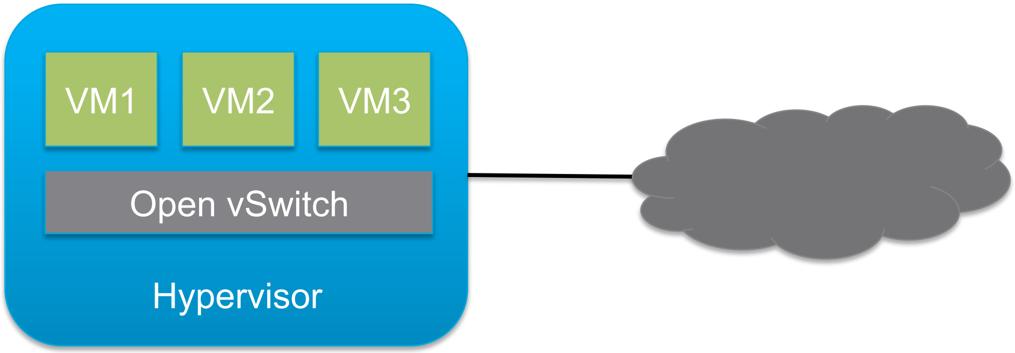
Open vSwitch and the Intelligent Edge

Justin Pettit

OpenStack 2014 Atlanta

Hypervisor as Edge



An Intelligent Edge

- We view the hypervisor as the edge of the network
- An intelligent edge is in a unique position (the “Goldilocks Zone”)
  - Greater context than in-network devices
    - Without tags, network must rely on fields that are easily spoofed
    - Tags provide limited amount of context
  - Reduced risk of attack than an agent running in the guest
    - Policies enforced in the hypervisor – outside of the guest
  - Enforce policies earlier
    - Clouds typically have over-subscribed links and untrusted sources
- Different parts of the system can coordinate with each other
- Can affect many things
  - Networking
  - Security

Network Control and Visibility

- In an ideal location
- Able to infer state by observing, or probe state with introspection
- Mapping of logical to physical before going into the fabric
- Can modify behavior
  - Enforce policy at tunnel ingress and egress
  - Modify bits in the inner or outer packet
  - TCP Pacing
  - TCP De-synchronization
  - Flowlets

Inferring State

- Sees every packet and knows local source
  - Learn MAC and IP on first use
  - IGMP and DHCP snooping
  - Which pairs are communicating
  - Flow characteristics

Guest Introspection

- An agent runs in the VM that communicates with a daemon in the hypervisor
- Types of data retrieved
  - Users
  - Identity for both inbound and outbound network connections
  - Identity (user and version/hash) of processes
  - Data transfer rates
  - Socket queue depth
  - System characteristics

Applications for Greater State

- QoS
- Load-balancing
- Selecting traffic to be sent to middlebox (NFV)
- Better firewalls
- Elephant flow detection and handling

Security

Implementing a Firewall

- Currently, two ways to implement a firewall in OVS
  - Match on TCP flags (Enforce policy on SYN, allow ACK|RST)
    - Pro: Fast
    - Con: Allows non-established flow through with ACK or RST set, only TCP
  - Use “learn” action to setup new flow in reverse direction
    - Pro: More “correct”
    - Con: Forces every new flow to OVS userspace, reducing flow setup by orders of magnitude
  - Neither approach supports “related” flows or TCP window enforcement

Connection Tracking

- We are adding the ability to use the conntrack module from Linux
  - Stateful tracking of flows
  - Supports ALGs to punch holes for related “data” channels
    - FTP
    - TFTP
    - SIP
- Implement a distributed firewall with enforcement at the edge
  - Better performance
  - Better visibility
- Introduce new OpenFlow extensions:
  - Action to send to conntrack
  - Match fields on state of connection
- Have prototype working. Expect to ship as part of OVS by end of year

Guest Introspection + Connection Tracking

- Possible to implement an advanced firewall
  - Know precisely what user is generating traffic
  - Know precisely what application and version is generating traffic

Elephant Flows

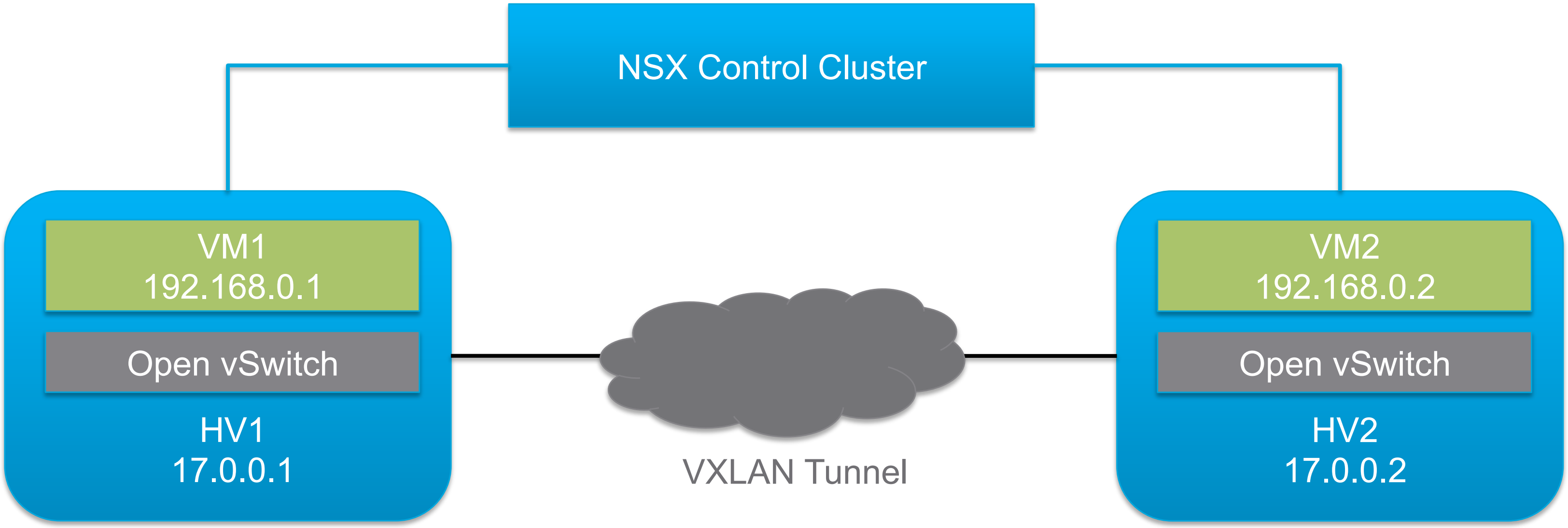
Elephants versus Mice

- Majority of flow are short-lived (mice), but majority of packets are long-lived (elephants)
- Mice tend to be bursty and latency-sensitive
- Elephants tend to transfer large amount of data and less concerned about latency
- Elephants can fill up network buffers, which introduce latency for mice
- At the edge, we are able to affect the underlay based on the overlay

Detection and Action

- Multiple mechanisms for detection:
  - Rate and time
  - Large segments (TCP only)
  - Guest introspection
- Multiple mechanisms for action:
  - Put mice and elephants into different queues
  - Route elephants differently from mice
  - Send elephants along a separate physical network
  - Intelligent underlay

NSX Deployment

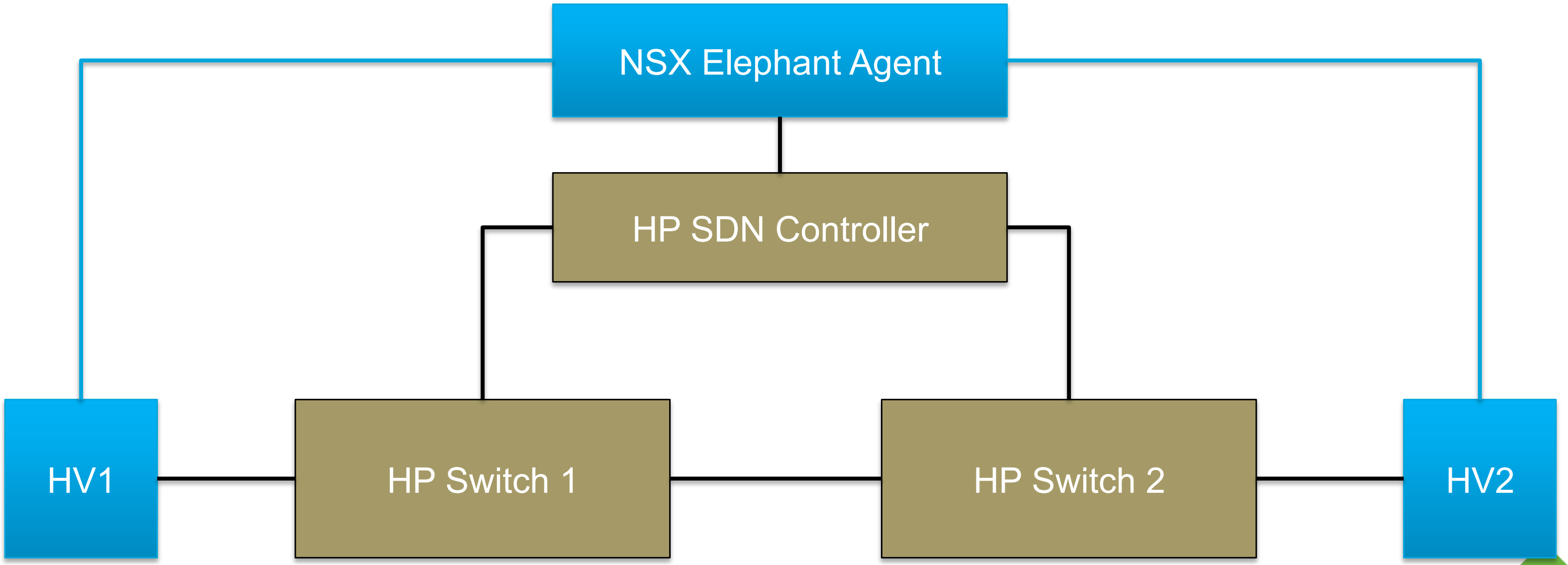


Handling Elephants in NSX

- Open vSwitch is at an optimal location at the edge
  - Has flow-level view of all the hypervisor’s traffic
  - Knows mapping between logical and physical addresses
- Detection and action occur separately, so can evolve independently
- Supported detection mechanisms:
  - Rate and time
  - Large segments
- Supported actions:
  - Mark DSCP bits in (outer) IP header
  - Add elephant flows to OVSDB column for underlay agent

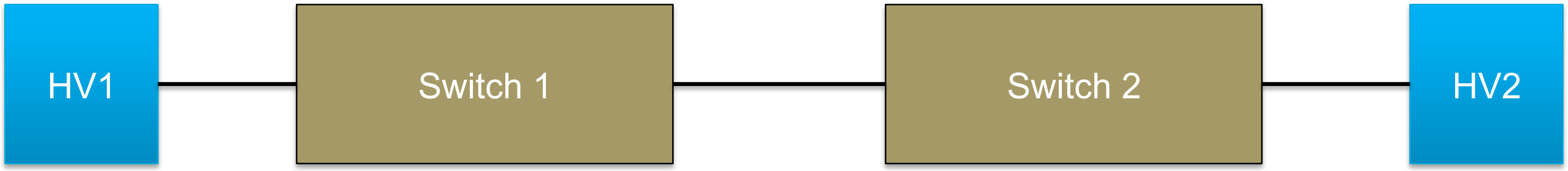
Elephant Flows with SDN Controller

- OVS identifies elephants as the appear on the wire through OVSDB
- An agent monitors OVSDB and makes appropriate API calls to the SDN controller
- Shown as a VMware-HP Technology Preview



Elephant Flows with DSCP Marking

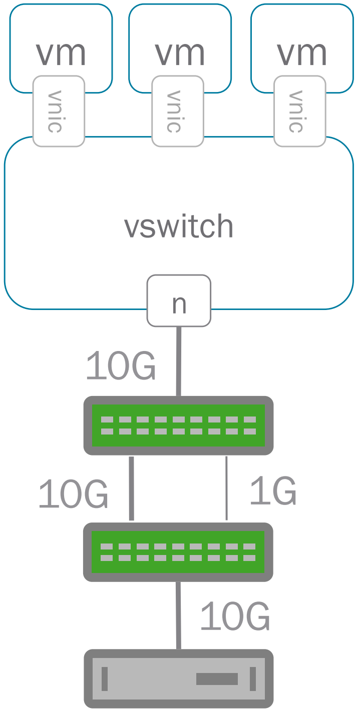
- Signaling of elephants occur at the hypervisor by marking the (outer) IP header
- Switches configured to handle elephant-marked packets appropriately
- Working on an Internet Draft for recommended DSCP values



Testing Results with Cumulus Networks

- Used a modified OVS that detects elephant flows by counting the number of bytes each flow generates. When the user-configurable threshold is crossed, elephants are marked with a particular DSCP value.
- The Cumulus switches place elephant marked flows into an alternate queue

Test Topology

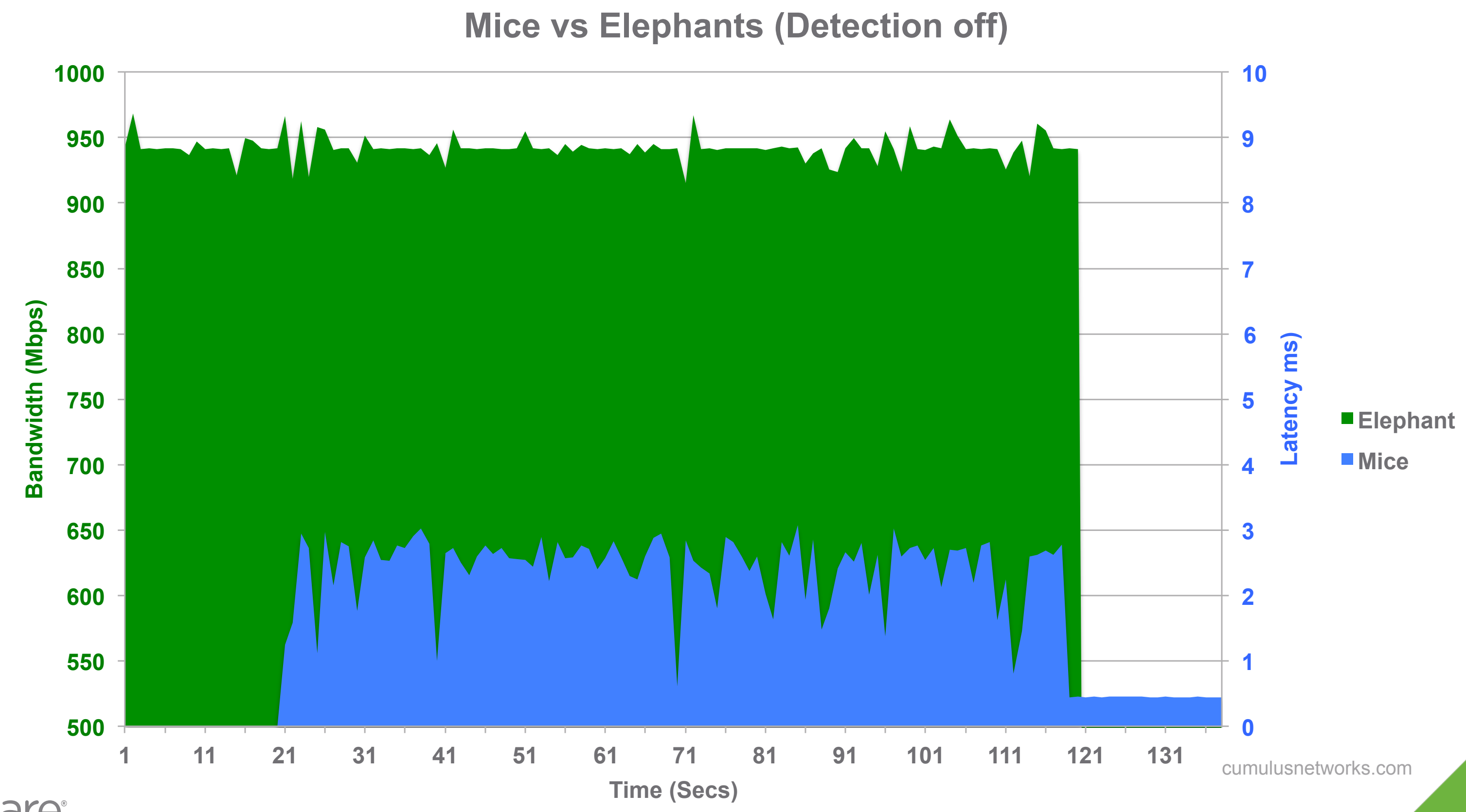


- Sources
  - VMs connected via vSwitch
    - 10G connection to network
- Network Paths
  - 1G “normal” link
    - easy to congest with VM traffic sources
  - 10G “alternative” link
- Sink
  - bare metal server
    - 10G connection from network

Traffic Generation and Result Measurement

- Generators
  - elephants – nuttcp
    - fixed time transfers, 4M window
  - mice – small (10ms) interval pings
    - mimics tcp-acks, lock release, small db transations
- Results
  - elephants
    - realized bandwidth, drops
  - mice
    - mean-time-to-completion, drops

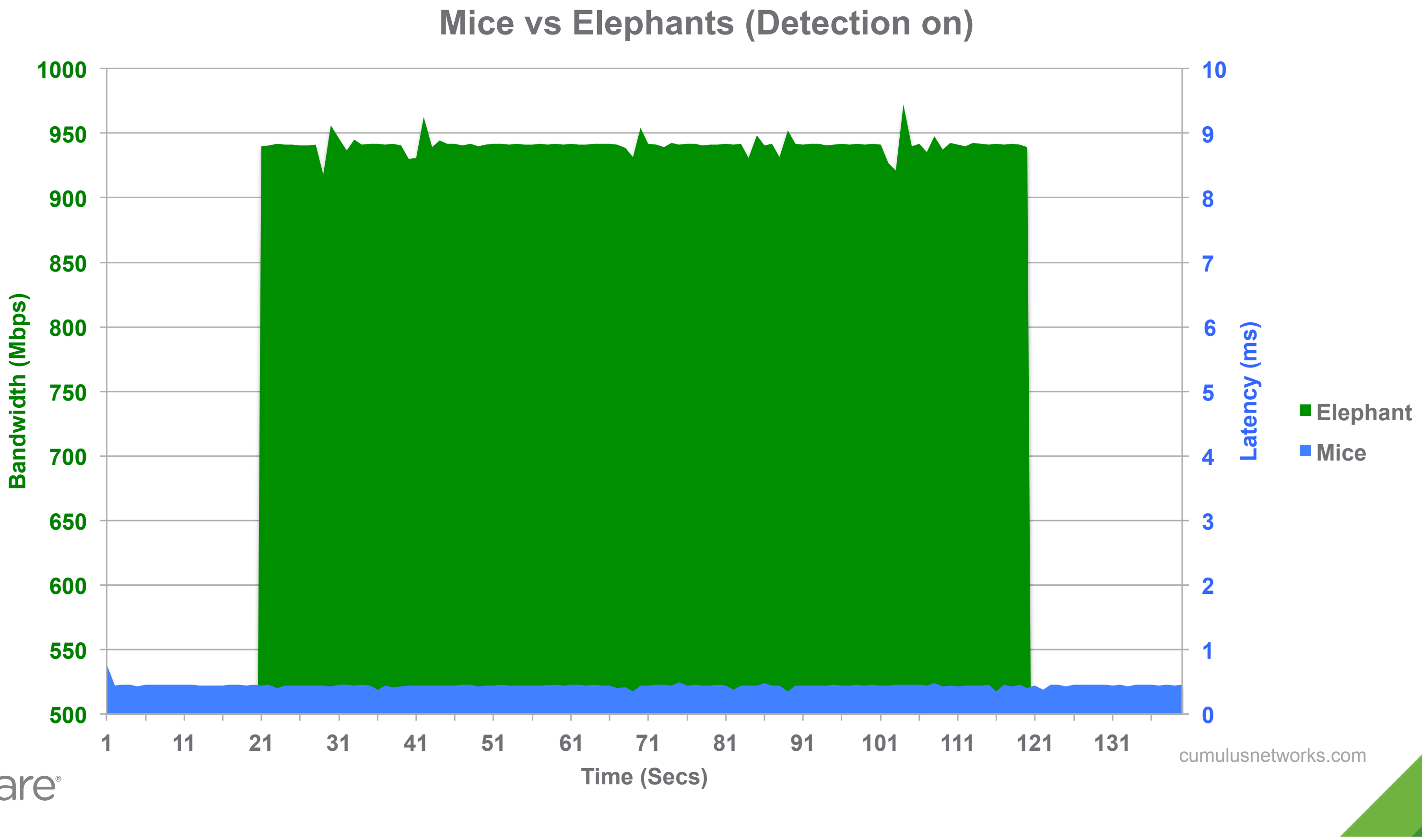
Results – flow statistic detection & alternate queue reaction



cumulusnetworks.com

vmware®

Results – flow statistic detection & alternate queue reaction



Results – flow statistic detection & alternate queue reaction

test case (120 sec period)	elephant		mouse	
	Mbps	drops	Latency (ms)	drops
elephant only	941	63	N/A	N/A
mouse only	N/A	N/A	0.444	0
mouse vs elephant no detection	941	61	3.055	0
mouse vs elephant w/detection	937	1223	0.401	0

Open vSwitch Elephant POC Architecture

- Implemented in kernel
- Supports both threshold-based detection and TSO packet size
- Just proof of concept to try out different detection mechanisms and actions
- Proof of concept code will be available on Github

Elephant Flow References

- Network Traffic Characteristics of Data Centers in the Wild
  - http://pages.cs.wisc.edu/~akella/papers/dc-meas-imc10.pdf
- Of Mice and Elephants
  - http://networkheresy.com/2013/11/01/of-mice-and-elephants/
- Elephant Flow Mitigation via Virtual-Physical Communication
  - http://blogs.vmware.com/networkvirtualization/2014/02/elephant-flow-mitigation.html

Learn more about VMware + OpenStack at the following sessions:

Monday

VMware Demo
1:00-1:15 pm, Demo Theater

Enterprise Grade Scheduling
4:40-5:20 pm, B206

Bridging The Gap: OpenStack For VMware Administrators
5:30-6:10 pm, B206

Software Defined Networking Performance And Architecture Evaluation
5:30-6:10 pm, B103 *Presented by Symantec & Mirantis*

Tuesday

Scaling Neutron For Large Deployments
4:40-5:20 pm, B101 *Presented by eBay & PayPal*

Open vSwitch And The Intelligent Edge
5:30-6:10 pm, B206

Wednesday

VMware + OpenStack: Accelerating OpenStack In The Enterprise
1:50-2:30 pm, B313

Deep-dive Demo for OpenStack On VMware
2:40-3:20 pm, B313

OpenStack Distribution Support For vSphere + NSX
3:30-4:10 pm, B313

Congress: A System For Declaring, Auditing, and Enforcing Policy In Heterogeneous Cloud Environments
4:30-5:10 pm, B313

VSAN and OpenStack
5:20-6:00 pm, B313

Thursday

Recap: Nova-network Or Neutron For OpenStack Networking?
9:50-10:30 am, B309

Leveraging VMware Technology To Build An Enterprise Grade OpenStack Cloud - It's Not Always About KVM!
2:20-3:00 pm, B101 *Presented by iLand*

Hands-on-Labs

OpenStack on VMware vSphere and NSX
Wed, May 14, 3:30-5:30 pm, B313

OpenStack Networking
Wed, May 14, 4:30-6:00 pm, B314

□ Session by VMware

▨ Session by VMware Customers / Partners

vmware®

The Enterprise-Grade Foundation For Your OpenStack Cloud