CONTRAIL CLOUD SOLUTION WITH MPLS-OVER-UDP OVERLAYS

Sethuraman Ramanathan - System test

Executive summary:

MPLS-OVER-UDP Tunnels are used on datacenter environment as overlays. Existing technologies (like MPLS-OVER-GRE) to encapsulate Multi-Protocol Label Switching(MPLS) over IP are not adequate for efficient load balancing of MPLS application traffic, such as Layer3 Virtual Private Network (L3VPN) traffic across IP networks. This document specifies IP-based encapsulation technology, referred to as MPLS-in-User Datagram Protocol (UDP), which can facilitate the load balancing of MPLS application traffic across IP networks. This document also gives details on how to enable MPLS-OVER-UDP encapsulation when MX router interop with contrail controller.

Description:

Figure shows the frame formats of MPLS-OVER-GRE and MPLS-OVER-UDP.There no easy way to load share the traffic between 2 tunnel end points when MPLS-OVER-GRE encapsulation is used.The issue is GRE header is same for all the flows between the 2 Tunnel end points.
But when we use MPLS-OVER-UDP encapsulation the source port value(in the MPLS-OVER-UDP header) can be generated based on the certain fields in the customer packets(in our case based on the packets generated by VMs in the vrouter).With this the routers beween the Tunnel end points can load share the packets using hash of five-tuple of UDP packets.This is one of reason why MPLS-OVER-UDP overlays prefered over MPLS-OVER-GRE overlays.

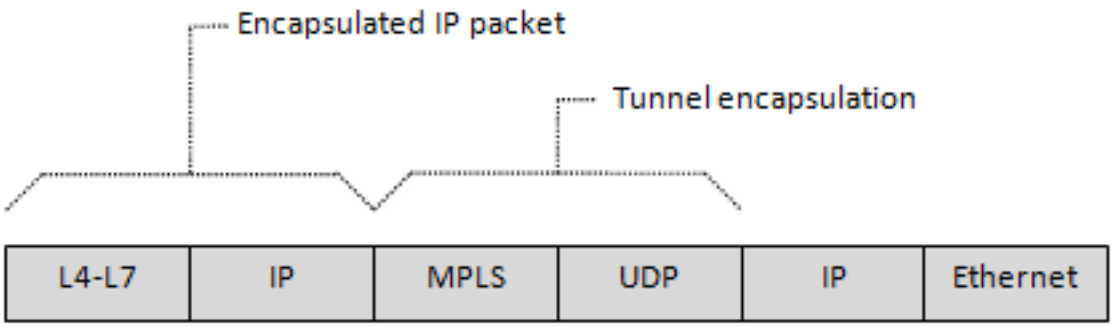


Figure 11: IP over MPLS over UDP Packet Format

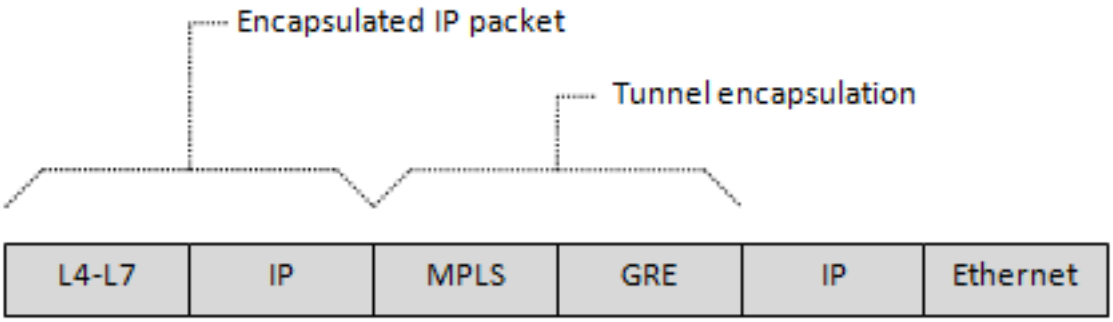
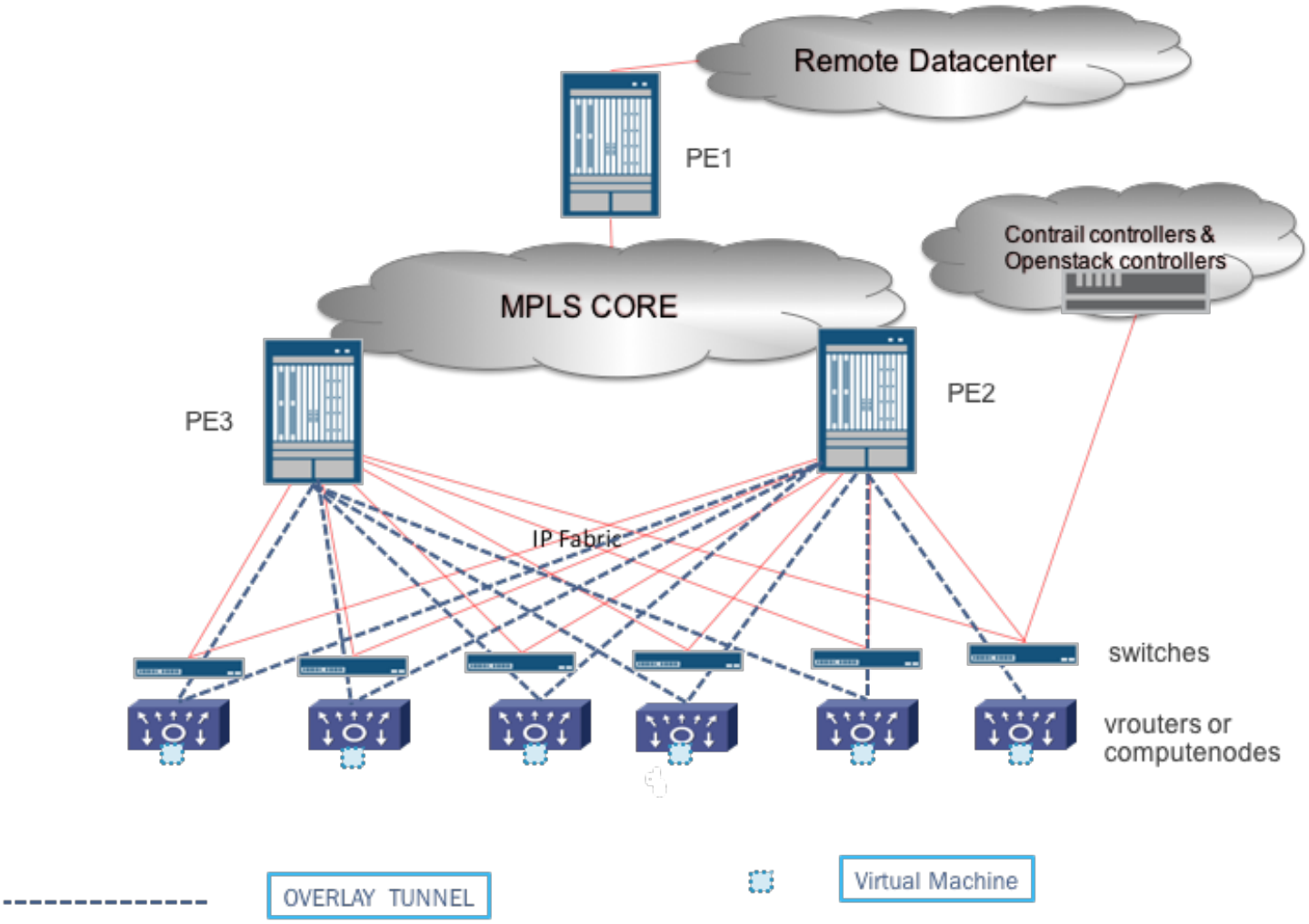


Figure 8: IP over MPLS over GRE Packet Format



The picture given above shows the implementation of Contrail cloud solution. In this topology PE2 & PE3 will be acting as local Datacenter gateways.PE1 will be the remote Datacenter gateway.Between PE2/PE3 and compute nodes (or vrouters) MPLS-OVER-UDP overlays will be implemented.So all traffic between compute

nodes to PE2/PE3 will be encapsulated with MPLS-OVER-UDP header.Contrail controllers will be managing the vrouters using XMPP.PE2 &PE3 will talk to Contrail controllers with BGP.The vrouters in contrail cloud will act similar to PE router in L3 VPN environment.When vrouter need to send traffic to vrouters in another datacenter it will encapulate packets with MPLS-OVER-UDP header and forward it to PE2 or PE3.PE2 or PE3 will decapsulate the MPLS-OVER-UDP header and then encapsulate it with MPLS headers and forward it to PE1.PE1 decapsulate MPLS header and forward the packets to the node in remote Datacenter.

When a VM in a vrouter need to send traffic to VM in another vrouter in the same datacenter,the source vrouter will encapsulate the packets with MPLS-OVER-UDP/MPLS-OVER-GRE header and then forward it to the IP fabric.After the destination vrouter receives the packet it will decapsulate the packets and forward it to the destination VM.

Configuration on MX gateway:

BGP Configuration :

BGP group contrail:

This section gives details on Bgp configuration of PE2 & PE3.Both the PEs will have similar configuration. Bgp group “contrail” (given below) peers to contrail controller. In this configuration ,Family route-target knob controls the advertisement of routes from PE2 & PE3 to contrail controller.It make sure that PE2 & PE3 advertise only required routes based on the route-targets received from the contrail controller.
Contrail controller advertise the route targets(which is applied to the virtual Networks created in contrail controller) to PE2 & PE3.PE2 & PE3 advertise routes with matching route targets. Unmatched routes will not be advertised. This avoids unnecessary routes advertised to contrail controller. The knob external-paths 5 is required when you have 3 controllers in a HA environment.

Before advertising any routes to contrail controller the next-hop of the route (through policy from-remote-pe-vrf1) is changed to PE2 or PE3 loopback address.This avoids the need for the controller to learn remote PE1 loopback address.

This policy also adds encapsulation extended community to the remote datacenter route before advertising it to the contrail controller.This community tells the vrouter(compute nodes) to use MPLS-OVER-UDP encapsulation before it forwards traffic for this route.If this community is not advertised then vrouter will use the default encapsulation (which is MPLS-OVER-GRE).

The encapsulation community is in the format of "members 0x030c:XXX:13". This identifies it as opaque extended community (type 0x03) of sub-type encapsulation (0x0c).Administrator field is set to 0 since it's reserved(But JUNOS do not allow 0 so you need to set it to some value recommended is AS NUMBER) and encap value is 13 (for MPLSoUDP).
The configuration given below is used in PE2 and PE3 routers.

```
{master}[edit]
regress@PE2# show protocols bgp group contrail
```

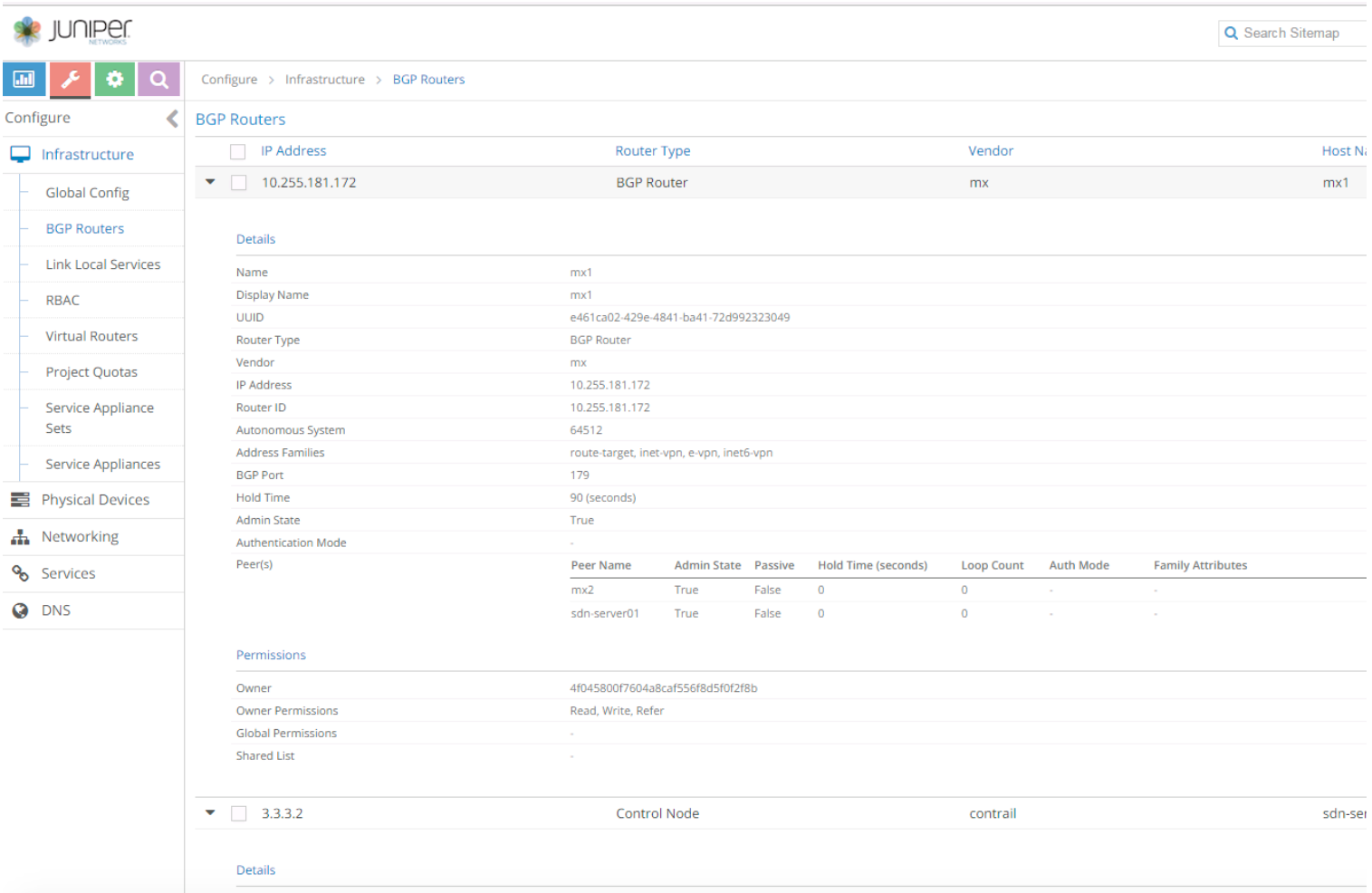
```
type internal;
local-address 10.255.181.172;
family inet-vpn {
   any;
}
family inet6-vpn {
   any;
}
family route-target {
   external-paths 5;
   advertise-default;
}
export from-remote-pe-vrf1;
vpn-apply-export;
cluster 2.2.2.2;
neighbor 3.3.3.2;

{master}[edit]
regress@PE2#

master}[edit]
regress@PE2# show policy-options policy-statement from-remote-pe-vrf1
term 1 {
   from {
        protocol bgp;
        route-filter 103.0.4.0/24 orlonger;
          }
   then {
      next-hop self;
      accept;
   }
}

{master}[edit]
regress@PE2# show policy-options community udp
members 0x030c:64512:13;

{master}[edit]
regress@PE2#
```

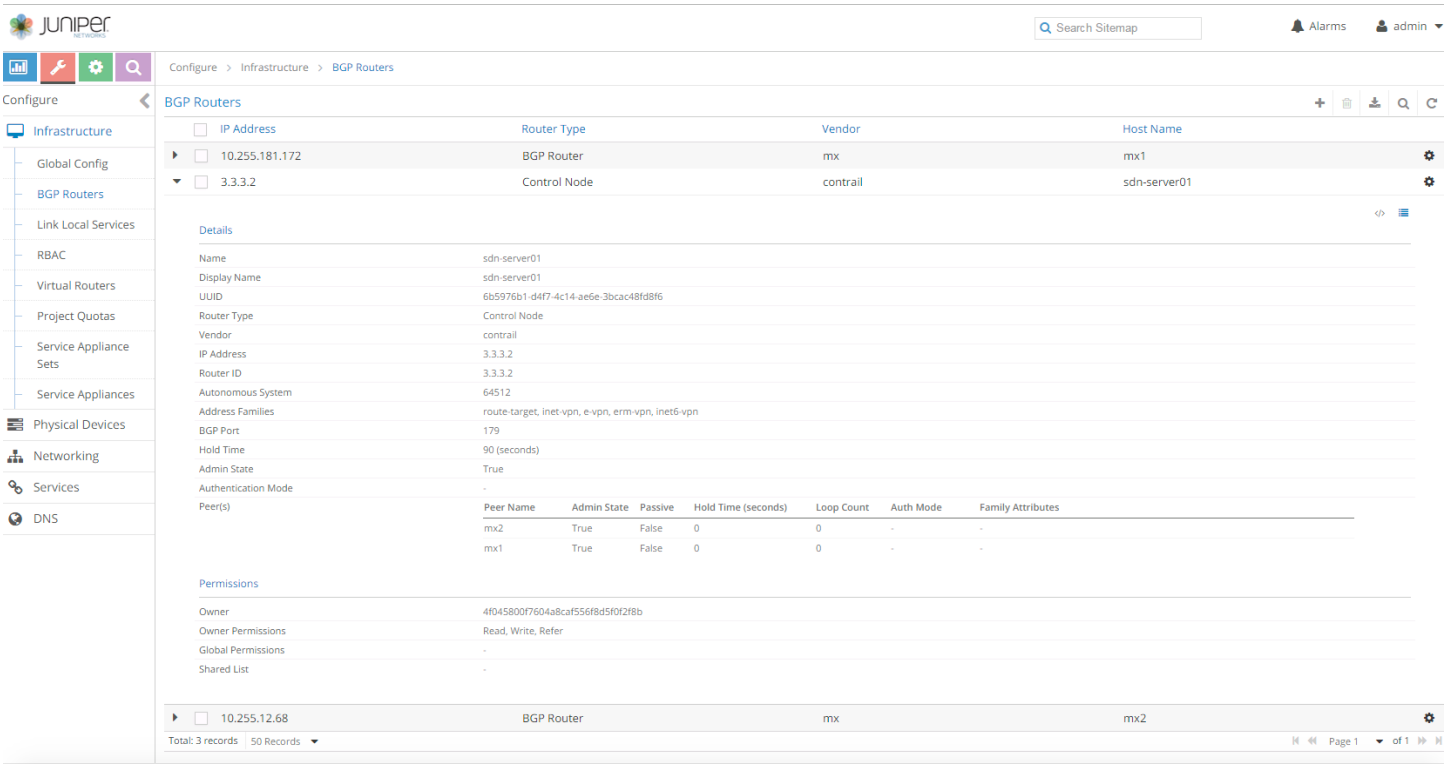
You can verify encapsulation status in vrouter(in compute node) as shown below.For this first find the tap interface.Then check the route using the vrf number.This output shows the type of encapsulation,source and destination IPs of the tunnel and other details.

```
root@vm6:~# vif --list
Vrouter Interface Table

Flags: P=Policy, X=Cross Connect, S=Service Chain, Mr=Receive Mirror
     Mt=Transmit Mirror, Tc=Transmit Checksum Offload, L3=Layer 3, L2=Layer 2
     D=DHCP, Vp=Vhost Physical, Pr=Promiscuous, Vnt=Native Vlan Tagged
     Mnp=No MAC Proxy, Dpdk=DPDK PMD Interface, Rfl=Receive Filtering Offload, Mon=Interface is Monitored
     Uuf=Unknown Unicast Flood, Vof=VLAN insert/strip offload

vif0/3     OS: tap6b9bb806-20
        Type:Virtual HWaddr:00:00:5e:00:01:00 IPaddr:0
```

```
        Vrf:1 Flags:PL3L2D MTU:9160 Ref:5
        RX packets:15533540  bytes:4717521394 errors:0
        TX packets:15659512  bytes:4721801850 errors:0


root@vm6:~# rt --dump 1 |grep 103.0.8.0/24
103.0.8.0/24         24         P        -          24       -
root@vm6:~# nh --get 24
Id:24        Type:Composite      Fmly: AF_INET  Rid:0  Ref_cnt:6        Vrf:1
           Flags:Valid, Policy, Ecmp,
           Sub NH(label): 16(17) 12(17)


Id:12        Type:Tunnel         Fmly: AF_INET  Rid:0  Ref_cnt:2        Vrf:0
           Flags:Valid, MPLSoUDP,
           Oif:0 Len:14 Flags Valid, MPLSoUDP,  Data:02 00 08 00 00 2b 52 54 00 23 13 39 08 00
           Vrf:0  Sip:60.60.0.6  Dip:10.255.181.172

root@vm6:~#
```

The picture given below shows the bgp configuration in Contrail controller.There are 2 nodes shown here.Router Type “BGP Router” shows the PE1 configuration details and Router Type “Control Node” details BGP configuration of Contrail Controller.





The output from MX router (given below) shows that bgp connection established between MX and Contrail Controller.

```
{master}[edit]
regress@PE2# run show bgp summary
Groups: 4 Peers: 5 Down peers: 0
Table          Tot Paths  Act Paths Suppressed    History Damp State    Pending
bgp.rtarget.0
                      17         13          0          0          0          0
bgp.l3vpn.0
                      58         54          0          0          0          0
bgp.l3vpn.2
                       0          0          0          0          0          0
bgp.l3vpn-inet6.0
                       0          0          0          0          0          0
bgp.l3vpn-inet6.2
                       0          0          0          0          0          0
Peer                     AS      InPkt     OutPkt    OutQ   Flaps Last Up/Dwn State|#Active/Received/Accepted/Damped...
3.3.3.2                64512         40         56       0       0       11:15 Establ
  bgp.rtarget.0: 13/17/17/0
  bgp.l3vpn.0: 4/4/4/0
  bgp.l3vpn-inet6.0: 0/0/0/0
  vrf1.inet.0: 4/4/4/0
```

BGP Group core:

This BGP Group “core” peers to the remote PE(PE1) connected to remote data center. The policy “change-next” will change the next hop of routes advertised to remote PE1.

```
{master}[edit]
regress@PE2# show protocols bgp group core
type internal;
local-address 10.255.181.172;
family inet-vpn {
    any;
}
family inet6-vpn {
    any;
}
export change-next;
vpn-apply-export;
neighbor 10.255.178.174;

{master}[edit]
regress@PE2#


{master}[edit]
regress@PE2# show policy-options policy-statement change-next
Dec 07 14:27:33
term 1 {
    from protocol bgp;
  then {
        next-hop self;
        accept;
    }
}

{master}[edit]
regress@PE2#
```

Dynamic tunnel configuration:

This section explains Dynamic Tunnel Configuration between MX to Contrail Controller and MX to vrouter.

Dynamic Tunnel to contrail controller:

PE2 & PE3 needs to have dynamic tunnel (MPLS-OVER-UDP/MPLS-OVER-GRE) created to the contrail controller. This configuration will create route to the controller ip address in inet.3 table.Without this route MX will not advertise bgp.l3vpn.0 table routes to controller(when you have family route-target enabled on PE2 & PE3).This Tunnel status can be verified as shown below.This tunnel will be in Dn state as there are no routes received from controller with protocol next-hop of contrail controller IP address(in this case it is 3.3.3.2).This is expected as contrail controller will not be advertising any route with 3.3.3.2 as protocol next hop.

```
{master}[edit]
regress@PE2# show routing-options dynamic-tunnels to-controller
source-address 10.255.181.172;
udp;
destination-networks {
   3.3.3.0/24;
}

{master}[edit]
regress@PE2#


{master}[edit]
regress@PE2# run show dynamic-tunnels database terse
Table: inet.3

Destination-network: 3.3.3.2/32
Destination        Source         Next-hop              Type       Status
3.3.3.2/32         10.255.181.172  0x487e67c nhid 0        udp        Dn    nexthop not installed

master}[edit]
regress@PE2# run show route 3.3.3.0
Dec 07 13:53:43

inet.0: 43 destinations, 44 routes (42 active, 0 holddown, 1 hidden)
+ = Active Route, - = Last Active, * = Both

3.3.3.0/24         *[OSPF/150] 00:40:14, metric 0, tag 0
                  > to 1.0.1.1 via ae0.0

inet.3: 18 destinations, 31 routes (18 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

3.3.3.0/24         *[Tunnel/300] 00:37:18
                    Tunnel

{master}[edit]
regress@PE2#
```

Dynamic Tunnels to vROUTERs:

PE2 & PE3 should create Dynamic TUNNELS to all the vROUTERs. In our case it is MPLS-OVER-UDP TUNNEL created between PE2 & PE3 to all vROUTERs.When PE2 or PE3 receives traffic from remote PE1 it uses this MPLS-OVER-UDP tunnel to forward the traffic to the respective vROUTER.The Tunnel status can be verified as given below.The Tunnel will come up only when a route received with the protocol nexthop in the segment 60.60.0.0/16(which is configured in destination-networks as shown below).In this case 60.60.0.0/16 is the IP segment used vrouters.

```
regress@PE2# show routing-options dynamic-tunnels
contrail-udp {
   source-address 10.255.181.172;
   udp;
   destination-networks {
     60.60.0.0/16;
   }
}
{master}[edit]
regress@PE2# run show dynamic-tunnels database
Table: inet.3
Destination-network: 60.60.0.6/32
Tunnel to: 60.60.0.6/32
  Reference count: 1
  Next-hop type: UDP
    Source address: 10.255.181.172 Tunnel Id: 1610612742
    Next hop: tunnel-composite, 0x4875634, nhid 710
      Reference count: 2
      State: Up


{master}[edit]
regress@PE2# run show dynamic-tunnels database terse
Table: inet.3

Destination-network: 60.60.0.6/32
Destination        Source         Next-hop              Type      Status
60.60.0.6/32       10.255.181.172  0x48793f4 nhid 741       udp       Up
```

This is the Routing instance configuration is created on PE2 & PE3 to install the routes to vrf table.This vrf configuration is required as it is a L3VPN scenario.

```
{master}[edit]
regress@PE2# show routing-instances vrf1
instance-type vrf;
interface lo0.1;
route-distinguisher 64512:1;
vrf-import test1-import;
vrf-export test1-export;
vrf-table-label;
}
```

```
{master}[edit]
regress@PE2# show policy-options policy-statement test1-export
term 1 {
   from protocol  direct ;
   then {
      community add testtarget1;
      accept;
   }
}

{master}[edit]
regress@PE2#

regress@PE2# show policy-options policy-statement test1-import
term 1 {
   from community testtarget1;
   then accept;
}

{master}[edit]
regress@PE2#
```

Remote datacenter PE configuration:

This configuration is applied on the remote PE router (in this case PE1).This is a simple L3VPN configuration.

```
[edit]
regress@PE1# show routing-instances vrfs1
Dec 06 13:21:03
instance-type vrf;
interface xe-1/1/1.1;
interface xe-1/1/1.2;
interface xe-1/1/1.3;
interface xe-1/1/1.4;
route-distinguisher 64512:1;
vrf-import test1-import;
vrf-export test1-export;
vrf-table-label;

[edit]
regress@PE1# show policy-options policy-statement test1-export
Dec 06 13:21:10
term 1 {
   from protocol direct;
   then {
      community add testtarget1;
      accept;
   }
}

[edit]
```

```
regress@PE1# show policy-options policy-statement test1-import
term 1 {
    from community testtarget1;
    then accept;
}

[edit]
regress@PE1#

[edit]
regress@PE1# show protocols bgp
precision-timers;
group contrail-1 {
    type internal;
    local-address 10.255.178.174;
    family inet {
        unicast;
    }
    family inet-vpn {
        any;
    }
    family inet6-vpn {
        any;
    }
    cluster 3.3.3.3;
    neighbor 10.255.181.172;

[edit]
regress@PE1#
```

Conclusion

In this document we have shown the use case of MPLS-OVER-UDP overlays. In addition to this configuration required to integrate MPLS-OVER-UDP overlays between MX and contrail controller is explained in this document.

APPENDIX A

The tunnel details from the FPC can be checked as shown below.The nexthop id can be used to check the tunnel details from the FPC.The Tunnel id,tunnel destination,Tunnel source in the FPC should match with the CLI.

```
{master}[edit]
regress@PE2# run show dynamic-tunnels database
Dec 06 14:06:45
Table: inet.3

Destination-network: 60.60.0.6/32
Tunnel to: 60.60.0.6/32
  Reference count: 1
  Next-hop type: UDP
    Source address: 10.255.181.172 Tunnel Id: 1610612742
    Next hop: tunnel-composite, 0x4875634, nhid 710
      Reference count: 2
      State: Up

regress@PE2# run request pfe execute target fpc3 command "show nhdb id 710 extensive"
Dec 07 13:58:26
================ fpc3 ================
SENT: Ukern command: show nhdb id 710 extensive

  ID     Type     Interface   Next Hop Addr    Protocol     Encap     MTU          Flags  PFE internal Flags
----- -------- ------------- --------------- ---------- ------------ ---- ------------------ ------------------
  710    Compst  -           -               MPLS         -    0  0x0000000000000000  0x0000000000000000

BFD Session Id: 0

Composite NH:
  Function: Tunnel Function
  Hardware Index: 0x0
  Composite flag: 0x0
  Composite pfe flag: 0xe
  Lower-level NH Ids:
  Derived NH Ids:
  Tunnel Data:
     Type     : UDP-V4
     Tunnel ID: 1610612742
     Encap VRF: 0
     Decap VRF: 0
     MTU      : 0
     Flags    : 0x2
     Encap Len: 28
```

```
    Encap    : 0x45 0x00 0x00 0x00 0x00 0x00 0x40 0x00
               0x40 0x2f 0x00 0x00 0xac 0xb5 0xff 0x0a
               0x06 0x00 0x3c 0x3c 0x00 0x00 0x00 0x00
               0x00 0x00 0x00 0x00
    Data Len : 8
    Data     : 0x3c 0x3c 0x00 0x06 0x0a 0xff 0xb5 0xac
     Feature List: NH
       [pfe-0]: 0x08ce6d4c00100000;
       [pfe-1]: 0x08c12e3000100000;
      f_mask:0xe000000000000000; c_mask:0xe000000000000000; f_num:3; c_num:3,  inst:0xffffffff
     Idx#0        -:
       [pfe-0]: 0x2bfffffd5e00a500
       [pfe-1]: 0x2bfffffd5e00a500

     Idx#1        -:
       [pfe-0]: 0x23fffffc0000000c
       [pfe-1]: 0x23fffffc0000000c

     Idx#2        -:
       [pfe-0]: 0x08c045d800080000
       [pfe-1]: 0x08c03d8800080000

Tunnel ID 1610612742
==============
      Ref-count 1
TunnelModel:
Dynamic Tunnel Model:
      Name = MPLSoUDP
      MTU = 0
      VRF = default.0(0)
      Source Entropy = 1
      Packets = 0 Bytes = 0

Source IP    : 10.255.181.172
Destination IP: 60.60.0.6


Ingress:
Index:0
  PFE(0): 0x2bfffffd5e006500
  PFE(1): 0x2bfffffd5e006500

Index:1
  PFE(0): 0x8ce6c8c00100000
  PFE(1): 0x8c12d7000100000

Handle JNH
0x8c045d800080000
0x8c03d8800080000
Egress:
Index:0
  PFE(0): 0x2bfffffd5e008500
  PFE(1): 0x2bfffffd5e008500

Index:1
  PFE(0): 0x23fffffc0000020a
```

```
  PFE(1): 0x23fffffc0000020a

Index:2
  PFE(0): 0x878b15400100000
  PFE(1): 0x878b2b000100000

Handle JNH
0x8ce6cec00100000
0x8c12dd000100000


  Routing-table id: 0
```

Full configuration:

PE2:

```
{master}[edit]
regress@PE2# show protocols ospf
area 0.0.0.0 {
    interface all {
        bfd-liveness-detection {
            minimum-interval 300;
            multiplier 3;
        }
    }
    interface fxp0.0 {
        disable;
    }
}

{master}[edit]
regress@PE2#

regress@PE2# show protocols mpls
ipv6-tunneling;
interface all;
{master}[edit]
regress@PE2#

egress@PE2# show protocols ldp;
interface ae0.0;
interface ae1.0;
interface ae2.0;
interface lo0.0;
{master}[edit]
regress@PE2#

{master}[edit]
regress@PE2# show protocols bgp group contrail
type internal;
local-address 10.255.181.172;
family inet-vpn {
    any;
```

```
}
family inet6-vpn {
   any;
}
family route-target {
 external-paths 5;
 advertise-default;
}
export from-remote-pe-vrf1;
cluster 2.2.2.2;
neighbor 3.3.3.2;
{master}[edit]
regress@PE2#

{master}[edit]
regress@PE2# show protocols bgp group core
type internal;
local-address 10.255.181.172;
family inet-vpn {
   any;
}
family inet6-vpn {
   any;
}
export change-next;
vpn-apply-export;
neighbor 10.255.178.174;
{master}[edit]
regress@PE2#

regress@leopard# show routing-options
Dec 13 12:19:18
ppm {
   redistribution-timer 120;
}
nonstop-routing;
autonomous-system 64512;
dynamic-tunnels {
   gre next-hop-based-tunnel;
   controller {
      source-address 10.255.181.172;
      udp;
      destination-networks {
         3.3.3.2/32;
      }
   }
contrail-udp {
      source-address 10.255.181.172;
      udp;
      destination-networks {
         60.60.0.0/16;
  }
}
}
```

```
regress@PE2# show policy-options policy-statement from-remote-pe-vrf1
term 1 {
   from {
        protocol bgp;
        route-filter 103.0.4.0/24 orlonger;
          }
   then {
      next-hop self;
      accept;
   }
}

{master}[edit]
regress@PE2# show policy-options community udp
members 0x030c:64512:13;

{master}[edit]
regress@PE2#
{master}[edit]
regress@PE2# show policy-options policy-statement change-next
Dec 07 14:27:33
term 1 {
   from protocol bgp;
  then {
      next-hop self;
      accept;
   }
}

{master}[edit]
regress@PE2#
```