OpenContrail as SDN controller for NFV infrastructure in AT&T network

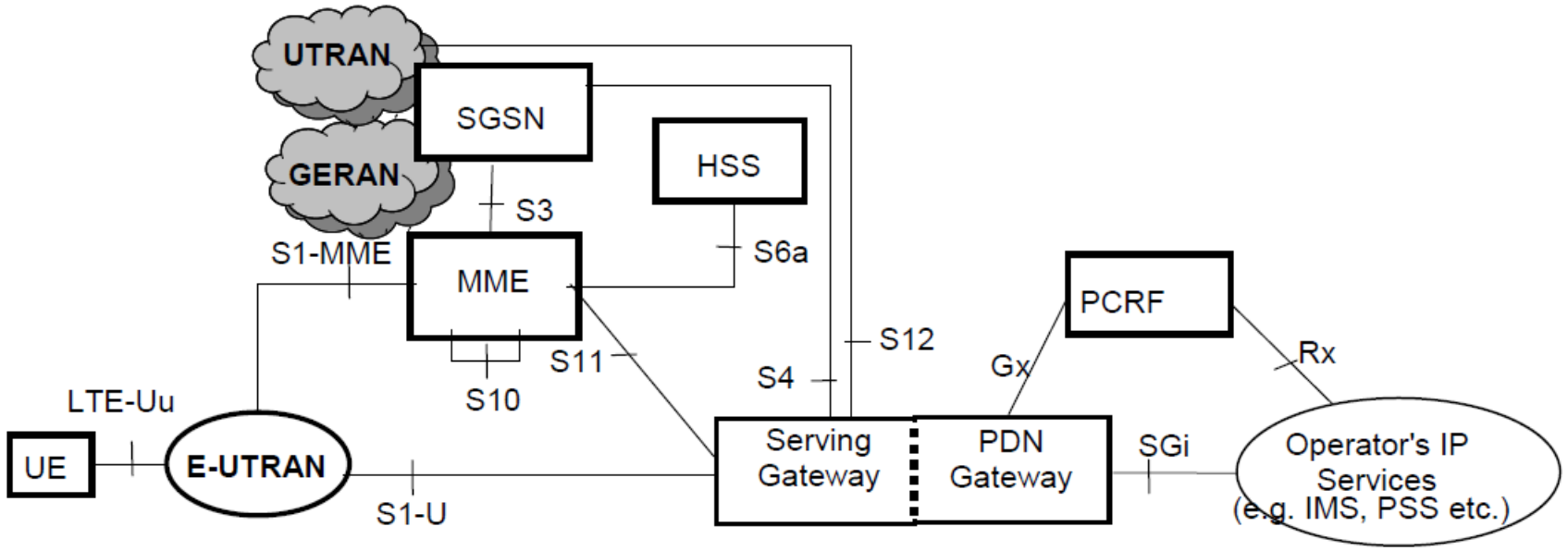
Alexey Gorbunov

Network Architect

CCIE 41088

What we're doing: AT&T Wireless Mobility Network

We're a supporting a large network infrastructure to connect all Mobility network elements.
Our network is providing Wireless services for millions of subscribers.
Below is the 3GPP Mobility Network. We need to interconnect all 3GPP entities with load balancers, firewalls, routers, monitoring systems, proxies and etc.



3GPP TS 23.401

New drivers for Mobility Network

IOT, 5G and multimedia services are main drivers for Mobility Network. They are defining new requirements and challenges for Mobility Networks which need to be solved. High level overview of requirements which are applicable to network:

1. Fast deployment of new services.
2. Ability to scale dynamically and fast.
3. Deep packet inspection of application flows and dynamic policy routing based on results.
4. Support of multiple customers with overlapping ip addresses.
5. Securing network against IOT threats.
6. Full support of IPv6.

Let’s compare traditional physical infrastructure and architecture based on SDN.

Physical Infrastructure vs NFV/SDN

Time to deploy Network Infrastructure

- Average time to add new Network Instance is a few months
- Capacity planning needs to be done in way ahead.
- Software upgrades require interaction multiple teams.
- Provisioning is manual and is very time consuming
- Limitation of supported numbers of VLAN.
- Policy chaining is complex and not dynamic.

Improved time

- Average time to add new Network Instance is few hours.
- Transparent scaling without impacting other network nodes
- Quick capacity augmentation via incremental VM spin-off
- Savings on hardware (CPU) costs
- All configuration is done automatically
- In-service VM-by-VM upgrade
- Significantly reduced time-to-market

Requirements for SDN

We have tried different approaches to introduce SDN in our network. We have identified following requirements for SDN:

1. Scalability to support hundreds of nodes.
2. Performance to meet Service Provider requirements.
3. Reliability.
4. Multi-customer support.
5. Low latency.
6. Full QOS support.
7. Process to get new features. Telco cloud has different requirements then enterprise or web service provider. We need to get them fast.
8. Integration to the existing network.
9. Support of all traditional network security features

Comparison of Different approaches

Name	**Integration to existing MPLS network.**	**Advantage**	**Reliability**	**Performance**	**Scalability**	**Contribution**
VMware	Easy	Very mature solution	Very reliable	Excellent	Difficult to scale	**Difficult**
Commercial Physical Device combining Firewall, LB, router	Easy	Significant cost saving on hardware	Very reliable with on-site support	Excellent	Depends on vendor support	Depends on vendor support
Open Contrail SDN	Easy	Significant cost saving, ability to contribute	Reliability is around 99.9%	Excellent	Excellent	Very easy to add new features and correct bugs
Any OpenFlow Contoller	Not clear how					

Comparison of Different approaches

VMware

We have successfully deployed fully virtualized production network on VMware platform. VMware solutions with ESXI hosts and vSphere are very reliable with good performance.

Although it's very difficult to deploy it with the scale which we need. Also there is a big room for improvement in packet per second performance.

Proprietary solution to combine network elements to one physical box.

We also have successfully deployed a proprietary solution with combining load balancer, router, and firewall into on physical box. It works very well. But it's not open source so there are challenges with support and development of future releases.

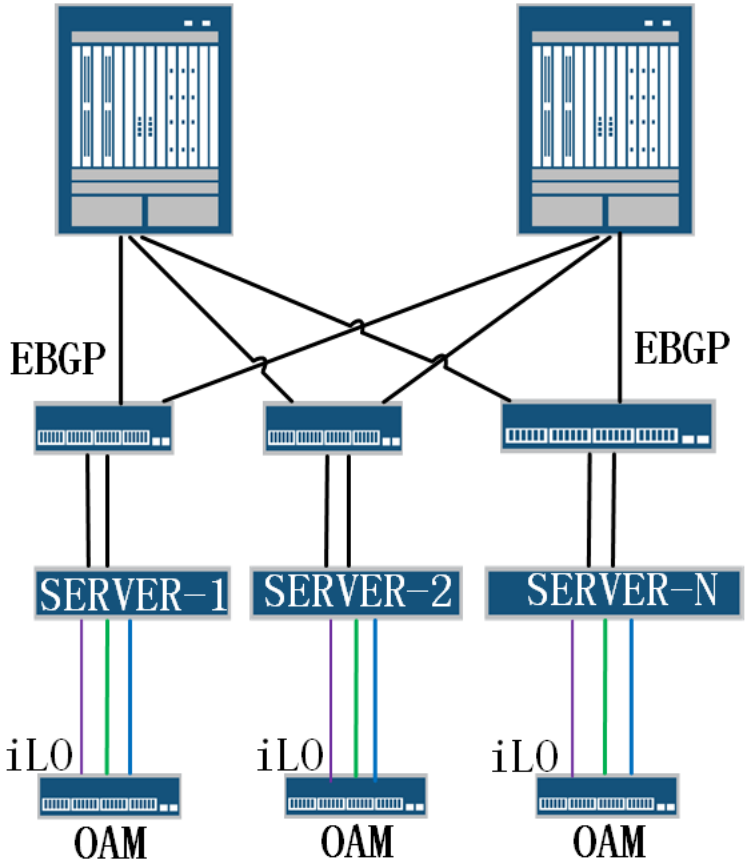
Why Open Contrail was selected?

- ✓Only Open Contrail supports all needed features to virtualize all network and mobility elements.
- ✓Open Contrail can be natively integrated into existing MPLS network.
- ✓Open Contrail is based on very mature MP-BGP technology.
- ✓Open source provides capability to troubleshoot and commit bugs faster.

All these items are very important to successfully deployed NFV with SDN.

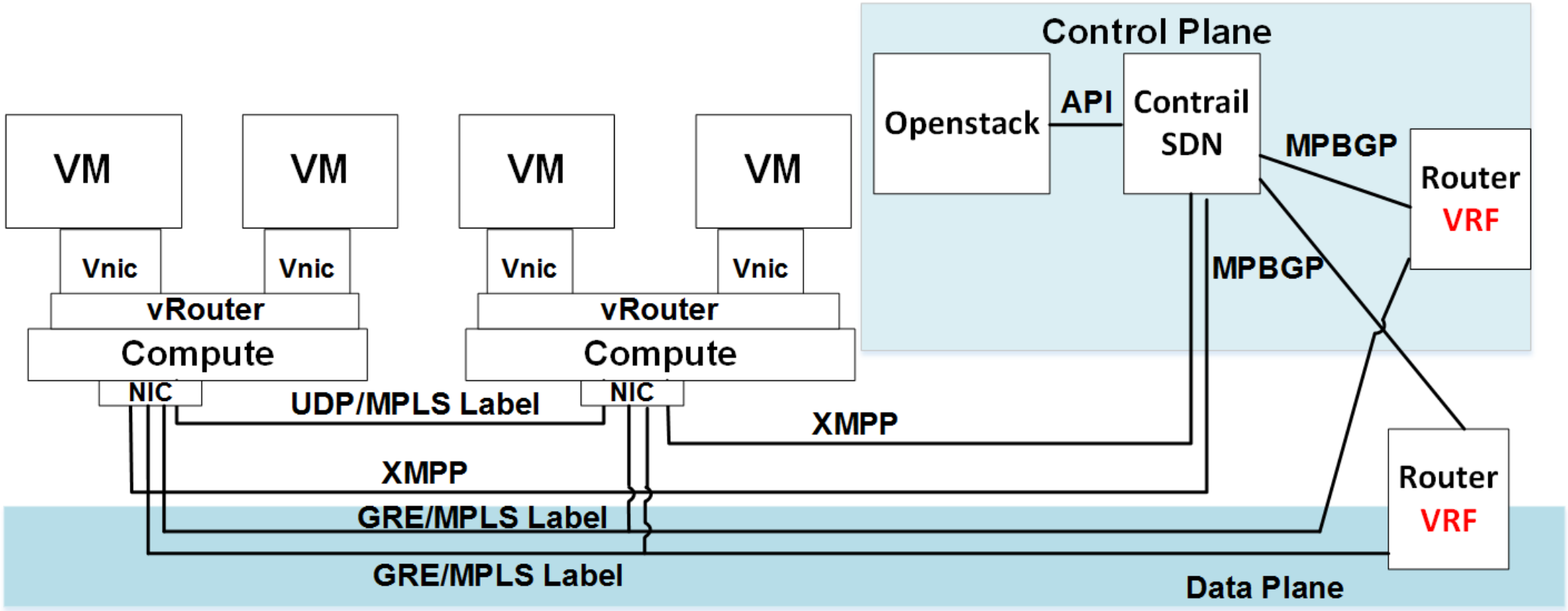
What is the Open Contrail?

- Open Contrail is a overlay SDN solution. It means that you need to deploy an underlay for Open Contrail. Main goal of underlay is to provide E2E connectivity between computes and external router.



What is the Open Contrail

High level overview of Open Contrail. Open Contrail can be used with OpenStack and Containers Networking.

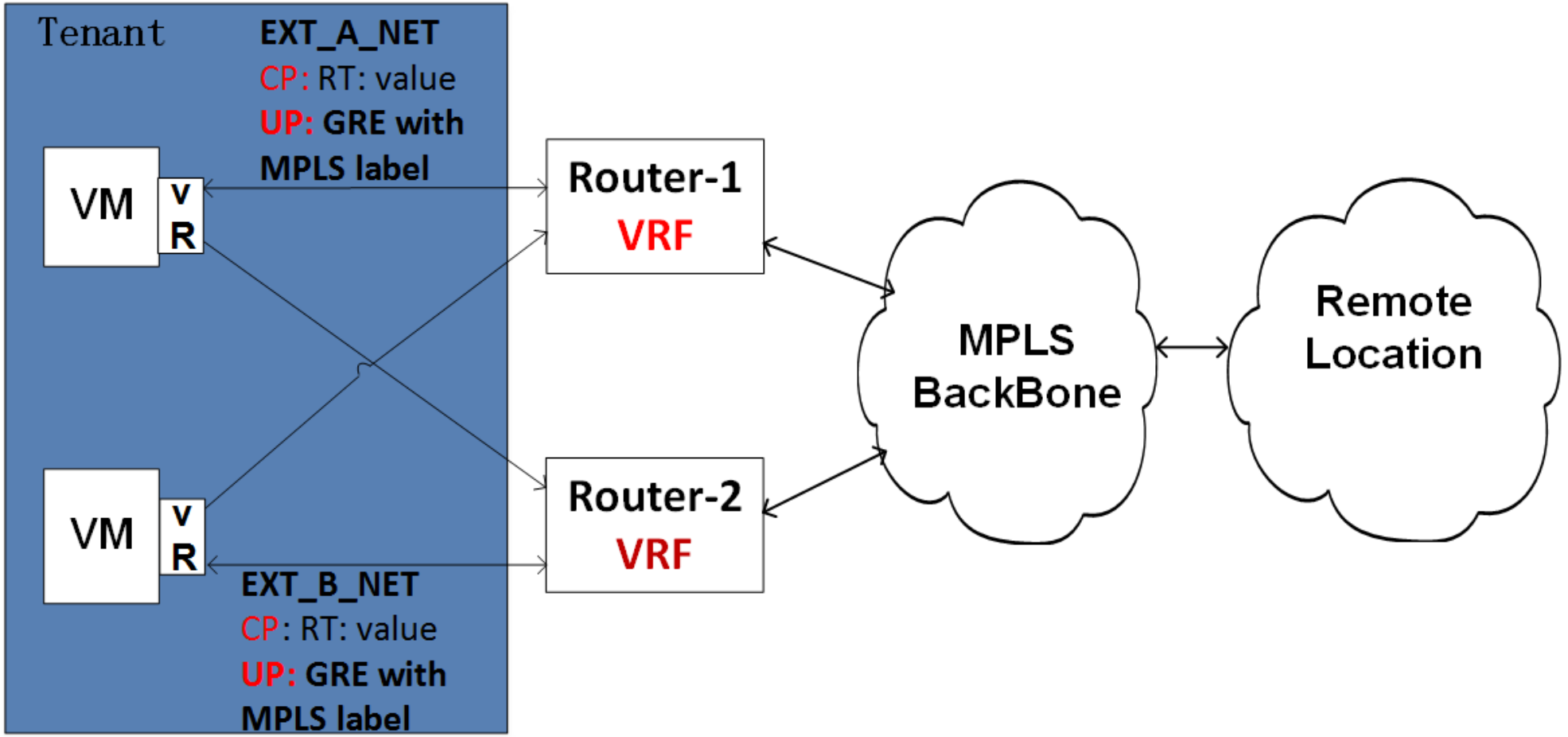


Open Contrail: Connectivity to an External Router

Control Plane:
Open Contrail networking can be easily integrated with MPLS network. Any external neutron network would have a BGP Route Target. Contrail controller would announce this network to the external physical router via MP-BGP. Based on the router configuration it would be injected to the router VRF.

User Plane:
Compute host would establish a GRE tunnel to the router. For each tenant network MPLS labels is assigned. Packet would reach physical router and based on MPLS label would be mapped to the VRF. UDP tunnel is used between compute hosts.

Open Contrail: Connectivity to an External Router



Open Contrail: Connectivity to an External Router

Here is a wireshark trace with ssh packet and output of routing table to make it more visual and clear:

```
⊞ Frame 955: 137 bytes on wire (1096 bits), 137 bytes captured (1096 bits)
⊞ Ethernet II, Src: JuniperN_44:a4:80 (f4:b5:2f:44:a4:80), Dst: HewlettP_10:74:b0 (8c:dc:d4:10:74:b0)
⊞ Internet Protocol Version 4, Src: 10.10.10.10 (10.10.10.10), Dst: 172.17.0.6 (172.17.0.6)
⊟ Generic Routing Encapsulation (MPLS label switched packet)
  ⊞ Flags and Version: 0x0000
    Protocol Type: MPLS label switched packet (0x8847)
⊟ MultiProtocol Label Switching Header, Label: 17, Exp: 0, S: 1, TTL: 62
    0000 0000 0000 0001 0001 .... .... .... = MPLS Label: 17
    .... .... .... .... .... 000. .... .... = MPLS Experimental Bits: 0
    .... .... .... .... .... ...1 .... .... = MPLS Bottom Of Label Stack: 1
    .... .... .... .... .... .... 0011 1110 = MPLS TTL: 62
⊞ Internet Protocol Version 4, Src: 172.20.214.28 (172.20.214.28), Dst: 10.250.250.5 (10.250.250.5)
⊞ Transmission Control Protocol, Src Port: 51247 (51247), Dst Port: 22 (22), Seq: 1, Ack: 1, Len: 43
⊞ SSH Protocol
```

show route table MOB_OAM 10.250.250.5

10.250.250.5/32 *[BGP/170] 00:17:33, MED 100, localpref 200, from 172.17.0.10

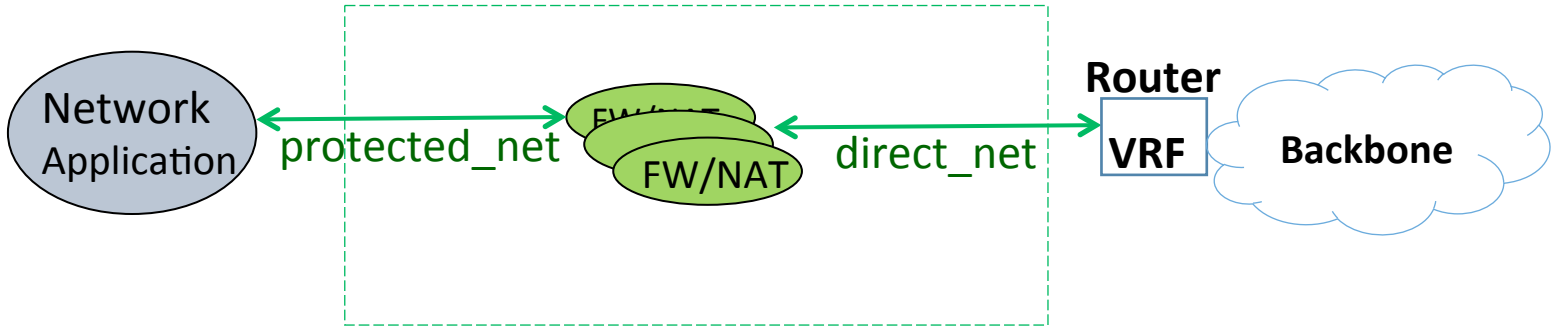
AS path: ?, validation-state: unverified

> via gr-0/1/0.32774, Push 17

Open Contrail: Features

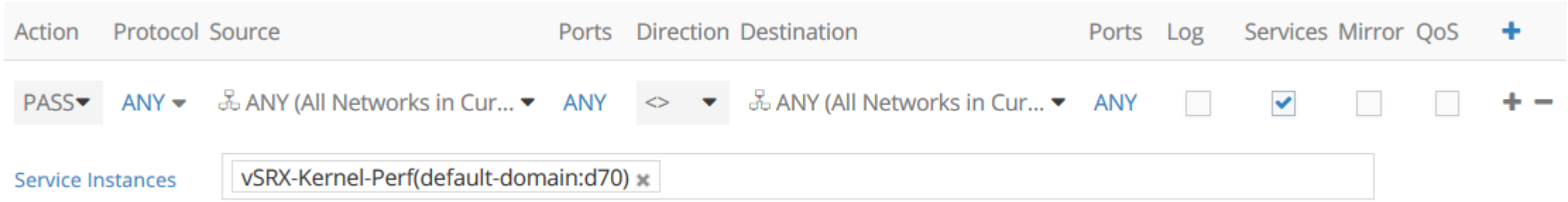
Policy routing or Service Chaining.

With Policy Routing you can route traffic based on predefined policy rules. For example, static policies might specify different path for HTTP/HTTPS and DNS traffic. HTTP/HTTPS traffic might be sent to firewall inspection and DNS packets to DNS Load Balancer.



Open Contrail: Features

Policy routing or Service Chaining.



Open Contrail: Features

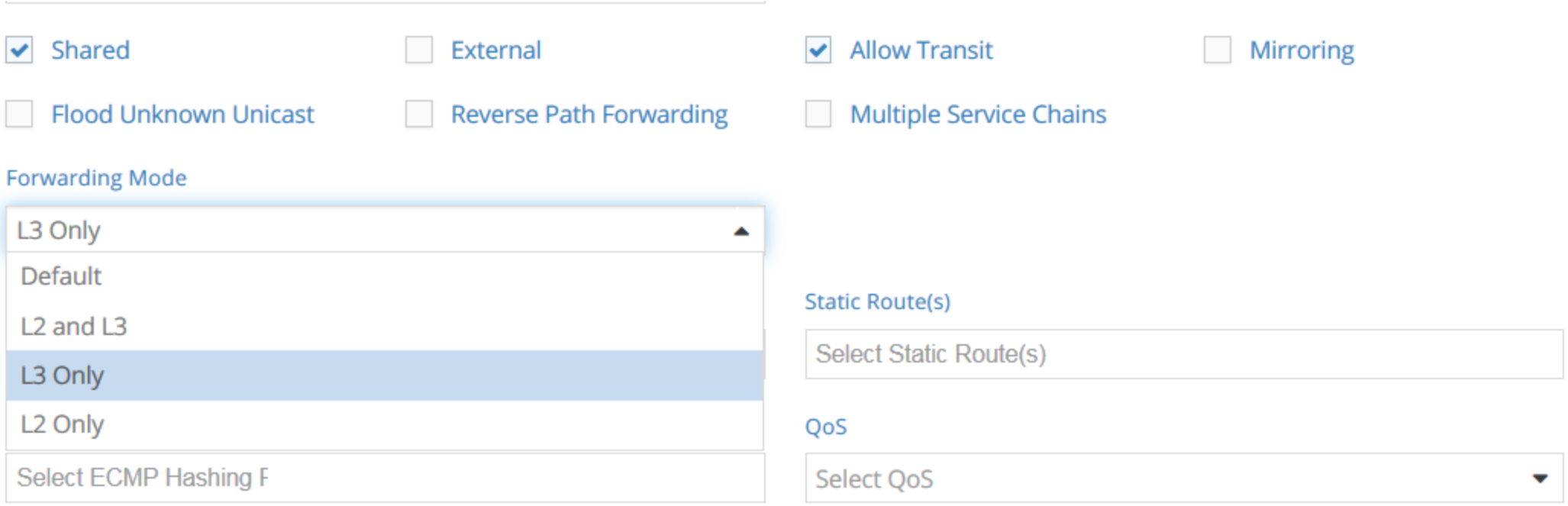
✓**Public IP addresses without NAT.**

SIP doesn’t work very well with NAT. So all SIP applications would benefit.

✓**L2 network**

Traditional telco applications heavily use L2 networks.

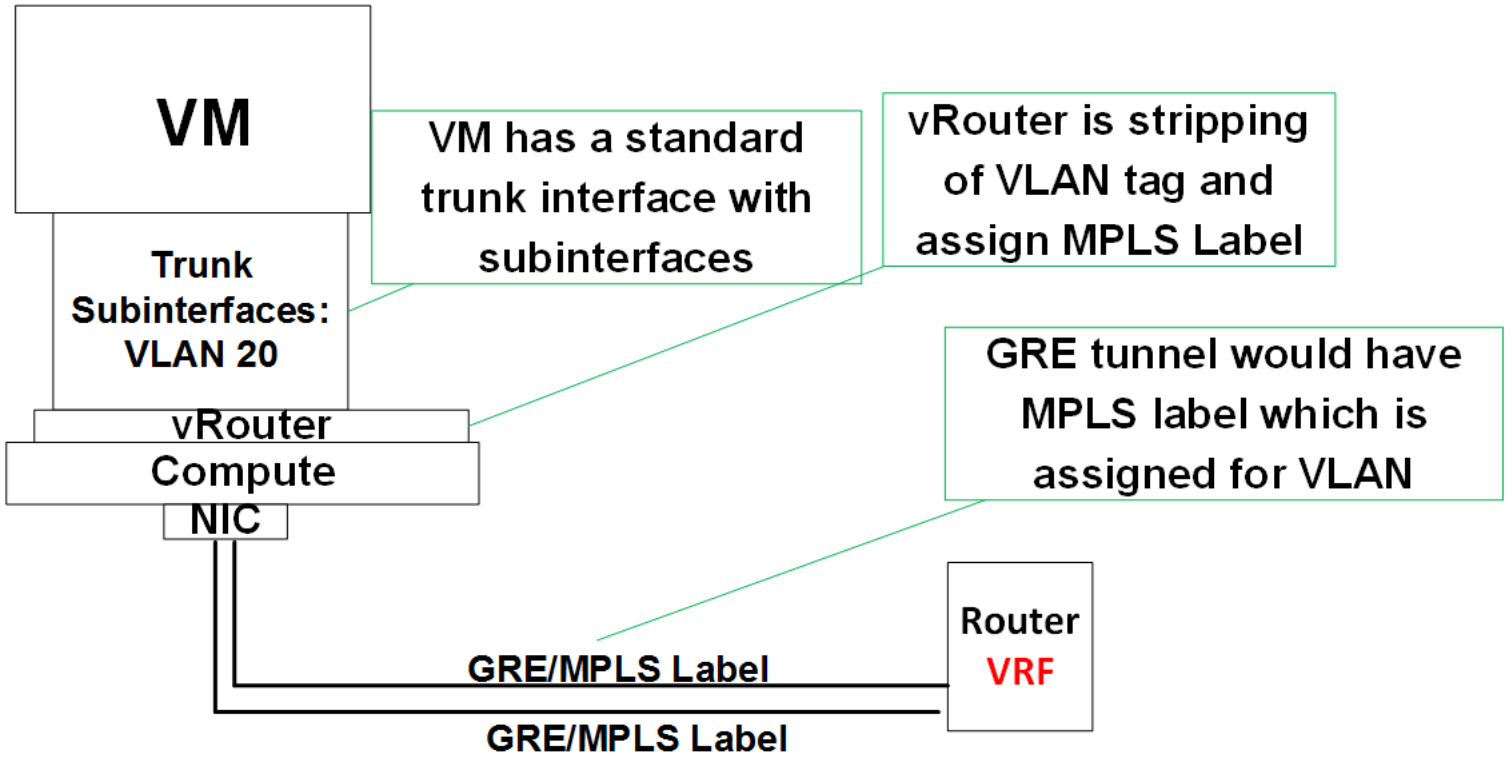
Open Contrail: Features



Open Contrail: Features

✓**Trunking**

Contrail perfectly supports all applications which need to have 802.1q trunk. Customers in SP network are often separated with VLAN tagging.



Open Contrail: Features

✓**QoS Marking and Remarking**

It's extremely important for all VOIP applications. Please note that in the underlay (fabric) network we don't have any qos policies. But QOS polices exist in the Backbone network.

✓**Support of priority queues for QoS.**

It's very common requirement for VOIP applications. It's applied on the hardware level.

✓**Jumbo Frame support**

Almost all Telco application require jumbo frames.

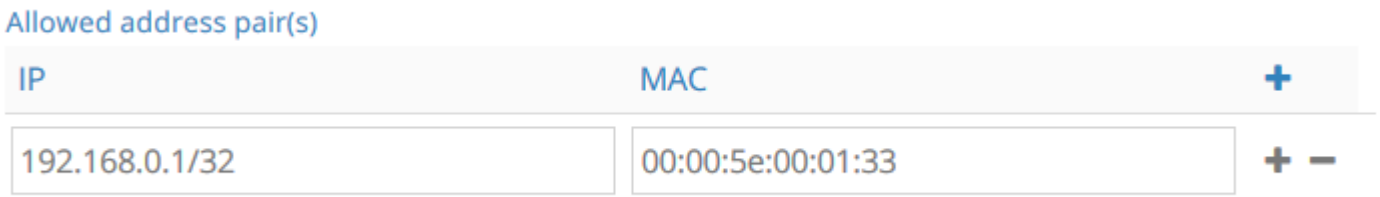
✓**Full support of SCTP protocol.**

This requirement is very important for all wireless mobility elements. SCTP protocol is still heavily used in 3G/4G network.

Open Contrail: Features

✓**High availability**

SDN and cloud technologies should provide High Availability features similar to VRRP/HSRP. Contrail has implemented feature “Allowed Address Pair”. AAP allows assigning loopback to multiple VMs and load sharing between them based on ECMP hash.



Open Contrail: Features

✓**Static route Advertisement**

Some Telco applications need to participate in routing decisions. They might inject routes to advertise new prefixes. The most simple example is advertising loopback on virtual router.

interface_route_table_routes: { interface_route_table_routes_route: [{ interface_route_table_routes_route_prefix: { get_param: sctp-b-static_route_SGs-2 }, }], }

Show route on Juniper:

2600:ae00:3001:2405::fe/128

**[BGP/170] 4d 21:37:35, MED 200, localpref 100, from 172.16.0.4*

AS path: ?, validation-state: unverified

*> **via gr-0/1/0.32788, Push 262***

[BGP/170] 4d 21:37:35, MED 200, localpref 100, from 172.16.0.6

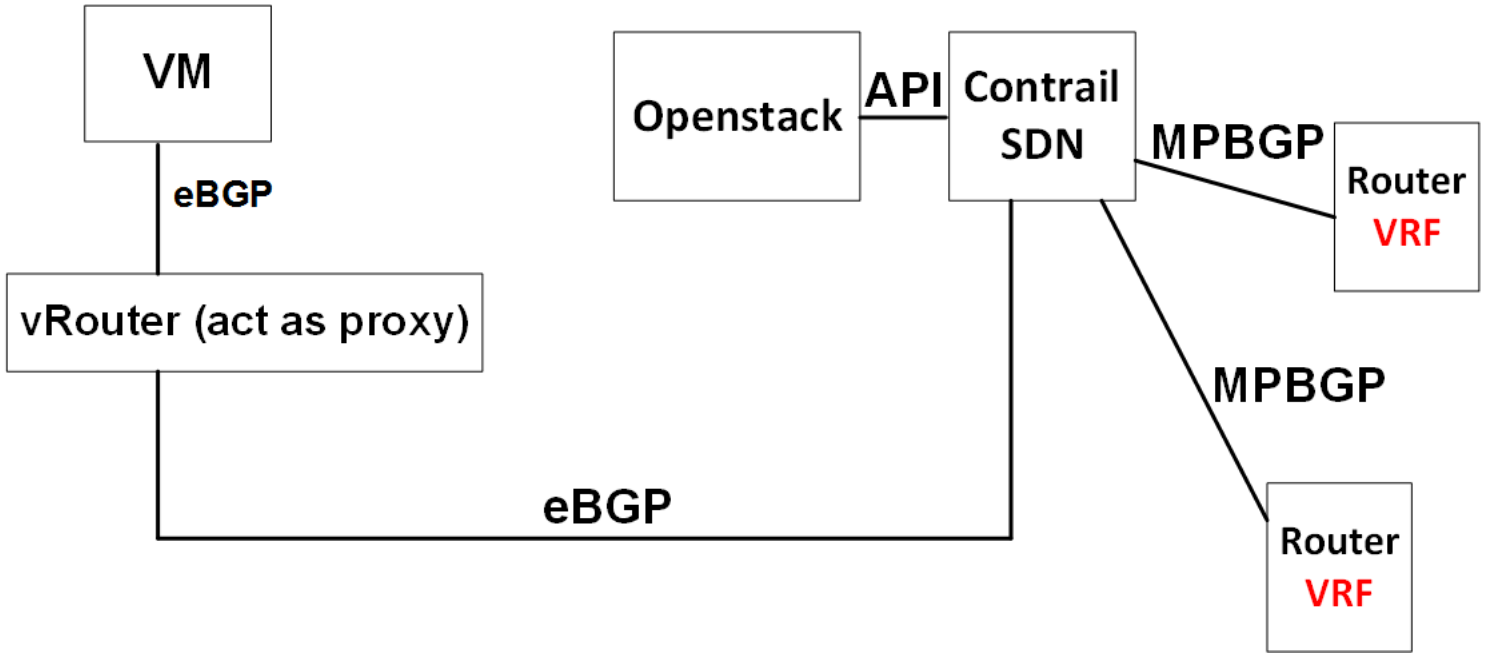
AS path: ?, validation-state: unverified

*> **via gr-0/1/0.32788, Push 262***

Open Contrail: Features

✓**Dynamic routing (BGP as a service)**

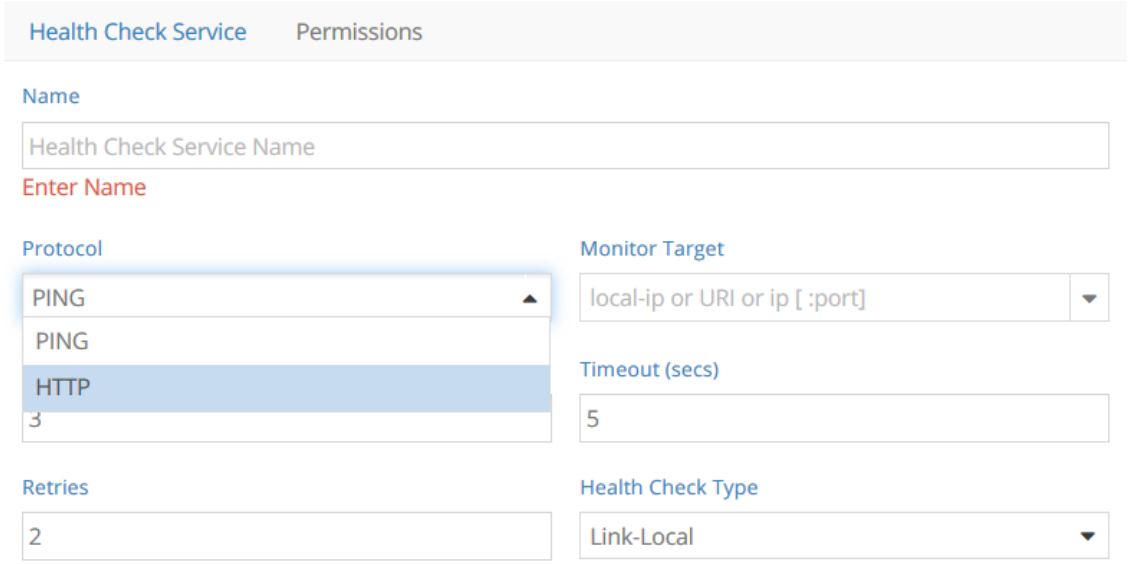
Some Telco applications need to participate in routing decisions. They might inject routes to advertise new prefixes. For example virtual router might advertise network prefixes via BGP.



Open Contrail: Features

✓**Port Health Check**

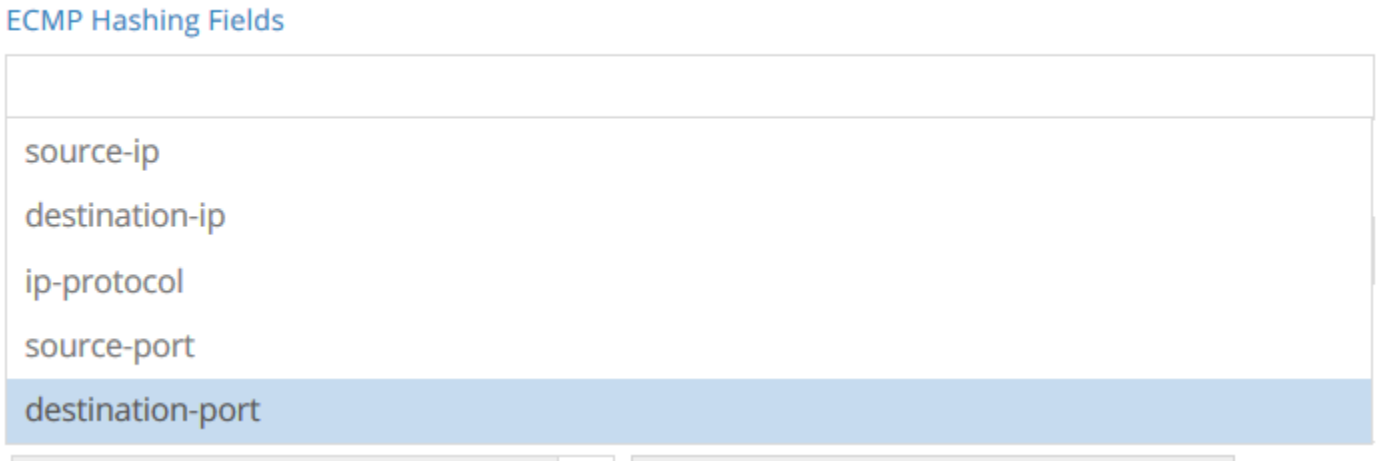
Maximum high availability of SDN is around 99.9%. Port health check helps to remove VM from routing in case of failure. It's very closed to the Cisco IP SLA.



Open Contrail: Features

✓**ECMP hashing**

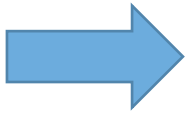
Hashing would help to make routing more predictable and load balance traffic more equally.



Automation with Open Contrail

Open Contrail is fully integrated with OpenStack. Openstack support automation with heat templates. Contrail provides heat resources which you can use to create all network resources that are mentioned in this presentation. You can describe the whole network just with heat templates and basic configuration. To automate with heat templates you need to have two files. First file is environmental file which have all variables like ip addresses, names and etc.

Environmental File: base.env
Yaml File: base.yml



heat stack-create BASE -e base_mcc.env -f base_mcc.yml

```
+-------------+-------------------+
| stack_name  | stack_status      |
+-------------+-------------------+
| BASE        | CREATE_COMPLETE   |
```

Automation with Open Contrail

Here is an example of L3 network with route targets:

ENV File:

fq_name: 'default-domain:Base'
sctp_a_net_name: Base
sctp_a_net_cidr: 107.243.37.224
sctp_a_allow_transit: True
sctp_a_forwarding_mode: l2_l3
sctp_a_net_rt: "target:13979:105717“

YML File:

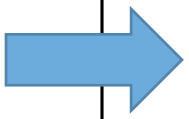
```
sctp_b_net:
  type: OS::ContrailV2::VirtualNetwork
  depends_on: [ template_NetworkIpam_sctp_b ]
  properties:
    fq_name: { get_param: fq_name }
    name: { get_param: sctp_b_net_name }
    virtual_network_properties:
      {
        virtual_network_properties_allow_transit: { get_param: sctp_b_allow_transit },
        virtual_network_properties_forwarding_mode: { get_param: sctp_b_forwarding_mode },
        virtual_network_properties_rpf: { get_param: sctp_b_rpf },
      }
    route_target_list:
      {
        route_target_list_route_target: [{ get_param: sctp_b_net_rt }],
      }
```

Migration to Heat Templates

Traditional design of any network element (firewall, load balancer and etc) involves writing document which includes configuration steps.

Process is slow and vulnerable to human error. Document might include:

- **Trunking**
- **VLAN assignment**
- **Policy route**
- **L2 networking**
- **Modification of BGP policy.**

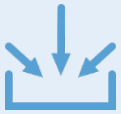


Heat templates describe the whole VNF design including all networking and configuration part.

They include:

- **Networks/Subnets**
- **Load Balancing**
- **Health Checks**
- **Cinder Storage Volumes**
- **VMs**
  - **Image & Flavor**
  - **Availability zone and Scheduler hints (affinity)**
  - **Name**
  - **Interfaces (ports)**
    - **Connected to specific networks**
    - **IP addresses**
  - **VM configuration information**

Network Design and Implementation



Collecting Requirements

1. Security
2. Capacity
3. High availability

Developing Heat Templates

Develop heat templates in the sandbox lab.

Identify, test and develop contrail features for the networking.

Prototyping whole Network.

Testing Heat Templates

Functionality is verified in the production like environment

Finalizing

The whole solution is finalized and is stored in the shared location

- ✓ **Functional Heat templates which have whole network description.**
- ✓ **Utilizing best practices from software development**

Lessons Learned

✓**Cloud Availability is around 99.9%.**

It means that you need to utilize all possible features for health check, process monitoring in guest VM (For example “supervisord” in Linux) and design the application in more tolerant way.

✓**IPv6 requires more attention**

If your focus is only ipv4 then it’s very easy to miss ipv6.

✓**Utilize best practices of software development.**

At least use git to keep all configuration files.

Lessons Learned

✓**Cinder Storage is very sensitive to packet loss.**

Need to take actions to make network very stable.

✓**Extensive E2E testing for Control and Data plane is required.**

For example, if you have 300 compute hosts you need to test deploy VMs on every compute host and run a ping test with Jumbo Frames to verify both control and data plane.

There are open source initiatives to implement this functionality.

✓**Performance testing is needed**