LEARN. NETWORK. EXPERIENCE OPEN SOURCE.

June 11-14, 2013
Boston, MA
Network Virtualization and Software-defined Networking
Chris Wright and Thomas Graf
Red Hat
June 14, 2013
Agenda

- Problem Statement
- Definitions
- Solutions
She can't take much more of this, captain!
Challenges
Managing forwarding elements

- vendor specific, often manual, error prone
Changes in traffic patterns

- increase bisectional bw, don't increase cost
- north-south, east-west
Dynamic workloads

- Respond in real time
- Virtualization, cloud, BYOD, mobility
Visibility

- Debugging complex networks is hard, let's go shopping
Definitions
Network

- Collection of endpoints and forwarding elements
- Job is to move packets between hosts
- Source hosts identify destination
- Forwarding elements direct traffic at each intersection
What is SDN?

- Separation of control plane from data plane
- Standardized programmatic control of traffic flows
- Global view of network
Traditional forwarding device

Management interface
CLI, Console, SNMP, ...

Control Plane
Forwarding Decision (Learning, RIB Lookup), Routing Protocols (OSPF, BGP, ...)

Data / Forwarding Plane
Fabric, Flow Table, Forwarding Engine
SDN forwarding logic
Network Virtualization

- Decouple logical topologies from physical topology
  - Build complete virtual network topologies
    - Provide layer 2-7 network services
  - Isolated tenant networks
- Network is an abstraction
  - API to dynamically manage network abstraction
Naive VLAN mapping
VLAN trunking

Nova Compute Node
VM1
vSwitch
VM1

Nova Compute Node
VM2
vSwitch
VM2

Nova Compute Node
VM3
vSwitch
VM3

Switch

Switch

Switch

#redhat #rhsummit
Network overlay
Network abstraction

Network Service API

Logical

Switch

Physical

Switch

Switch

Switch

Switch
Under the hood
OpenFlow

An Open Standard behind SDN

1. Match on bits in packet header L2-L4 plus meta data
2. Execute actions
   - Forward to port
   - Drop
   - Send to controller
   - Mangle packet

OpenFlow enables networks to evolve, by giving a remote controller the power to modify the behavior of network devices, through a well-defined "forwarding instruction set". The growing OpenFlow ecosystem now includes routers, switches, virtual switches, and access points from a range of vendors.
Dynamically update flow tables in a universal language.

In the Software Defined Networking architecture, the control and data planes are decoupled, network intelligence and state are logically centralized, and the underlying network infrastructure is abstracted from the applications.

Software-Defined Networking: The New Norm for Networks
ONF White Paper
April 13, 2012
Fine Grained Flow Table Control

Extensive flow matching capabilities
- Meta – Tunnel ID, In Port, QoS priority, skb mark
- Layer 2 – MAC address, VLAN ID, Ethernet type
- Layer 3 – IPv4/IPv6 fields, ARP
- Layer 4 – TCP/UDP, ICMP, ND

Chain of actions
- Output to port(s) (single, range, flood, mirror)
- Discard, Resubmit to other table
- Packet Mangling (Push/Pop VLAN header, TOS, ...)
- Send to controller, Learn
- Set tunnel ID
OpenFlow Capable Devices

Software Switches
- Open vSwitch, Cisco Nexus 1000V, VMware vSphere, NEC Hyper-V, ...

Hardware Switches
- Brocade, Cisco, HP, IBM, Juniper Networks, NEC, ...

SwitchLight
- Open source firmware and agent leveraging Ethernet switching ASICs to support OpenFlow
Open vSwitch

Open vSwitch is an open, virtual multi layer switch for hypervisors providing network connectivity to virtual machines.
Open vSwitch Project

- Multilayer virtual switch for VMs
- Portable / Multi Platform
- Developed by VMware (Nicira) & Community
- Apache License (User Space), GPL (Kernel)
- OpenFlow 1.1+ (+ extensions)
Controller programs flow table in the **slow path** that feeds the flow table in the **fast path** upon request.
Network Segregation

**VLAN isolation** enforces **VLAN membership** of a VM without the knowledge of the guest itself.

Caveat: MAX(VLAN_ID) limited
Tunneling (Overlay Networks)

Tunneling provides **isolation** and reduces **dependencies** on the physical network.
Visibility

Supports industry standard technology to monitor the use of a network.

- **sFlow**
- NetFlow
- Port Mirroring
  - SPAN
  - RSPAN
  - ERSPAN
Quality of Service

- Uses existing Traffic Control Layer
  - Policier (Ingress rate limiter)
  - HTB, HFSC (Egress traffic classes)
- Controller (Open Flow) can select Traffic Class
Up Next

Multithreading
- Enables parallel processing in slow path

MegaFlows
- Support for wildcard flows in the datapath
- Any non-present flow component is considered a wildcard
- **Reduction in # of flows in datapath by ~ 40%**

Zerocopy Upcall
- Avoid expensive memcpy() when copying packet to user space
Tunneling is not Tunneling

**kernel.org**
- No tunnel ports
- veth instead of patch ports
- Flow based tunneling
- VXLAN, GRE

**openvswitch.org**
- Port based Tunneling
- VXLAN, GRE, LISP

Regular merge
What is OpenDaylight?

OpenDaylight is an Open Source Software project under the Linux Foundation with the goal of furthering the adoption and innovation of Software Defined Networking (SDN) through the creation of a common industry supported framework.
Who is OpenDaylight?

Members as of April 8, 2013 and growing
OpenDaylight architecture
Putting it all together

OpenStack Networking

OpenDaylight Controller
OpenFlow

OVS
OpenStack Compute KVM

OVS
OpenStack Compute KVM

OVS
OpenStack Compute KVM
Questions?

- OpenDaylight
  - http://www.opendaylight.org/

- Open vSwitch
  - http://www.openvswitch.org/

- OpenFlow
  - http://www.openflow.org/

- Red Hat OpenStack
  - http://www.redhat.com/openstack/

- RDO
  - http://openstack.redhat.com/

- OpenStack
  - http://www.openstack.org/
Thanks for participating in the session

Network Virtualization & Software-defined Networking

http://www.keysurvey.com/f/521822/17d2/

• Access and complete a short, less than 2 minute survey
  Be entered in the Nexus7 32GB Tablet giveaway