Distributed Network Function Virtualization

Fred Oliveira, Fellow at Verizon
Sarath Kumar, Software Engineer at Big Switch Networks
Rimma Iontel, Senior Architect at Red Hat
Outline

- What is Distributed NFV?
- Why do we need Distributed NFV?
  - Verizon Use Case
- How do we implement Distributed NFV?
  - Architecture
  - Pitfalls
- Verizon + BigSwitch + Red Hat joint solution
  - Lab setup
  - Findings
- Wrap Up
- Q & A
Distributed NFV Architecture
Component Placement

● Distributed deployment of Network Functions at multiple sites with some level of remote control over those deployment models, traffic management for OpenStack and VNFs
  ○ Core Data Center
    ■ Deployment Tools
    ■ Network Controllers
    ■ Cloud Controllers
    ■ Orchestration
    ■ Monitoring, Troubleshooting and Analytics
    ■ Centralized Applications
  ○ Remote Sites
    ■ Compute Nodes running Edge Applications
Areas of Application

- Thick CPE (Customer Premise Equipment)
  - Enterprise
    - On-premise:
      - VNFs
        - Ex: FW, LB, WAN Optimization, NAT
      - Limited storage
    - In central DC:
      - Policy management and enforcement
      - Subscriber management
      - IPSec termination
      - Additional VNFs + SFC
  - Residential
    - On-premise:
      - VNFs
        - Ex: FW, NAT
      - Limited storage
    - In central DC:
      - Additional VNFs

- Remote POP
  - Web Cache
  - Video Streamers

- Mobile Edge Computing
Verizon Use Case - Distributed Network Services

- Support for new NFV services requires large number of small deployments
  - Low latency for highly interactive applications (VR, AR)
  - High bandwidth video and graphics distribution
  - Edge-Datacenter support with 4-16 servers at each hundreds of locations
  - Potentially scale to a single (micro) server (CPE) at 10s of thousands of retail locations
- Improve customer experience by providing on-demand software services
- Reduce cost of service delivery
- Multiple classes of Reliability and Availability
Verizon Scenario
Evolving Economics of Networking and Computing

- Historical Processing/Storage unit costs decreasing faster than Routing/Transport
- These trends drive placing cache (CDN) closer to end users
- Continuation of these trends will make Distributed NFV more economically compelling for other network services

**Content Delivery Cost** is a combination of **Processing + Storage** and **Routing + Transport Costs**
Goal: Customer Access to Distributed NFV Infrastructure

- Dynamic network services provided efficiently to customers
- Leverage most appropriate infrastructure to deliver the service
  - Efficient access to scalable services
  - Multiple reliability/availability classes of service
- Support for dynamic service graphs to enable distributed services
- Scalable highly-available service management
Lab Implementation Architecture
Challenges

● Deployment of Remote Compute Nodes across WAN
  ○ Extending L2 for provisioning
  ○ Network latency

● OpenStack Control Plane Communication
  ○ Network latency effect on the Message Bus and Database Access
  ○ Orchestration
  ○ Application deployment
  ○ Failure detection

● Service Resiliency
  ○ Headless operation
  ○ Service recovery

● Network Configuration, Maintenance and Troubleshooting
Lab Setup

Core Data Center

- Big Cloud Fabric Controller Cluster
- Spine switches
- TOR Leaf switches
- RHOSP Director (Undercloud)
- OpenStack Controllers (Overcloud)
- Compute nodes running Switch Light VX (virtual switch)

Remote Site-1

- TOR Leaf switches
- Compute nodes running Switch Light VX (virtual switch)

Latency Generator
Lab Setup: Physical Topology

**Core DC**
- BCF Controller Cluster
- RHOSP Director
- Openstack Controller
- Compute Nodes running SWL-VX

**Remote Site-1**
- Leaf
- Compute Nodes running SWL-VX

**Physical Topology**
- 10G Inband ports to the Leaf for virtual switch control path
- Management Switch for Out-of-band Management Network
- L2 link between Core DC & Remote Site-1 for BCF to physical switch control path
- Virtual Wire to send all traffic between Core DC & Remote Site-1, for Leaf to Spine data path
Test Objective

Validate fabric resiliency with WAN latency [0-40ms]

Control path latency

- Big Cloud Fabric out-of-band management network for physical switches
- Big Cloud Fabric in-band management network for virtual switches
- OpenStack control plane communications
Tests Performed

Ping from a VM in the Core DC to a VM on the Remote Site-1

Success Criteria: No ping packets lost

- Controller failures
  - Failover
  - Headless mode
- Spine and leaf switch disconnects and reconnects
- Spine and leaf switch interface up/down
  - Spine to leaf connectivity
  - Leaf to compute connectivity
- Spine and leaf switch reboots
Wrap Up

● **Telecom provider concerns**
  ○ Distributed NFV architecture is essential for a variety of carrier use cases and needs to be supported across the layers of the stack, from networking to message bus to applications
  ○ Latency and network availability might potentially affect both initial deployment and day two operation

● **Infrastructure providers’ answers**
  ○ Red Hat OpenStack Platform components are able to handle delays produced by deployment across the WAN
  ○ Big Switch Networks proved that the Big Cloud Fabric was resilient even across the WAN
Q & A