Running OpenStack over a VXLAN Fabric

Andre Pech
Arista Networks
Overview

• VXLAN Refresher
• Why VXLAN?
• Network Design Requirements
• Key Decisions Points
• OpenStack over VXLAN designs
• Thoughts on future work
VXLAN Refresher

• Standardized overlay technology for encapsulating layer 2 traffic on top of an IP fabric
Learning and Flooding in VXLAN

• MAC Learning
  – Learn based on traffic received over the tunnel
  – And/or use a protocol to distribute MAC tables

• Handling BUM Traffic
  – BUM = Broadcast, Unknown Unicast, and Multicast traffic
  – Common options for BUM traffic distribution:
    • IP Multicast
    • Head-end replication / replication node
Why VXLAN?

• Addresses 4K VLAN limitation, enabling up to 16M tenant networks
• Solves mac address scaling issues at the core of the network
• Allows for better scalability and failover with an L3 ECMP fabric
• VXLAN support is only required at endpoints, allowing greater vendor flexibility in the network
• Networking ASIC support
Real World Requirements to Deploy OpenStack over VXLAN

• No IP Multicast!
  – IP multicast is an efficient, protocol based mechanism for BUM traffic distribution
  – But no one wants to run multicast in their network

• Hardware VXLAN gateways
  – Get North-South traffic into / out of your cloud
  – Bridge physical infrastructure (storage, non-virtualized servers, etc) into virtual networks
  – The performance and density of software VXLAN gateways is not sufficient
Some Key Design Decisions

- Software vs Hardware VTEPs
- Replication node vs fully distributed head-end replication
- External SDN Controller vs Standalone Neutron
Software vs Hardware VTEPs

• Flexibility of Software vs Performance of Hardware
  – Software VTEPs are limited only by RAM and CPU cycles, but there’s an overhead cost of 10-30% per compute node
  – Hardware VTEPs have great density and performance, but are limited to the size of hardware tables

• Network management in a VXLAN environment
Replication Node vs Fully Distributed Head End Replication

• Replication nodes can be purpose-built
  – Flows can be spread across multiple replication nodes
  – But they to be managed and have an HA story

• Head-end replication at each VTEP requires no HA strategy
  – But burdens each VTEP with the cost of replication
External SDN Controller vs Standalone Neutron

• Hard tradeoff to quantify
• Generally comes down to functionality vs cost
OpenStack over VXLAN

• Three designs that fit the real world production requirements:
  – External SDN controller with a mix of Software and Hardware VTEPs
  – Standalone Neutron with all Hardware VTEPs
  – Standalone Neutron with a mix of Software and Hardware VTEPs
External SDN Controller, Software and Hardware VTEPs

Neutron Controller Plugin

SDN Controller (VMware NSX, PLUMgrid,...)

VTEP

VNI 5000

TOR 1

TOR 2

TOR 3

TOR 4

L3

L2

VM

VM

VM

VM

Compute A

Compute B

Compute C

Physical Infra
External SDN Controller, Software and Hardware VTEPs

• The SDN Controller (for example VMware NSX or PLUMgrid)
  – Manages virtual VTEPs and the VMs behind them
  – Integrates with the hardware VTEPs to configure gateway functionality for end-to-end provisioning driven by Neutron
  – Exchanges VXLAN MAC address table information between the physical and virtual VTEPs for a multicast-less VXLAN
ML2

• First, a quick plug for ML2
• ML2 is a new Neutron plugin in Havana which provides:
  – Separation between the state of tenant networks and how that state is then realized across the network
  – Flexibility in how the virtual and physical network are managed
  – Multi-vendor support via multiple “Mechanism Drivers” managing pieces of the network in parallel
• Talk on ML2 by Bob Kukura and Kyle Mestery on Friday at 11am
Standalone Neutron, All Hardware VTEPs
Standalone Neutron, All Hardware VTEPs

• Take advantage of hardware capabilities, reduce CPU utilization of each compute node

• Limited to 4K tenant networks (still limited by the VLAN space)
  – Though some work and ML2 multi-segment support, you could do rack-specific VLAN allocation and get beyond the 4K tenant network limit
Standalone Neutron, Software and Hardware VTEPs
Thoughts on Future Work

• Standalone Neutron with Software and Hardware VTEPs is hard to achieve today
  – Requires hook to share VXLAN connectivity info between the virtual and physical infrastructure
  – L2 population mechanism driver in ML2 is a step in the right direction

• Need a general model of VXLAN gateway nodes in Neutron
  – Dynamically attach/detach physical infrastructure into tenant networks
Questions?