Software Defined Network Based on Ethernet-in-Core OpenFlow-at-Edge (ECOE) Network Architecture

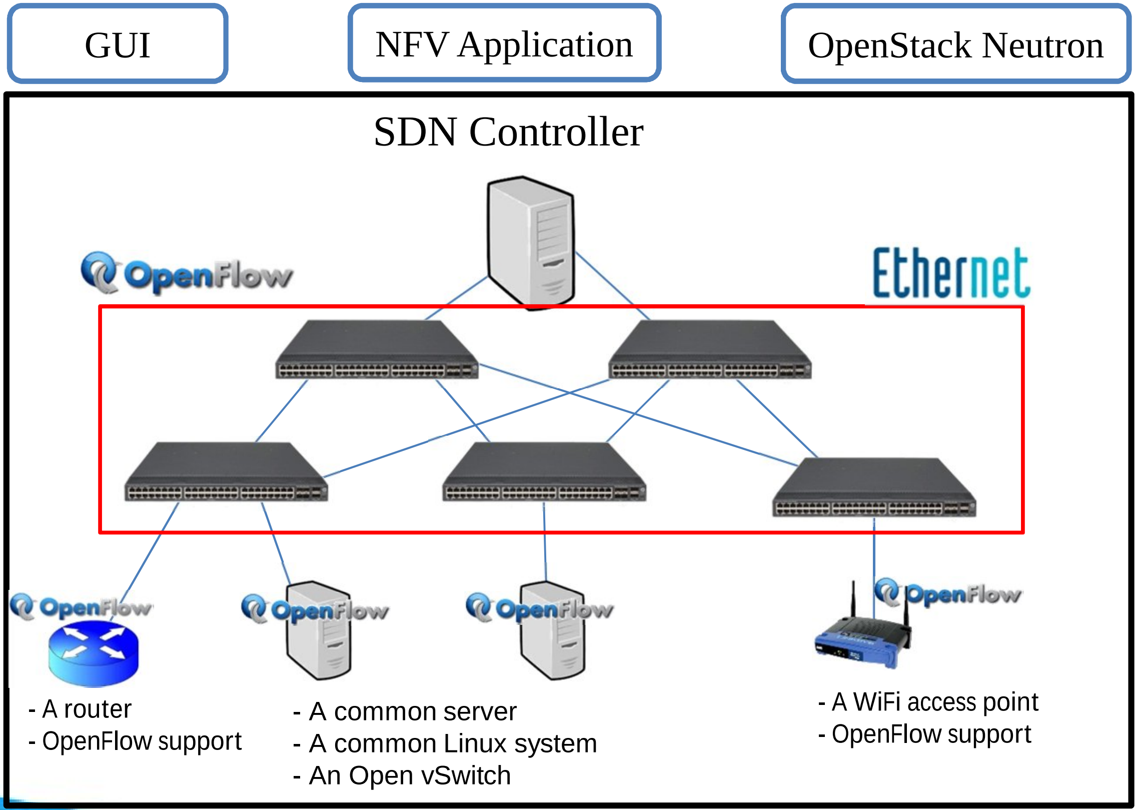
Ray Lee

Industrial Technology Research Institute (ITRI)

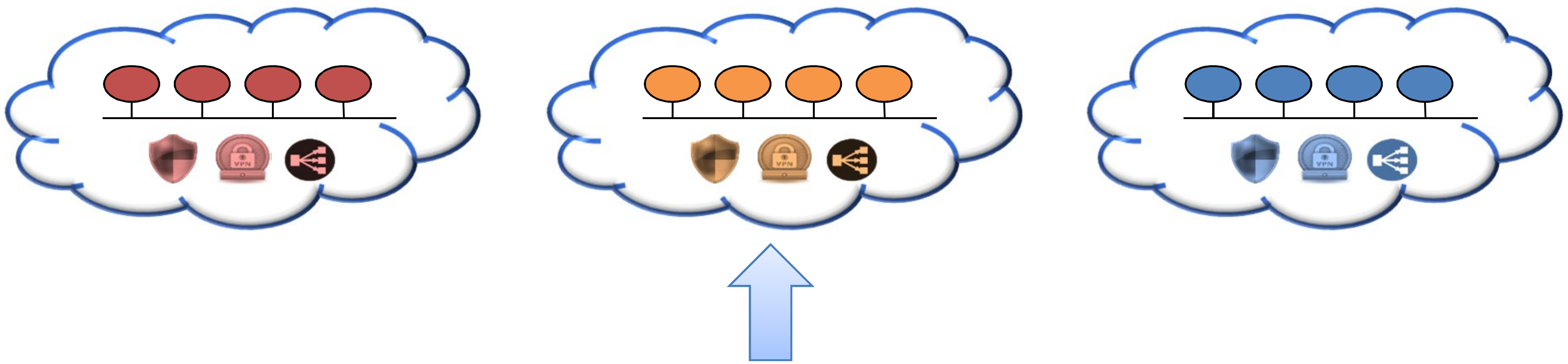
Information and Communication Research Lab (ICL)

What is ECOE Network Architecture

- ECOE Network Architecture
  - Ethernet-in-Core OpenFlow-at-Edge
  - Provides intelligent Edge by edge OpenFlow devices
  - Provides high speed and reliable packet delivery by core Ethernet devices.
  - “It ’s time for SDNv2”, Scott Shenker said. (2014.10)



Peregrine Hybrid SDN Solution



Peregrine hybrid SDN solution

ITRI contributes SNMP4SDN plugin to OpenDaylight, the plugin use SNMP and CLI to control Ethernet switches

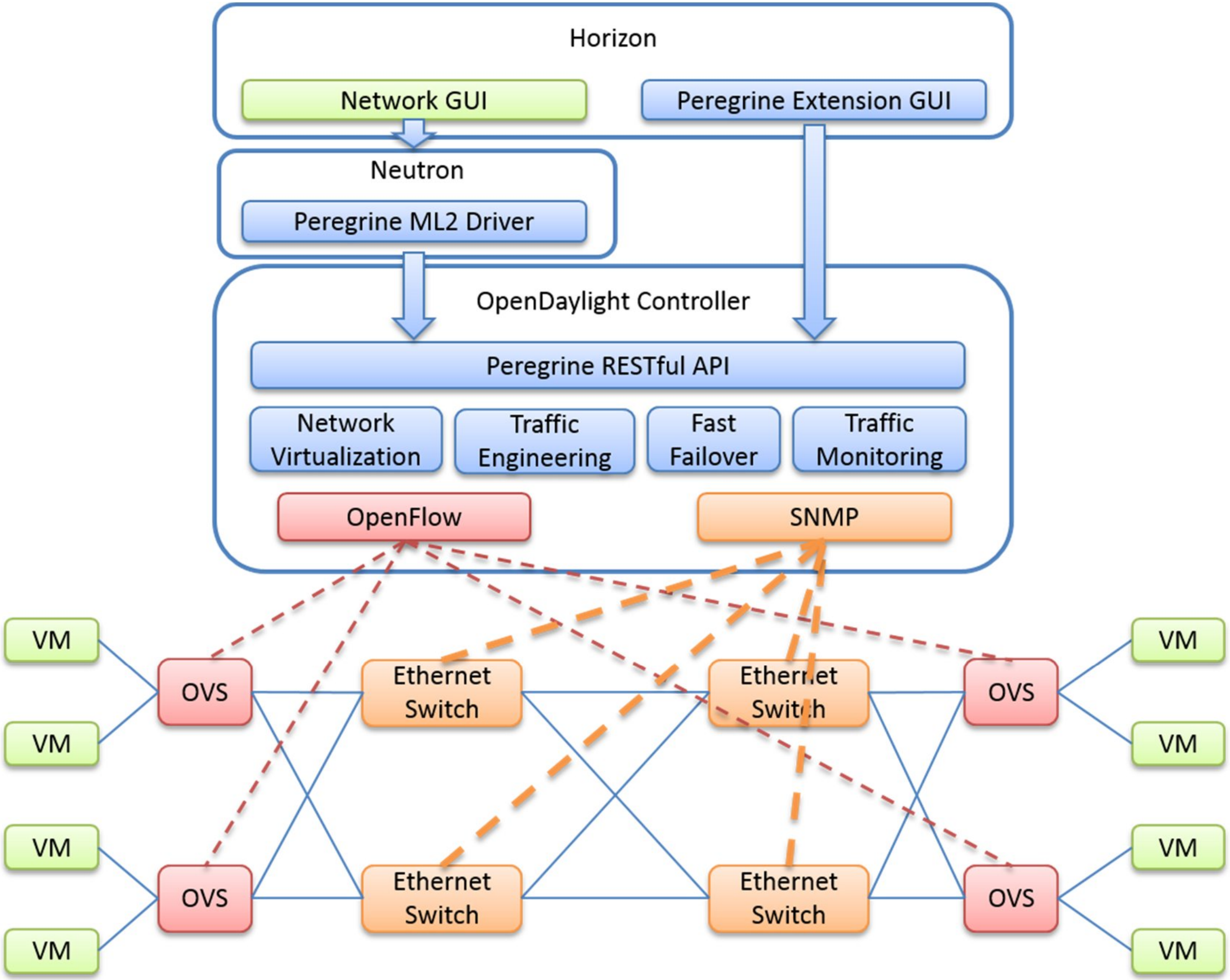
Commodity Ethernet Switch

No vendor lock-in and no need to spend money in expensive hardware

Virtual OpenFlow Switch (OVS)

Provide powerful edge intelligence

Peregrine Architecture



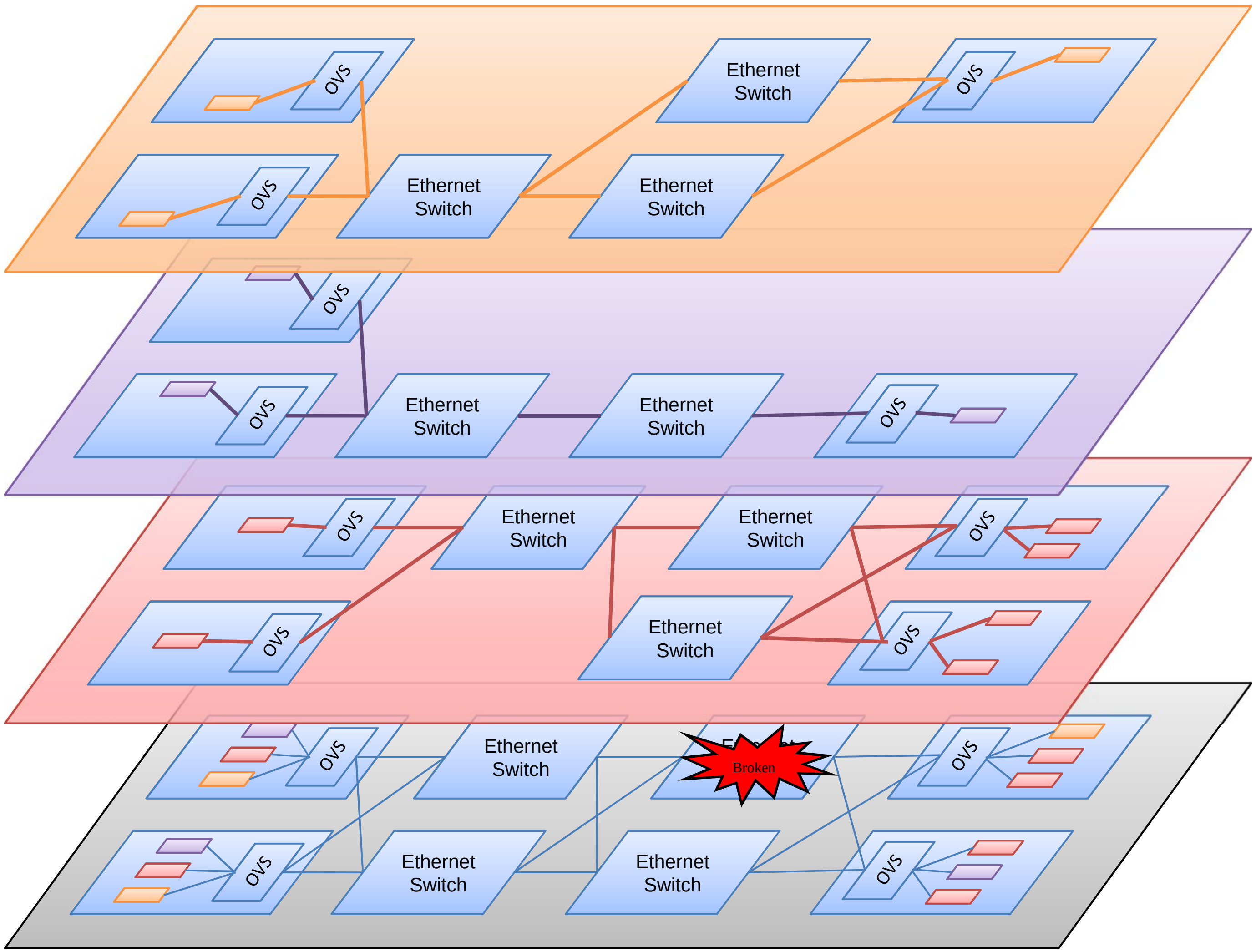
SDN Features

- Network Virtualization
- Traffic Engineering for Virtual Network
- Fast Failover for Virtual Network

🟥 Tenant A

🟪 Tenant B

🟧 Tenant C



Virtual Network

Virtualization Layer

Physical Network

Peregrine – Network Virtualization

Existing Solutions

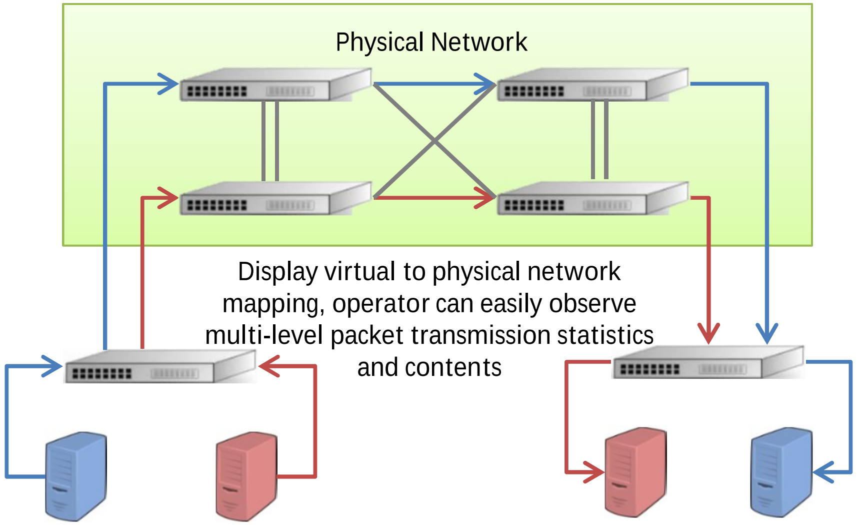
- ***Overlay Solution***
- +Flexible deployment (crossing L3 domains)
- +No hardware vendor lock-in
- +Support 16M virtual networks
- - No traffic load balancing or link failover
- -Incur extra computation and payload overhead

- ***Underlay Solution***
- +Use link aggregation for load balancing and link failover
- +Use VLAN to isolate virtual network - less overhead
- -Use hash-based link aggregation solution, packet transmission path is opaque
- -Fully integrated w/ and locked-in vendors' hardware devices
- -VLAN limitation - 4k virtual networks at most

🟩 + Pros
🟥 - Cons

The Next-Gen SDN

- ***Peregrine*** – Marrying the benefits of both worlds
- +Support OpenFlow and Ethernet device, no vendor lock-in
- +Use VLAN to isolate virtual network - less overhead
- +Support GLOBAL load balancing and FAST link failover
- +Extensive Monitoring GUI: virtual to physical network mapping, multi-level traffic flow display
- +VLAN and VxLAN integration to support 16M virtual networks
- +Flexible deployment (crossing L3 domains)



Display virtual to physical network mapping, operator can easily observe multi-level packet transmission statistics and contents

Peregrine – Traffic Engineering

Increase each VM network throughput in 1Gb network

0.12 Gbps (STP) -> 0.7 Gbps (ICL Peregrine SDN)

(the more the better)

Decrease traffic standard deviation

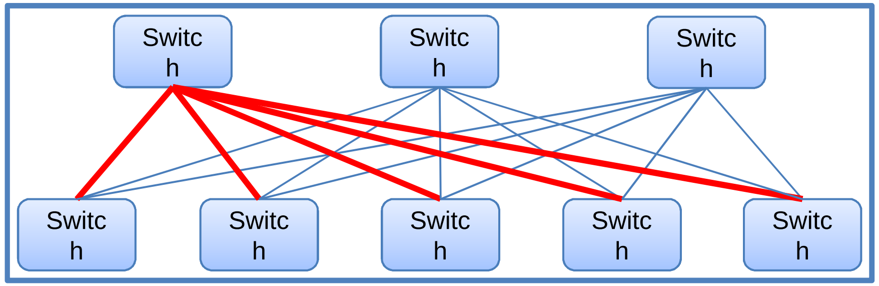
20% (STP) -> to 9.2% (ICL Peregrine SDN)

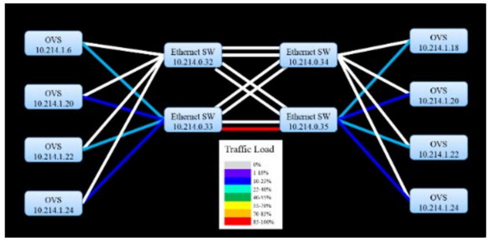
(the less the better)

▬ VLAN: 10

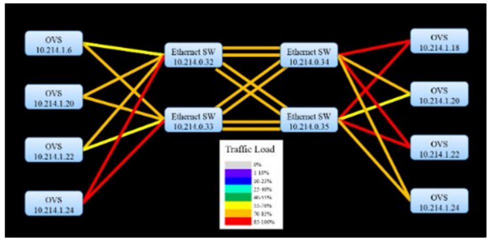
▬ VLAN: 20

▬ VLAN: 30

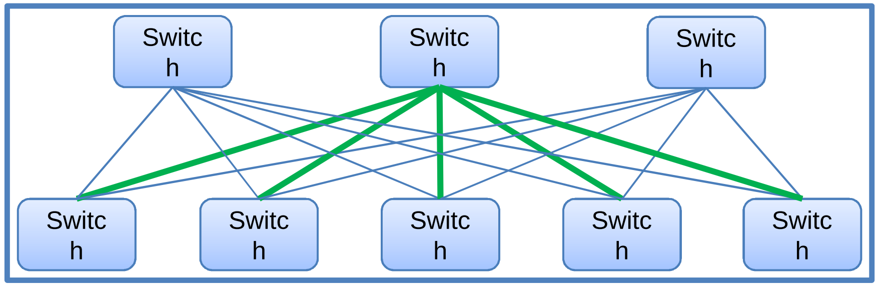


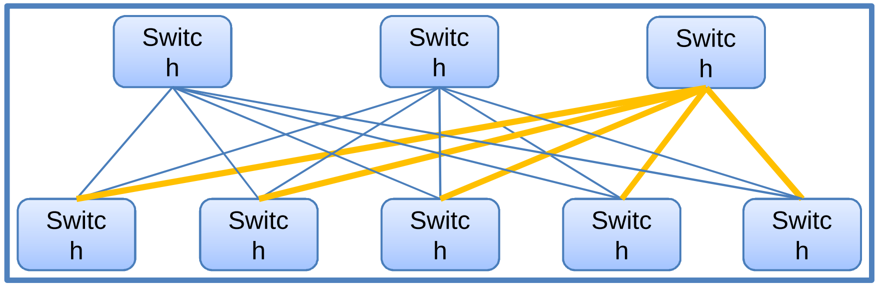


Traffic Load on STP



Traffic Load on Peregrine



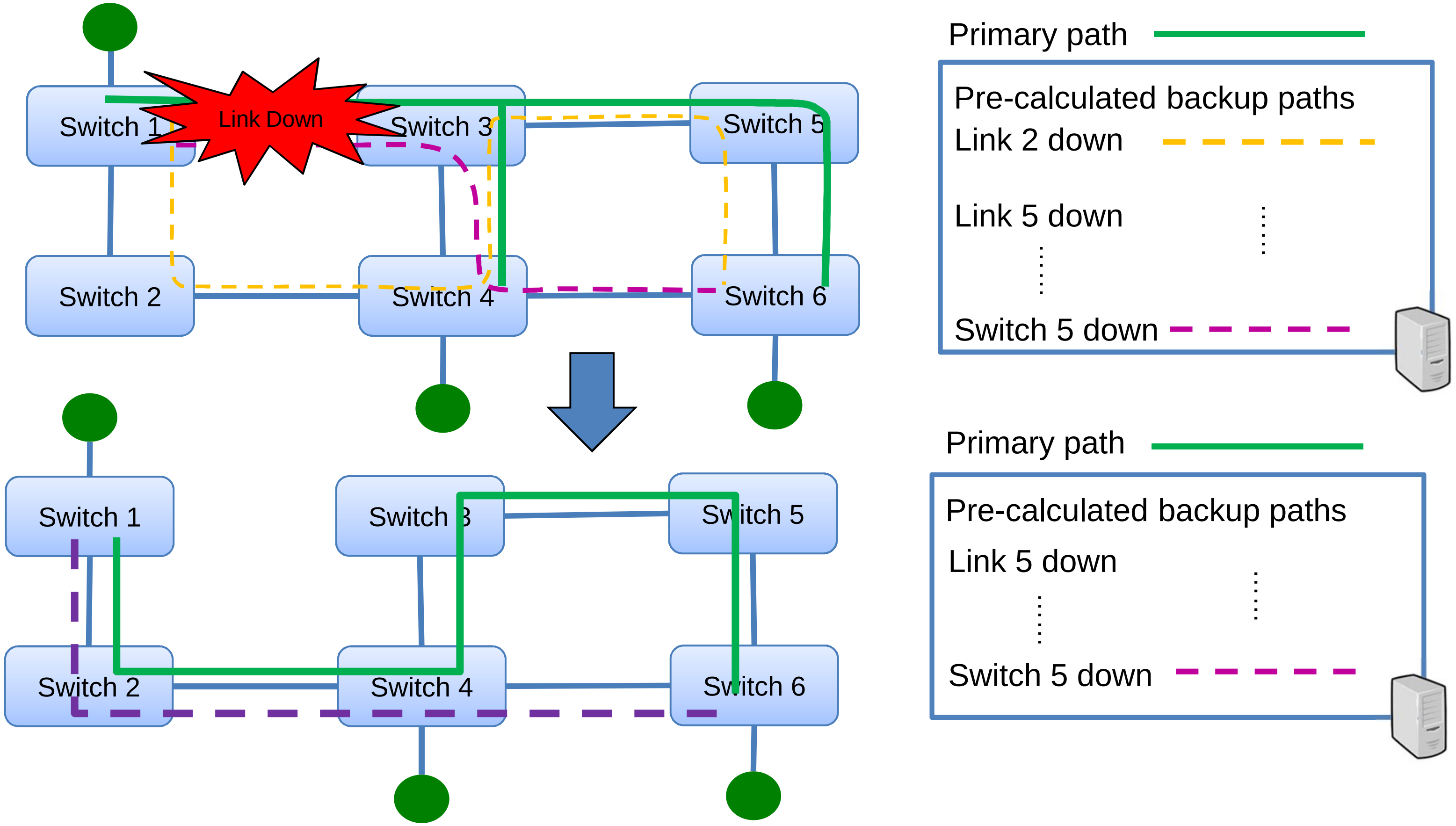


Peregrine – Fast Failover

Data traffic can be redirected from fail links to another route

30 secs (STP) -> 0.8 sec (ICL Peregrine SDN)

(the less the better)



Peregrine – Network Diagnosis GUI

Provides multi-resolution network traffic analysis and helps operator to find out the root cause of network problem

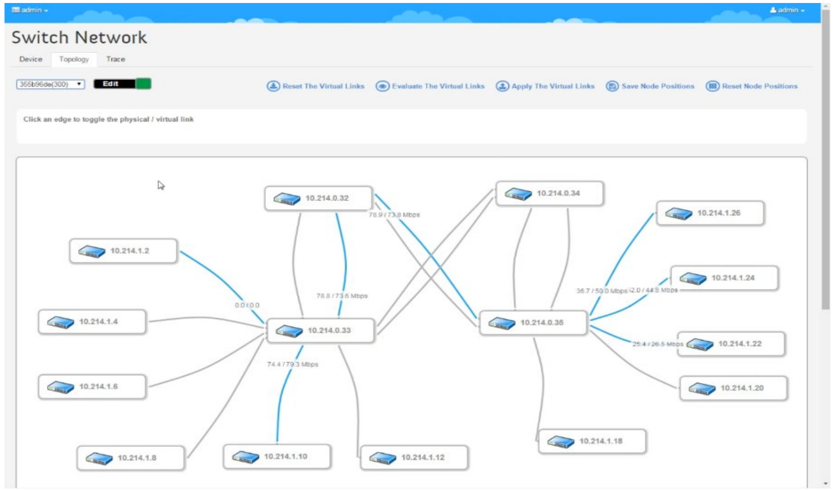
Physical & virtual topology

Physical & virtual traffic load

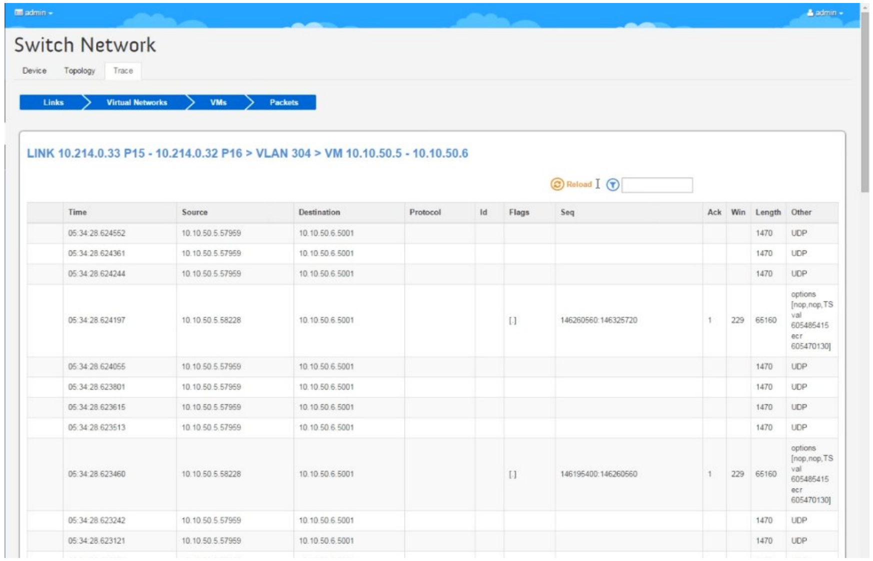
VM traffic analysis

User defined data path

OpenStack integrated

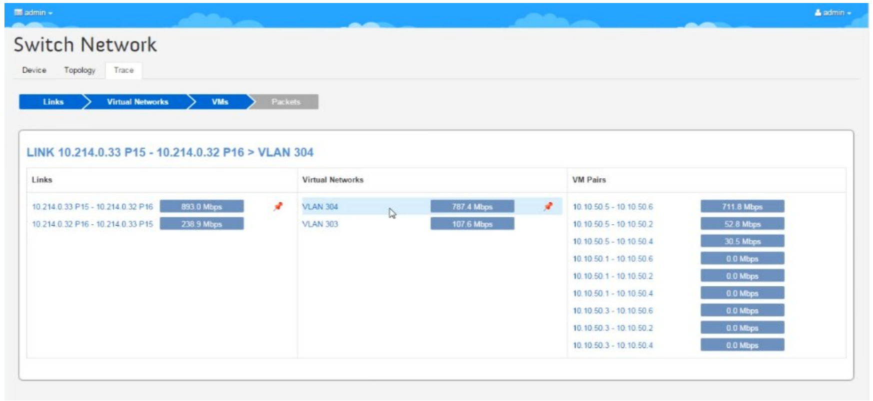


Click an edge to toggle the physical / virtual link



LINK 10.214.0.33 P15 - 10.214.0.32 P16 > VLAN 304 > VM 10.10.50.5 - 10.10.50.6

	Time	Source	Destination	Protocol	Id	Flags	Seq	Ack	Win	Length	Other
	05:34:28.624552	10.10.50.5.57959	10.10.50.6.5001							1470	UDP
	05:34:28.624361	10.10.50.5.57959	10.10.50.6.5001							1470	UDP
	05:34:28.624244	10.10.50.5.57959	10.10.50.6.5001							1470	UDP
	05:34:28.624197	10.10.50.5.58228	10.10.50.6.5001			[.]	146260560:146325720	1	229	65160	options [nop,nop,TS val 605485415 ecr 605470130]
	05:34:28.624055	10.10.50.5.57959	10.10.50.6.5001							1470	UDP
	05:34:28.623801	10.10.50.5.57959	10.10.50.6.5001							1470	UDP
	05:34:28.623615	10.10.50.5.57959	10.10.50.6.5001							1470	UDP
	05:34:28.623513	10.10.50.5.57959	10.10.50.6.5001							1470	UDP
	05:34:28.623460	10.10.50.5.58228	10.10.50.6.5001			[.]	146195400:146260560	1	229	65160	options [nop,nop,TS val 605485415 ecr 605470130]
	05:34:28.623242	10.10.50.5.57959	10.10.50.6.5001							1470	UDP
	05:34:28.623121	10.10.50.5.57959	10.10.50.6.5001							1470	UDP



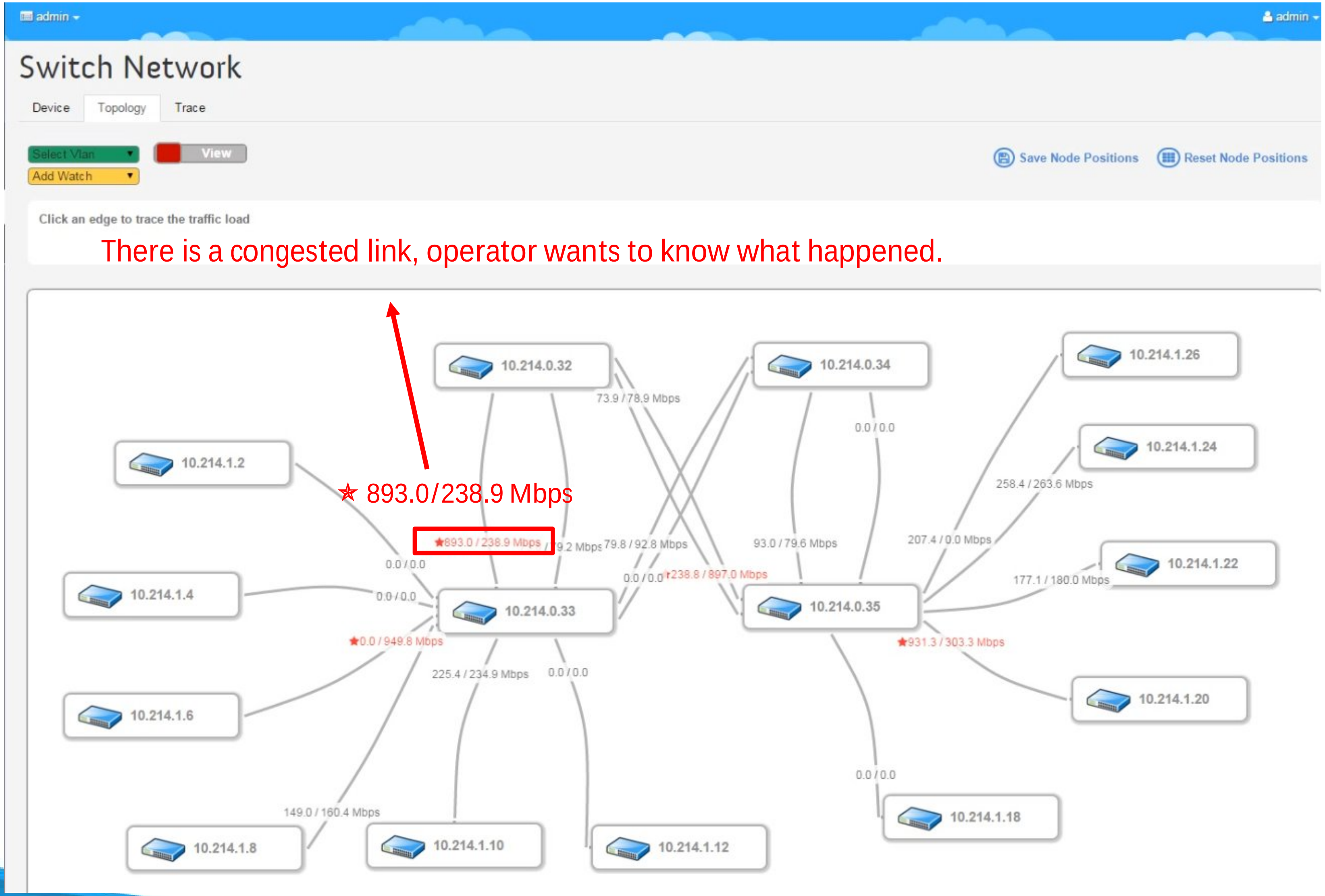
LINK 10.214.0.33 P15 - 10.214.0.32 P16 > VLAN 304

Links		Virtual Networks		VM Pairs	
10.214.0.33 P15 - 10.214.0.32 P16	893.0 Mbps	VLAN 304	787.4 Mbps	10.10.50.5 - 10.10.50.6	711.8 Mbps
10.214.0.32 P16 - 10.214.0.33 P15	238.9 Mbps	VLAN 303	107.6 Mbps	10.10.50.5 - 10.10.50.2	52.8 Mbps
				10.10.50.5 - 10.10.50.4	30.5 Mbps
				10.10.50.1 - 10.10.50.6	0.0 Mbps
				10.10.50.1 - 10.10.50.2	0.0 Mbps
				10.10.50.1 - 10.10.50.4	0.0 Mbps
				10.10.50.3 - 10.10.50.6	0.0 Mbps
				10.10.50.3 - 10.10.50.2	0.0 Mbps
				10.10.50.3 - 10.10.50.4	0.0 Mbps

ITRI
Industrial Technology
Research Institute

Use Case：Find Out The Elephant Flow (1/3)

Physical Traffic → Virtual Traffic → VM Pair Traffic → VM Packet



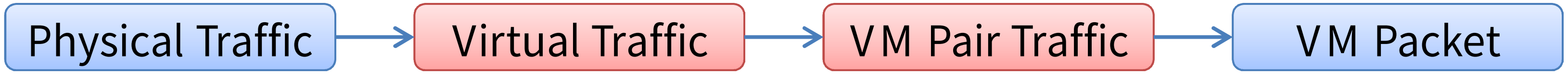
Click an edge to trace the traffic load

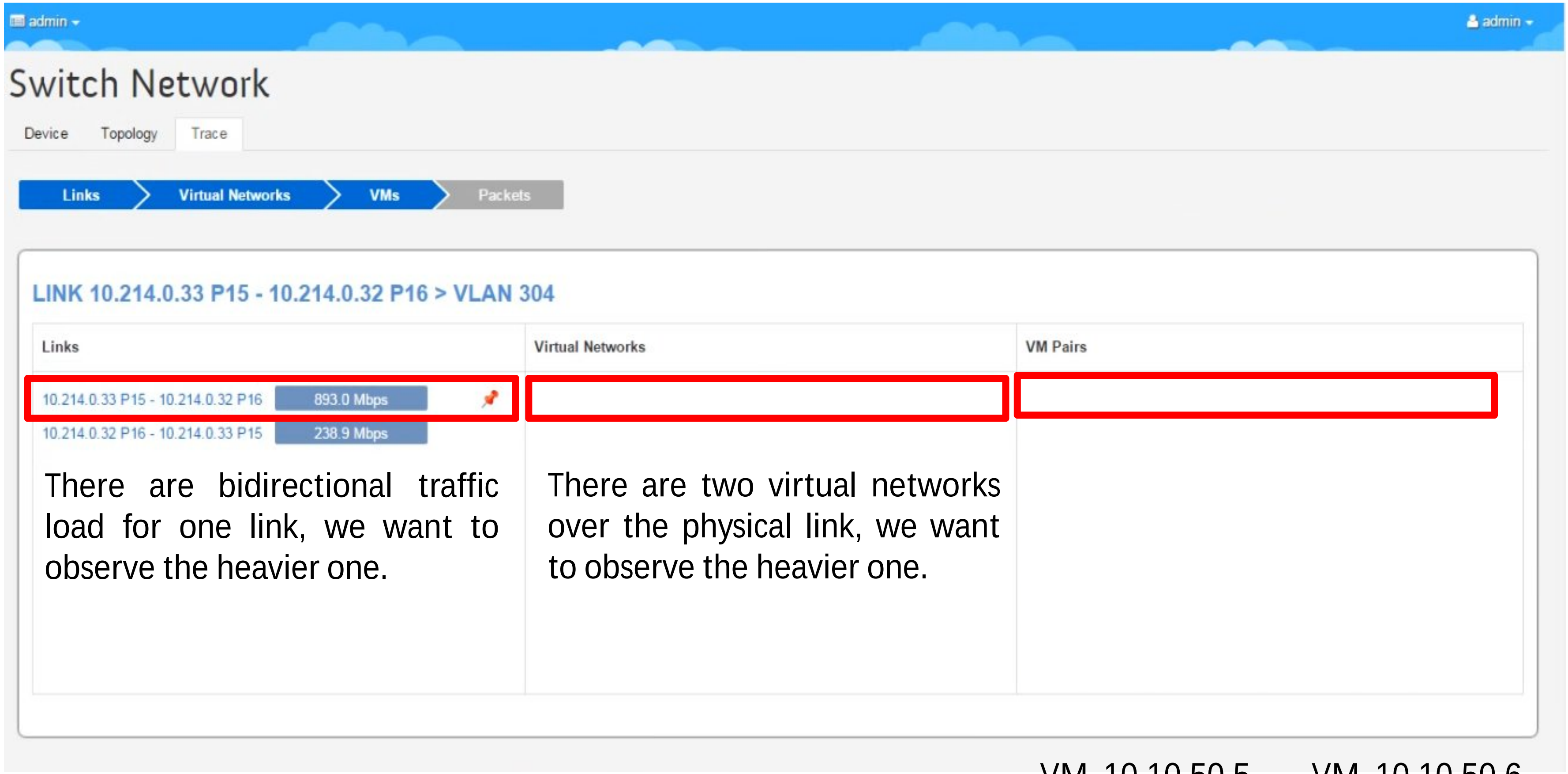
There is a congested link, operator wants to know what happened.

☆ 893.0/238.9 Mbps

Copyright 2016 ITRI 工業技術研究院

Use Case：find Out The Elephant Flow (2/3)





Switch Network

Device Topology Trace

Links > Virtual Networks > VMs > Packets

LINK 10.214.0.33 P15 - 10.214.0.32 P16 > VLAN 304

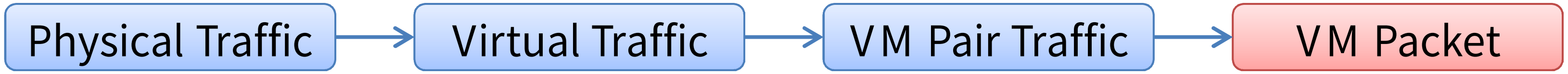
Links		Virtual Networks	VM Pairs
10.214.0.33 P15 - 10.214.0.32 P16	893.0 Mbps		
10.214.0.32 P16 - 10.214.0.33 P15	238.9 Mbps		

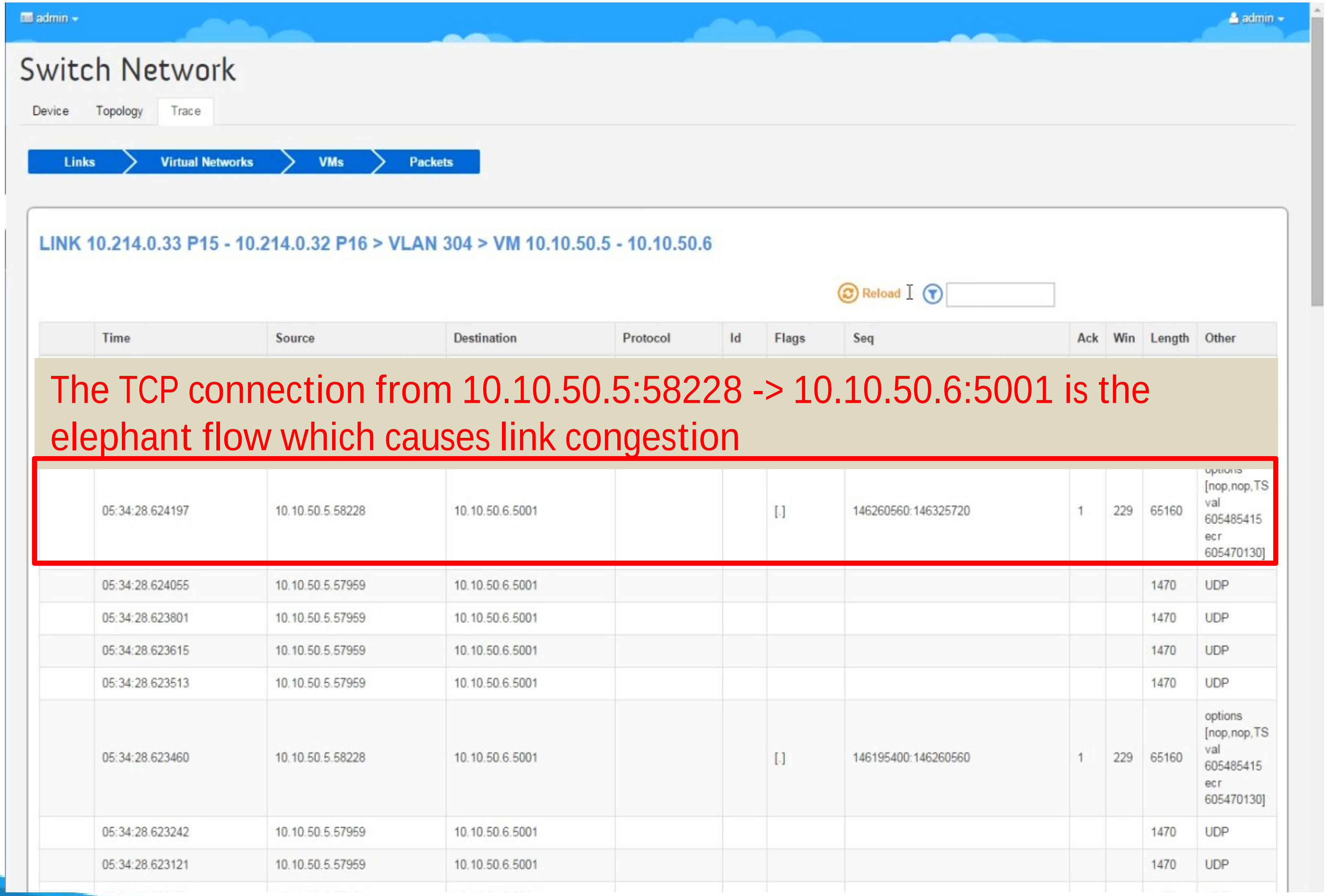
There are bidirectional traffic load for one link, we want to observe the heavier one.

There are two virtual networks over the physical link, we want to observe the heavier one.

VM 10.10.50.5 – VM 10.10.50.6 cost much more traffic load, we want to know what kind packets between those VMs.

Use Case：find Out The Elephant Flow (3/3)





Switch Network

LINK 10.214.0.33 P15 - 10.214.0.32 P16 > VLAN 304 > VM 10.10.50.5 - 10.10.50.6

The TCP connection from 10.10.50.5:58228 -> 10.10.50.6:5001 is the elephant flow which causes link congestion

	Time	Source	Destination	Protocol	Id	Flags	Seq	Ack	Win	Length	Other
	05:34:28.624197	10.10.50.5.58228	10.10.50.6.5001			[.]	146260560:146325720	1	229	65160	options [nop,nop,TS val 605485415 ecr 605470130]
	05:34:28.624055	10.10.50.5.57959	10.10.50.6.5001							1470	UDP
	05:34:28.623801	10.10.50.5.57959	10.10.50.6.5001							1470	UDP
	05:34:28.623615	10.10.50.5.57959	10.10.50.6.5001							1470	UDP
	05:34:28.623513	10.10.50.5.57959	10.10.50.6.5001							1470	UDP
	05:34:28.623460	10.10.50.5.58228	10.10.50.6.5001			[.]	146195400:146260560	1	229	65160	options [nop,nop,TS val 605485415 ecr 605470130]
	05:34:28.623242	10.10.50.5.57959	10.10.50.6.5001							1470	UDP
	05:34:28.623121	10.10.50.5.57959	10.10.50.6.5001							1470	UDP

Product Specification

VLAN-based Peregrine SDN Solution

Key Feature

- Network Virtualization
  - VLAN isolation
  - VLAN + VxLAN (2017 Q2)
- Traffic Engineering
  - Place the spanning tree associated with each VLAN on the physical network so as to balance their loads
- Fast Failover
  - Proactively compute for every VLAN an alternative spanning tree for every link failure in the main spanning tree
- Load Balancing
  - Neutron load balancing support
- VPN
  - Neutron multi-tenancy VPN support
- Firewall
  - Neutron firewall support

Reliability

- Failover in 1 second

Scalability

- 4K virtual network support
- 16M virtual network support (2017 Q2)

Management

- Physical and virtual topology monitoring
- Physical and virtual traffic load monitoring
- VM traffic load monitoring
- VM packet analysis
- Network diagnosis GUI

High Availability

- OpenDaylight Cluster
- MySQL Galera

API

- OpenStack Neutron API
- Extension RESTful API

THANKS FOR YOUR ATTENTION

ITRI
Industrial Technology
Research Institute