ACI for Telco-Cloud

Francois Lemarchand - Manager Sales Specialists
Héctor Fernández - Technical Solutions Architect
CKN - May 13, 2020
Datacenter Networking SP Focus

Distributed Networks  Transport Integration  Application Centric  Service Integration

Today’s first part
Assurance  Analytics and Telemetry

Today’s second part
Security

Very important but we had to chose – available in other CKN presentations
A new DC infra is required for 5G architecture

- **New IOT & MEC Services**
- **Dynamic slicing of 3GPP Resources**
- **Network exposure**
- **MEC hosting IT/Cloud native Policy Exposure**
- **3GPP 5G evolution**
  - More Spectrum low band + mmWave
  - New Core – Cloud Native, Edge & Slicing
  - New Radio – Low Latency, virtualized
- **5G Telco-DC Evolution**
  - Tight wan integration for Network slicing
  - Edge compute & vRAN optimized VIM
  - Distributed & hybrid cloud network automation

- **VRAN Real Time VIM**
- **Edge Distribution**
- **Cloud Native**
5G, MEC network architecture impact

- Edge cloud services (eg CDN) saves network capacity
- Distributed UPF & Peering for low latency / high BW services (eg Cloud Gaming)
- 5G NR URLLC + MEC enable ultra low latency applications (eg AR / VR)
RAN virtualization pushing DC to far Edge

Traditional distributed RAN infrastructure

Centralized & Virtualized, Open RAN evolution

- 10x less “active” sites
- All software flexibility
- Cloud based sharing

© 2019 Cisco and/or its affiliates. All rights reserved. Cisco Public
Telco cloud is expanding toward the Edge

Access

- Radio head
- vRAN
- distributed connectivity

Transport

- vRAN CU/DU
- 5G DataPlane
- 5G Core
- Edge services
- Internet CDN/Video
- MEC hosting

Core + Cloud

- OTT Edge DC
- Central Telco-DC
- Far Edge Telco-DC

Internet / Public Cloud

Number of sites:
- 10,000
- 1,000’s
- 100’s
- 10’s
- few
ACI Remote leaf – lean Edge & central SDN

Time synchronization protocols (PTP, SyncE) for Radio transport over Ethernet (ECPRI)

vRAN vDU (real time)

IP/MPLS Core

ACI Multi-Site Orch.

x10+ -> x100

ACI Controller

Leaf

Leaf

Leaf

Spine

Spine

Edge / Regional DC

ACI central controller & remote Leaf

Front/Mid/Back Haul

(Far) Edge DC

automated overlay VPN

E-CPRI

Nexus 9300 with G8275.2 PTP Telecom Profile and SyncE for virtual RAN applications

Stretches an ACI Fabric across the WAN to 100’s of locations – 500+ today & 1000’s in near future
5G transport slicing with Segment Routing

- Scalable for high service density / mesh
- Simplified operations with reduced protocol complexity
- Designed for Inter-domain connectivity
Simplicity always prevails

Simplified Network Operation
- Simplified automation & orchestration
- Efficient troubleshooting
- Robust routing code

Less protocols, Less configuration

Netconf (heavy configuration)
PCEP (centralized Path computation)
RSVP-TE (Traffic Engineering)
BGP-LU (Inter-AS routes)
LDP (default transport label)
IGP

Netconf (reduced)
PCEP (optional)
IGP

On-Demand SR policy & Automated Steering
Stateless Service Chain
Automatic 50ms Protection
Micro-Loop Avoidance

Seamless Deployment
Inter Domain

One common architecture to address all current capabilities

© 2019 Cisco and/or its affiliates. All rights reserved. Cisco Public
Traffic Engineering

Simple and scalable transit:
- No state in the transit network
- Flexible algorithm implemented in central SDN

Fair amount of configuration on Edge PE to steer traffic
- Map vpn route to specific IP NH (BGP, VRF...)
- Additional PBR, route pol.
- Create tunnel policy
- Resolve via central SDN/PCEP

Request: Low-Latency path to 7

SDN / SR-PCE

DC (BGP-SR)

WAN (IGP-SR)

PEER
FlexAlgo – further simplified SRTE

Path computed at the node (or central PCE) then encoded at the head-end router into the packet header.
- Potentially long label stacks
- Hard issue for computing intermediate detour path which respect initial intent

- With FlexAlgo – each node compute it’s own topology like standard IGP – just multiple ones (old multi-topology logic)
- Single label needed inside a domain
- Each node can compute adequate TI-LFA detour which respect constraints
ODN Per-Destination Automated Steering

- Automated Steering steers service routes on their matching (color + endpoint) SR Policy

BGP update:
1.1.1.1/32 via Node4
2.2.2.2/32 via Node4
BGP com color: 100

* Illustrative CLI
Per-Destination Automated Steering

- Automated Steering steers service routes on their matching (color + endpoint) SR Policy

BGP update:
3.3.3.3/32 via Node4
BGP com color:200

Min Delay
Min Cost

on-demand color 200
Match dscp 14 forward class 1
Match dscp 46 forwarding class 2
per-flow
forward-class 1
  sid-algorithm 1
forward-class 2
  sid-algorithm 128

* Illustrative CLI
Set by step detailed signaling example
Set by step detailed signaling example
ACI SR/MPLS & ODN benefits

- Inter-AS option A style
  - Heavy configuration of the PE
  - Scalability limits (i.e. BGP sessions)
  - Need to synchronize destination specific policies
  - Can be done with cross domain automation, dedicated DCGW (NSO, ACI / TSDN CFP)

- Greater Scaling (single BGP session)
  - Simplified provisioning (simplified VRF configuration – option B style)
  - Could replace DCGW function
  - Color routes / packets independent from VRF / endpoint specific configuration
  - SR MPLS bookending, single AS option
• With a simple ACI contract change to “low latency” map VM traffic to specific transport topology – in this case select lowest latency path to gaming server
**ACI SR / EVPN interwork B2B use case**

- Leveraging standard EVPN ACI can seamlessly map VM/Containers to external Enterprise VPN with shortest path routing
- Additional SR policies allows to select path based on highest metric/BW, HA criteria such as SRLG or security (country boundaries, encrypted links)
MEC & Hybrid cloud use cases

Distribute what you must:
- High BW downstream (eg CDN front cache)
- High BW upstream (eg video processing, analytics)
- Real time component (robotic control, AR)

Centralize the rest (Reduce capex/opex)
- Customer front end
- Management processes
- backend analytics, billing etc...

Ultra Low Latency
Ultra High Bandwidth

1ms – 10Gbps

MEC App Frontend
Operator Telco cloud
Operator Edge DC
Core / Internet
Secure VPN
Automated policies
Centralized O&M
Multi-Cloud

© 2019 Cisco and/or its affiliates. All rights reserved. Cisco Public
ACI Anywhere for hybrid cloud connectivity

- ACI Multi-Site Orchestrator
- ACI Controller
- MEC App Frontend
- Nexus/ACI Fabric
- SR xHaul
- Operator Edge DC
- Core / Internet
- Public Cloud
- BGP EVPN/IPSec
- Cloud ACI Controller
- MEC App Backend
- Native networking
- AWS APIs
- Azure APIs
- Native networking
- AWS
- Azure

© 2019 Cisco and/or its affiliates. All rights reserved. Cisco Public
Evolution from virtualization to containerization

- 25% of enterprises use containers in production
- 58% of IT Operations teams drive container adoption
- 52% said their incumbent networking vendor was "very helpful" in supporting container-based environments

<table>
<thead>
<tr>
<th>Server Type</th>
<th>% of Production Workloads Run Today</th>
<th>% of Production Workloads Run 24 Months from Now</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bare metal servers</td>
<td>35%</td>
<td>29%</td>
</tr>
<tr>
<td>Virtual machines</td>
<td>43%</td>
<td>37%</td>
</tr>
<tr>
<td>Containers</td>
<td>22%</td>
<td>34% &gt;50% Growth</td>
</tr>
</tbody>
</table>

Source: ESG “Trends in Modern Applications Environments”, Aug-19
ACI fit in Cisco’s Telco DC portfolio

- Overlay SDN: VIM integration (Neutron, CNI) Service chaining, LB, micro-seg
- DC Automation & Management: Multi-fabric Mgt, ZTP, O&M, Analytics
- Distributed DC Fabric: Remote leaf, EVPN, SR/MPLS

Cloud ACI for Hybrid Cloud

- IP WAN xHaul, Core & Edge (MPLS, SR, L3VPN/EVPN)
**Drastic SDN simplification**

*Overlay Controller Model*

- **NFVO/VNFM**
  - Some Proprietary API
  - Performance Bottleneck
  - Immature & HW dependent

- **VNF**
  - SDN vswitch
  - SmartNIC

- **Overlay SDN**
  - Fabric controller
  - Switch Fabric

- **Basic integration still required (eg day 0)**
- **Advanced integration for WAN, PNF**
- **Overlay/underlay correlation, operations...**

- **SmartNIC**
- **Performance Bottleneck**
- **Immature & HW dependent**

**Cisco ACI**

- **NFVO/VNFM**
  - Standard API
  - Standard for CP/Mgt
  - Standard SRIOV zero overhead

- **VNF**
  - VIM OVS
  - Standard NIC

- **VIM**
  - Neutron plugin
  - VIM LCM

- **Switch Fabric**
  - Nexus ACI Fabric

- **APIC**
- **APIC**

- **No SDN overlay! Simplified/optional VIM integration**
- **All hardware performance, zero SW overhead**
- **Open API, Open protocols (New SR-MPLS/EVPN!)**

- **Complex integration & Maintenance**
- **Unpredictable performances**
- **VNF onboarding uncertainty**
OpenStack and ACI

You make networking possible
OpenStack Neutron Challenges

- Non distributed L3 services
- No underlay visibility
- Performance
- Complexity of troubleshooting
Why Cisco ACI and OpenStack?

- **Distributed, Scalable Virtual Networking**
  - Full Neutron Node datapath replace
  - Fully distributed Layer 2, anycast gateway, DHCP, and metadata
  - Distributed NAT and floating IP address

- **Hardware-Accelerated Performance**
  - Automatic VXLAN tunnels at top of rack
  - No wasted CPU cycles for tunneling
  - Optional use of SRIOV, OVS DPDK (VPP support roadmap)
  - VNFs automated BGP peering SVI with 64-way line-rate Load Balancing

- **Integrated Overlay and Underlay**
  - Fully managed underlay network through Cisco® APIC
  - Capability to connect physical servers and multiple hypervisors to overlay networks

- **Operations and Telemetry**
  - Troubleshooting across physical and virtual environments
  - Health scores and capacity planning per tenant network
ACI Unified plugin is a Neutron ML2 Plugin

- ML2 plugin: a framework allowing OpenStack Networking to utilize the variety of layer 2 networking technologies.
- When running the ACI integration, The following Type and Mechanism Drivers will be used:
  - Type Drivers
    - opflex
  - Mechanism Drivers
    - apic_aim
ML2 (Neutron) – APIC Mapping

- With the ML2 Standard Neutron model, the following mapping happens.
- All the operations are done on OpenStack through Horizon, CLI or Heat

<table>
<thead>
<tr>
<th>Openstack/Neutron Object</th>
<th>APIC Object</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project</td>
<td>Tenant and Application Profile Name</td>
</tr>
<tr>
<td>Network</td>
<td>EPG and Bridge Domain</td>
</tr>
<tr>
<td>Subnet</td>
<td>Subnet</td>
</tr>
<tr>
<td>Router</td>
<td>Contract, consumed and provided by any EPGs corresponding to the Neutron Networks connected to the router.</td>
</tr>
<tr>
<td>Security Group</td>
<td>Host Protection Profile (HPP) policies</td>
</tr>
</tbody>
</table>
ACI Neutron Plugin Main Components

Main Components are:
- ACI ML2 Plugin
- AIM
- Opflex Agent (Optional)
- OpFlex Proxy
- VMM Manager
ACI + OpenStack – With OpFlex Agent

Architecture

OpFlex for OVS

- Open Source OpFlex agent extends ACI into the host
- OpFlex Proxy exposes new open API in ACI fabric

Cisco ACI fabric provides line rate distributed routing and switching capabilities

OVS Rules programmed by the OpFlex Agent are used for policy enforcement

Cisco ACI fabric provides line rate distributed routing and switching capabilities

OpFlex Proxy receives Policies from APIC

OpFlex Agent receives Policies from ACI leaf

APIC Driver converts neutron to Cisco ACI Policy

OpFlex Agent

ML2 Plugin

Neutron-server

Controller node

AIM

Bond0

OpFlex-agent

OvS

vm1

vm2

vm3

vm4

L2/L3

VLAN or VXLAN

© 2019 Cisco and/or its affiliates. All rights reserved. Cisco Public
ACI + OpenStack – Without OpFlex Agent

Architecture

PhysDom Integration

- In some scenarios Opflex agent is not installed, in which case we talk about PhysDom integration
- A typical use case for this is when we have SR-IOV based compute hosts not having OVS
- In this model the EPG is configured with VLAN static binding to the compute nodes.
### Compute nodes: OpFlex vs non OpFlex

<table>
<thead>
<tr>
<th>Mode / Optimization</th>
<th>Distributed Routing on host</th>
<th>Distributed Routing on Leaf</th>
<th>Distributed NAT</th>
<th>DHCP Optimization</th>
<th>Metadata Optimization</th>
<th>OVS rules or IPTables</th>
</tr>
</thead>
<tbody>
<tr>
<td>OpFlex</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>OVS Rules</td>
</tr>
<tr>
<td>Non OpFlex/PhysDom</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>IPTables</td>
</tr>
</tbody>
</table>

- You can mix and match modes on the same hosts

<table>
<thead>
<tr>
<th>Acceleration mode possible on the same host</th>
<th>OVS with OpFlex</th>
<th>SR-IoV</th>
<th>OVS-DPDK no OpFlex</th>
</tr>
</thead>
<tbody>
<tr>
<td>OVS with OpFlex</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>SR-IoV</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>OVS-DPDK no OpFlex</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>
Distributed Routing and Policy Enforcement

- Traditionally in OpenStack the routing is done on the servers hosting neutron services only.
- With ACI integration the opflex-agent is taking care of the routing of the VMs. Since each compute node has an opflex-agent, the routing is done in a distributed manner.

Also, the opflex-agent performs local policy enforcement through OVS rules locally on the same hypervisor where the instance lives.
Distributed DHCP

- Traditionally VMs get IP from Neutron DHCP Server
- Agent-OVS learns info of the VM from Endpoint Files
- Agent-OVS responds to the VMs with DHCP responses
- DHCP allocation and options passed back to Neutron server.

DESCRIPTION

OpFlex-agent will retrieve VM IP information from EP file and provide to the VM through DHCP DORA.
Distributed Metadata

Traditionally in OS VMs get the metadata information from the service running on Neutron Server.

Neutron metadata agent is reading the Service File.

Metadata agent locally performs proxy.

Metadata agent updates the neutron server with VM Metadata.
Distributed NAT

- Floating IP configured by OpenStack Neutron using standard mechanism
- OVS performs NAT function using OpenFlow rules from OpFlex agent for Floating IP

OvS functions as distributed virtual router for VMs. If destination network is external to OpenStack router directly connected subnets and NAT policy is defined, it will source NAT the VM IP.
External Networks and NAT

- External networks can be
  - Dedicated for a tenant – with or without NAT
  - Shared by multiple tenants – NAT
- One tenant may have one or many external network, with or without NAT
- Control Plane vs Data Plane
ACI Plugin NFV features

- Neutron Trunk ports
- Neutron SVI integration (BGP Peering)
- Neutron SFC integration
- OVS-DPDK support
Service Chaining – deployed service graph
Hierarchical port binding (HPB)

- Net type opflex
- 3rd party agent or mech driver asks for vlan port binding
- A local vlan segment is created
- port is bound using local segment
- opflex plugin creates static bindings on APIC to allow the new vlan segment
- This segment is stored in segments table
Symmetric PBR

Thanks to Symmetric PBR, incoming and return traffic go to same PBR node.
Symmetric PBR with Resilient hashing

Some traffic could be load-balanced to different PBR nodes that don’t have existing connection info.
N+M symmetric PBR with resilient hashing

Standby node is not used unless an active node is down.
K8s and ACI

You make networking possible
Why ACI for Application Container Platforms

Turnkey solution for node and container connectivity

Flexible policy: Native platform policy API and ACI policies

Hardware-accelerated: Integrated load balancing and Source NAT

Visibility: Live statistics in APIC per container and health metrics

Enhanced Multitenancy and unified networking for containers, VMs, bare metal

Fast, easy, secure and scalable networking for your Application Container Platform
Cisco ACI CNI plugin features

- IP Address Management for Pods and Services
- Distributed Routing and Switching with integrated VXLAN overlays implemented fabric wide and on Open vSwitch
- Distributed Firewall for implementing Network Policies
- EPG-level segmentation for K8s objects using annotations
- Consolidated visibility of K8s networking via VMM Integration
Visibility

**EPG - kube-default**

<table>
<thead>
<tr>
<th>End Point</th>
<th>MAC</th>
<th>IP</th>
<th>Learning Source</th>
<th>Hosted Server</th>
<th>Reporting Controller Name</th>
<th>Interface</th>
<th>Multicast Address</th>
<th>Encap</th>
<th>Configured Access Policies</th>
<th>Contracts</th>
<th>Controller End Points</th>
</tr>
</thead>
<tbody>
<tr>
<td>frontend-1768586195-mjLh</td>
<td>10:33:02:FF:3E:02</td>
<td>10.33.1.11</td>
<td>learned</td>
<td>k8s-62</td>
<td>Kubebryte</td>
<td></td>
<td>225.22.176.81</td>
<td>vlan-7998766</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>redis-slave-3837261623-qw894</td>
<td>02:A0:41:D4:83:0F</td>
<td>10.33.6.154</td>
<td>learned</td>
<td>k8s-63</td>
<td>Kubebryte</td>
<td></td>
<td>225.22.176.81</td>
<td>vlan-7998766</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Dual level Policy Enforcement by ACI

Both Kubernetes Network Policy and ACI Contracts are enforced in the Linux kernel of every server node that containers run on.

Containers are mapped to EPGs and contracts between EPGs are also enforced on all switches in the fabric where applicable.

Both policy mechanisms can be used in conjunction.

Native API Default deny all traffic

```yaml
apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
  name: default-deny
spec:
  podSelector: {}
  policyTypes:
  - Ingress
  - Egress
```
Mapping Network Policy and EPGs

Cluster Isolation

Single EPG for entire cluster.  
(Default behavior)  
No need for any internal contracts.

Namespace Isolation

Each namespace is mapped to its own EPG.  
Contracts for inter-namespace traffic.

Deployment Isolation

Each deployment mapped to an EPG  
Contracts tightly control service traffic.
ACI and Kubernetes External Services - Summary

- Two options
  - Exposing services via ingress
  - Exposing up to 500 services directly with Service Graph with PBR
- Can be used at the same time even for the same service
ACI and Kubernetes Load Balancer
ACI and Kubernetes Ingress
Kubernetes and NAT

• ACI CNI now support distributed SNAT

• POD Initiated traffic can NAT'd
  • SNAT IP: Single IP or Range
  • Ability to apply SNAT Policy at different levels:
    • Cluster Level
    • Namespace
    • Deployment
    • LoadBalanced Service
Take away

You make networking possible
Different workload types
Drastic simplification for Telco DC

- NetOp Day 2
- Overlay SDN
- Fabric Automation
- Load Balancers
- Edge Routing

Traditional Telco Networking Stack

ACI Fabric

- Drastic network simplification
- Application visibility & Control
- Open: Multi-VIM, Neutron/CNI, EVPN
- NFV ready (GTP LB, Service chaining)
- 5G Scale ready to network (Far) Edge
- 5G performance ready – full Hardware offload
- 5G Slicing/automation – Segment Routing
- HW Telemetry / ML Ops
Thank you