northbound API
- NorthNorthbound API





OpenFlow Architecture

- **OpenFlow switch**: A data plane that implements a set of flow rules specified in terms of the OpenFlow instruction set
- **OpenFlow controller**: A control plane that sets up the flow rules in the flow tables of OpenFlow switches
- **OpenFlow protocol**: A secure protocol for an OpenFlow controller to set up the flow tables in OpenFlow switches



OpenFlow Controller

OpenFlow Protocol (SSL/TCP)



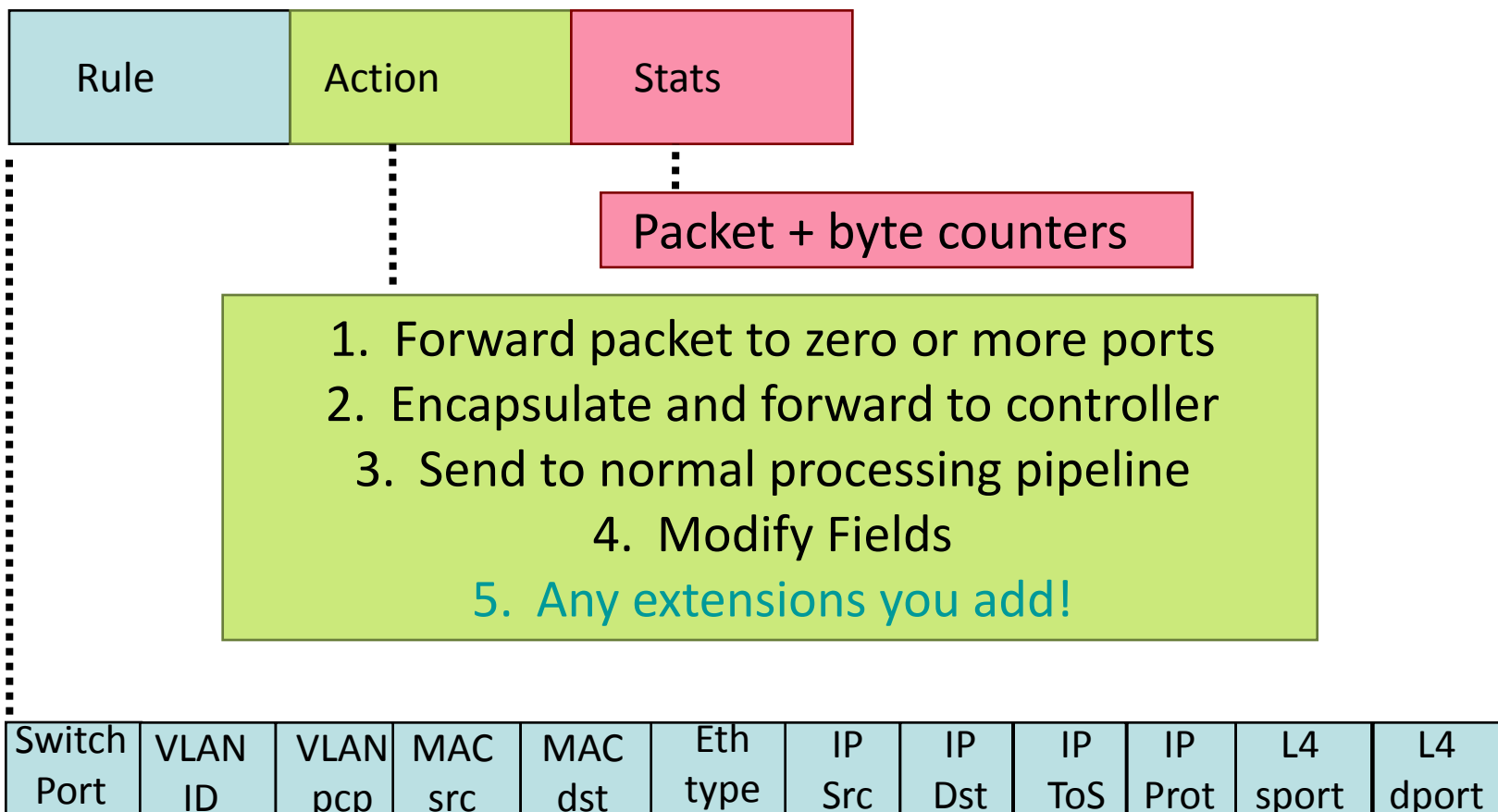
Control Path

Flow Table

Data Path (Hardware)



OpenFlow Basics



+ mask what fields to match



Examples

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

Flow Switching

Switch Port	MAC src	MAC dst	Eth type	VLAN ID	IP Src	IP Dst	IP Prot	TCP sport	TCP dport	Action
port3	00:20..	00:1f..	0800	vlan1	1.2.3.4	5.6.7.8	4	17264	80	port6

Firewall

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



Examples

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

VLAN Switching

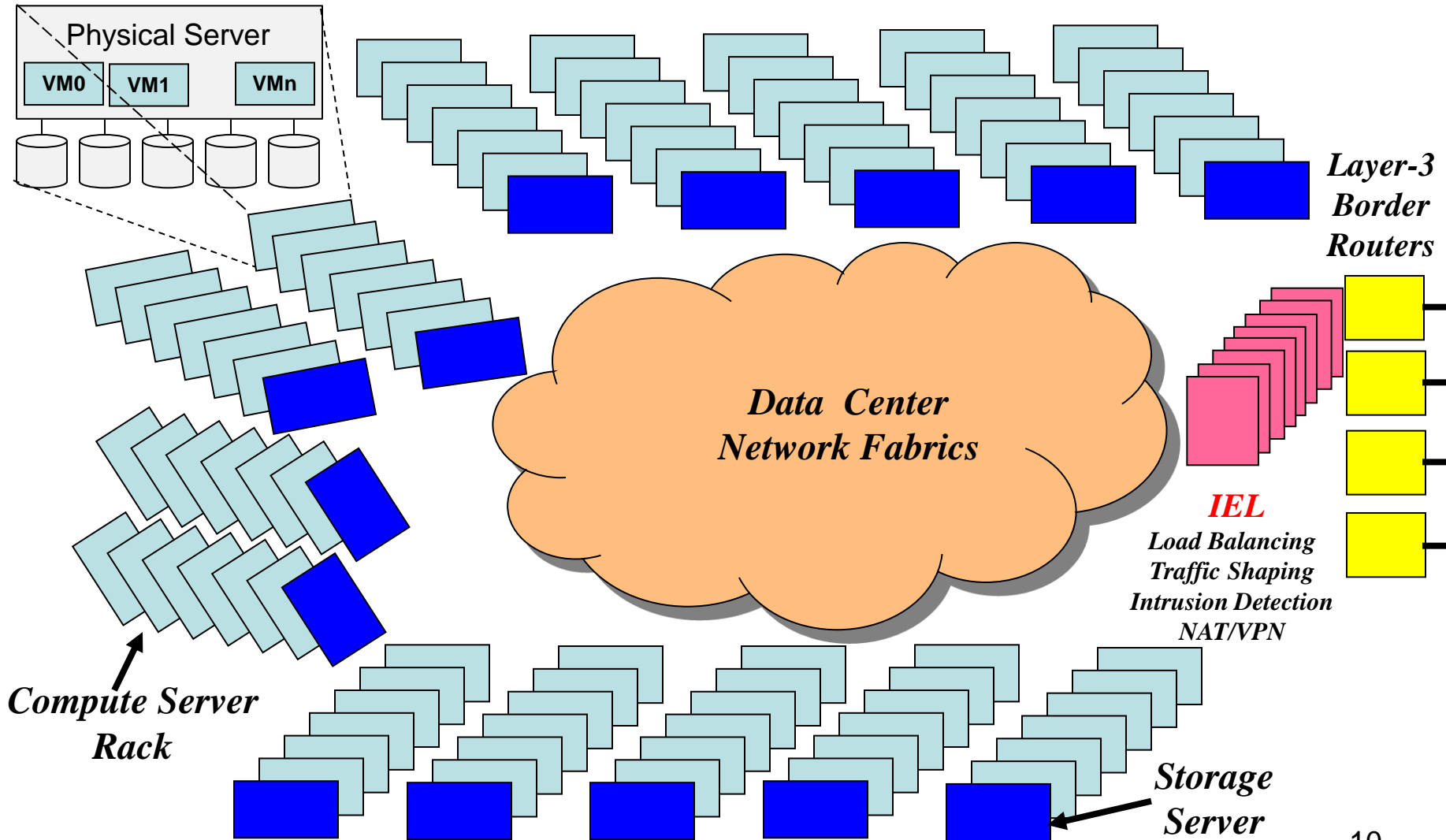
Switch Port	MAC src	MAC dst	Eth type	VLAN ID	IP Src	IP Dst	IP Prot	TCP sport	TCP dport	Action
*	*	00:1f..	*	vlan1	*	*	*	*	*	port6, port7, port9



SDN \neq OpenFlow

- Can we apply SDN to Ethernet switches, especially in cloud data center space? **Peregrine**

Cloud Data Center Architecture





Cloud Data Center Network

- Cloud data centers are **Big** and **Shared**
- **Scalable and available data center fabrics**
 - Not all links are used
 - No load-sensitive routing
 - Fail-over latency is high (> 5 seconds)
- **Network virtualization**: Each virtual data center (VDC) gets to define its own network
 - VMs in a VDC belong to one or multiple subnets (broadcast domains)
 - Each VDC has its own private IP address space
 - Each VDC has a set of public IP addresses for service entry points and for VPN connections with external sites
 - Each VDC has its Internet traffic shaping policy, intra-VDC and inter-VDC firewalling policy, and server load balancing policy



What's Wrong with Ethernet?

- Spanning tree-based
 - Not all physical links are used
 - No load-sensitive dynamic routing
 - Fail-over latency is high (> 5 seconds)
- Cannot scale to a large number of VMs (e.g. 1M)
 - Forwarding table is too small: 16K to 64K
- Does not support VM migration and visibility
- Does not support network virtualization

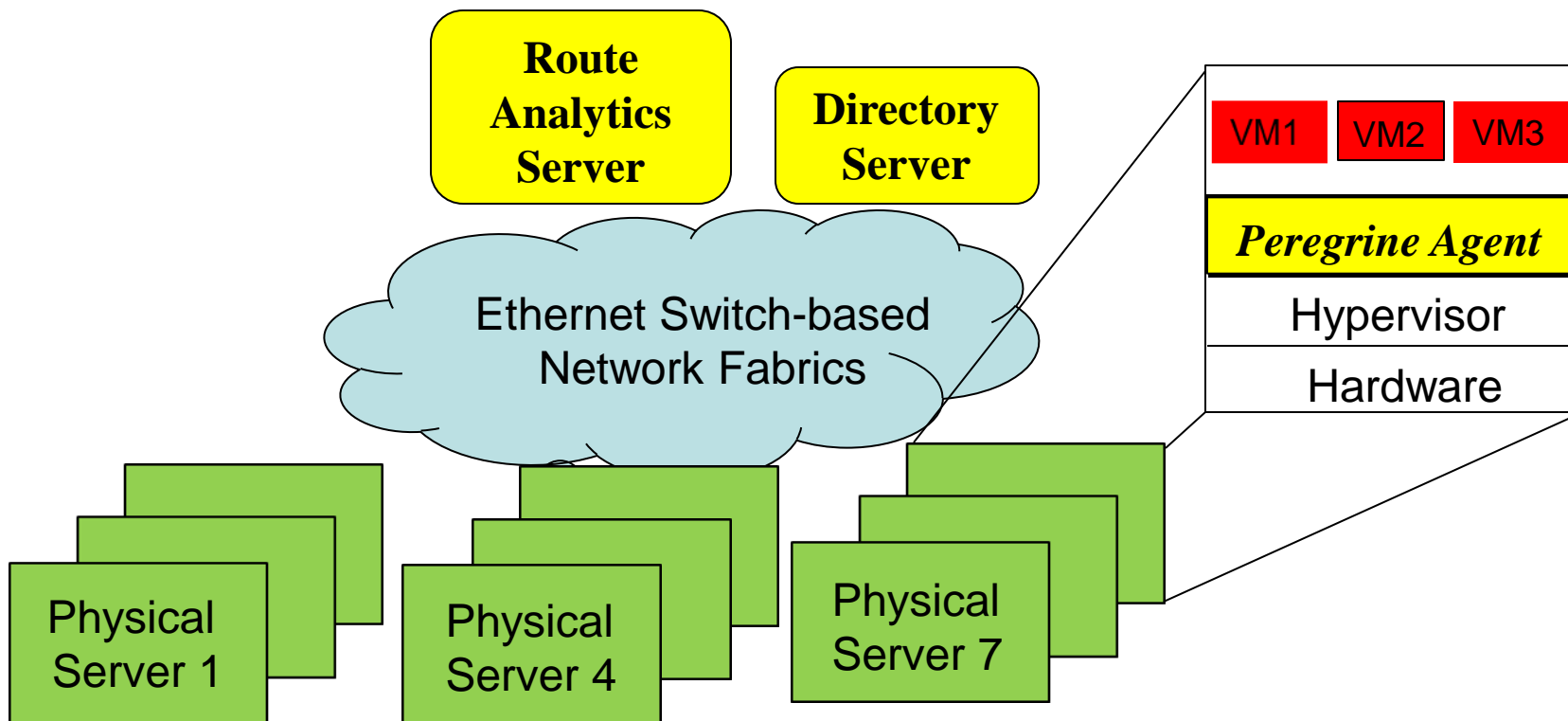


Peregrine

- A unified Layer-2-only data center network for LAN and SAN traffic
- A SDN architecture using only **commodity** Ethernet switches:
centralized control plane and **distributed** data plane
- Turn off Ethernet's control protocol: spanning tree, source learning, flooding of unknown-destination-MAC packets, broadcast of ARP and DHCP
 - VLAN is optional because VLAN ID space is limited
- Centralized load-balancing routing using real-time traffic matrix
- Fast fail-over using pre-computed primary/backup routes
- Native support for network virtualization
 - Private IP address space reuse
 - Multiple subnets per virtual network



Peregrine Software Architecture



Dynamic Traffic Engineering

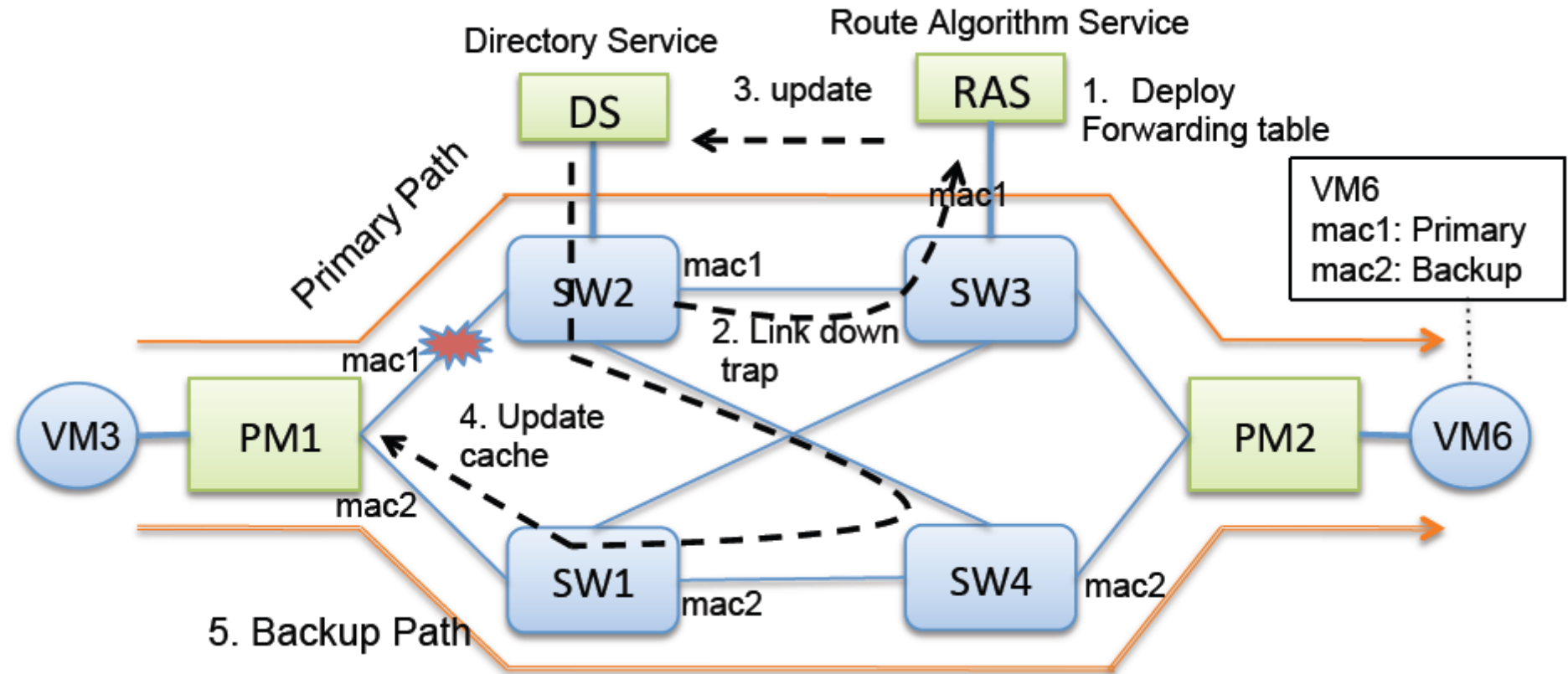
- Periodic collection of real-time traffic matrix
 - Traffic volume between each pair of VMs
 - Traffic volume between each pair of PMs
- Load balancing routing algorithm
 - Loads on the physical links
 - Number of hops
 - Forwarding table entries
 - Prioritization
- Computed routes are programmatically installed on the forwarding tables of Ethernet switches



Fast Failure Recovery

- For a given node N, proactively computes two disjoint paths (primary and backup) from every other node to N, and installs them on the associated switches
- All primary (secondary) paths to a given node N form a spanning tree
- Each node has two MAC addresses, which are installed on the switches of its primary and backup trees, respectively
- When a primary path from S to D goes down, S is notified that it should use D's backup MAC address to reach D
- RAS and DS support fail-over

Fast Failure Recovery Example





Network Virtualization

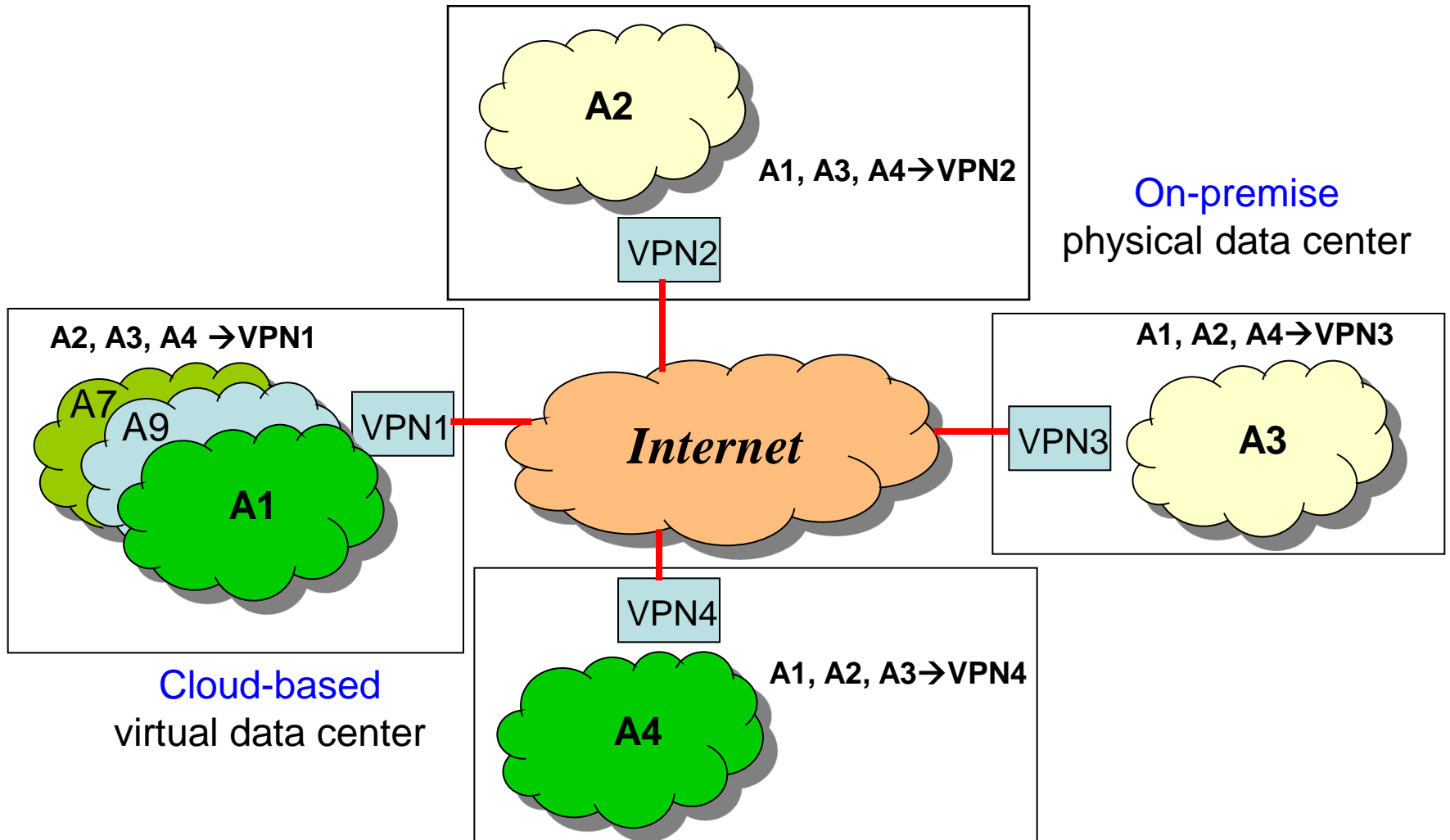
- Multiple virtual networks running on a single physical network
- The network of each virtual data center (VDC) consists of
 - VMs' MAC addresses are pre-assigned
 - A single layer-2 network
 - A complete private IP address space, organized into multiple subnets each with its own broadcast domain
 - A set of public IP addresses
 - Its own copy of the DHCP and DNS service
 - **Security**: Intra-VDC and inter-VDC firewall policy
 - **SLA**: Traffic shaping policy
 - **Scalability**: Server load balancing policy



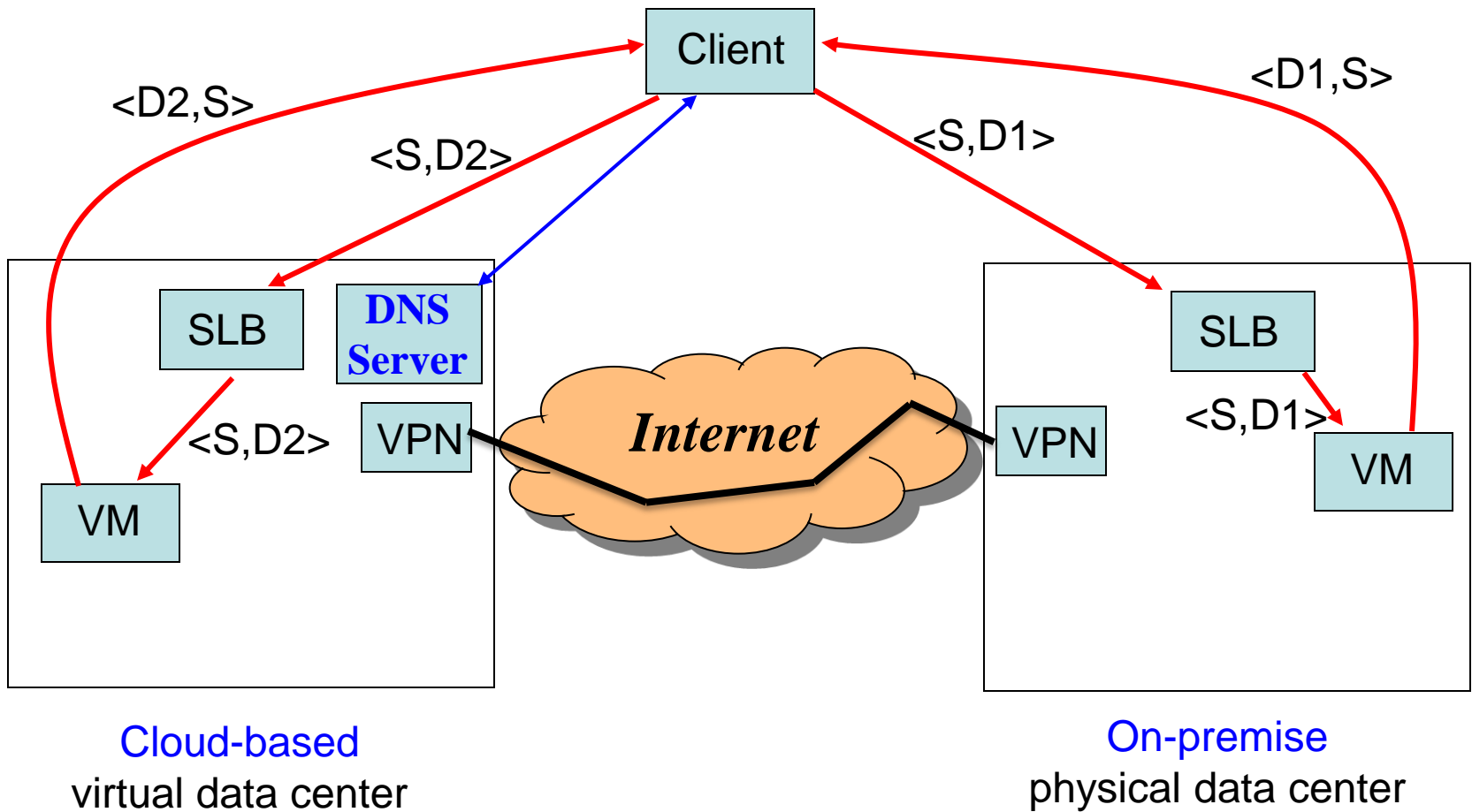
Private IP Address Space Reuse

- Requirement: Every VDC has a VDC ID and its own full 24-bit private IP address space (10.x.x.x), even though multiple VDCs run on top of the same data center network
- Two approaches:
 - Ethernet over TCP/UDP:
 - Every Ethernet packet is encapsulated inside an TCP/UDP packet or TCP/UDP connection as an Ethernet link
 - Needs to implement in software such Ethernet switch functions as source learning, flooding, VLAN, etc.
 - Can work with arbitrary IP networks
 - Multi-tenancy-aware IP-MAC mapping: our approach
 - VDC ID + private IP address → MAC address
 - Runs directly on L2 networks, no need for Ethernet switch emulation
 - Inter-virtual-data-center isolation

Hybrid Cloud Support



Cross-Site Global Load Balancing





Peregrine in SDN Framework

- **Data plane:** Ethernet switches
- **Southbound API:** SNMP and CLI
- **Controller:** (1) Physical network resource set-up, (2) Physical topology record keeping, (3) SNMP trap processing, (4) Ethernet switch configuration, including forwarding table programming, and (5) Traffic load information
- **Northbound API:**
 - Failure/congestion notification, including SNMP trap packet delivery
 - ARP request packet delivery
 - Forwarding table programming
 - Physical topology/traffic load querying
- **Control plane applications:**
 - Dynamic traffic engineering
 - Fast fail-over for data/control plane failures
 - Network virtualization



Wish-List Ethernet Switch Features

- Ability to turn off flooding of unicast packets with unknown destination MAC address
- Ability to turn off source MAC address check
→ enables asymmetric routing
- Bulk uploading of forwarding table
- Fast link/switch failure detection
- Link traffic load counting
- Interception and redirection of packets of selected type



Comparisons

- Scalable and available data center fabrics
 - IEEE 802.1aq: Shortest Path Bridging
 - IETF TRILL
 - Competitors: Cisco, Juniper, Brocade
 - **Differences**: commodity switches, centralized load balancing routing and proactive backup route deployment
- Network virtualization
 - OpenFlow protocol: between OpenFlow controller and OpenFlow switches
 - Competitors: Nicira, NEC and NTT
 - Generality carries a steep performance price
 - Every virtual network link is a tunnel
 - **Differences**: Simpler and more efficient because it runs on L2 switches directly
 - Flow table state management and flow table lookup performance overhead



Current Status

- A fully operational Peregrine prototype that works on a 10-switch and 100-server test-bed
 - Start-up and shut-down without out-of-band control network
 - Fast fail-over for both data plane and control plane failures
 - Dynamic traffic engineering
- On-going work:
 - An agentless implementation of Peregrine
 - Encapsulate Peregrine with Internet Edge Logic functionalities as a Quantum plug-in implementation
 - Port dynamic traffic engineering, fast fail-over, and network virtualization to an OpenFlow platform, i.e. OF controller + OF switches



Summary

- Peregrine is a **network system** technology, not a network device technology, and consists of
 - A hypervisor agent running on every compute node
 - L7/Web application firewall and traffic shaping
 - A centralized route server and directory server
 - A VDC-aware Internet Edge Logic cluster
 - Server load balancing, VPN, NAT, and L4 firewall
- Network virtualization technology that does **not** use tunneling or VLAN
- A software defined network (SDN) architecture that runs on commodity Ethernet switches, and is able to manage **both** legacy Ethernet and OpenFlow switches



Thank You!

For More Information:

Y.F. Juan / 阮耀飛

Deputy Director

Strategy & Business Development

Cloud and Mobile Computing Center

Industrial Technology Research Institute

t: +886 (0) 3.591.6173

m: +886 (0) 975.876.919

e: yf.juan@itri.org.tw