**MPLS TE, MPLS DiffServ and DS-TE Design on IOS-XR**

**Business Drivers & Technical requirements**

To meet the business requirements to optimize bandwidth utilization and differentiate internal and external customers traffic flows across an aggregated transit environment such as MPLS Core backbone we design a MPLS TE solution with automated TE tunnels provisioning along with capability of MPLS TE to route traffic based on EXP value, DiffServ Traffic Engineering. The MPLS DiffServ Techniques such as MPLS VPN QoS also play a major role to remark customer traffic flows at ingress.

Below is outlined the different design approaches to select the proper TE solution for the customer (Keep in mind business drivers are factors of the business segment of the customer. Therefore the business requirements are unique per customer even though there are many similarities when it comes to technical requirements among distinct network environments.)

1. **MPLS TE and MPLS DiffServ:** 
   * Deployed where differentiation and optimization are required
   * ***Per-VRF Tunnel Selection*** *- With this technique we will provision manual TE tunnels with path protection and FRR with Link/node protection from PE to PE and forward high priority traffic (Trading VRFs) on these tunnels. Low priority traffic (Non-Trading VRFs) will flow following the Link-State Topology. This solution will require a QoS architecture describing admission control at the ingress and MPLS DiffServ through the MPLS Core. This increases the operational complexity of the network and reduce its scalability. We should probably look into a method to automate the provisioning of the TE tunnels.*
   * ***Auto-mesh and LDP tunneling*** *- LDP enabled on the links between PE and P routers, auto-mesh tunneling between P routers with LDP tunneling between PE routers. Admission Control and MPLS DiffServ will differentiate Customer traffic respectively at the ingress PE and on ingress P routers. We will also rely on CEF for traffic load balancing on the LDP links. This solution provides some limitations when it comes to Traffic Engineering Service differentiation as each VPN packet will have the same fate from TE forwarding perspective.*
2. **MPLS DiffServ and DS-TE** 
   * Deployed where strong differentiation and fine-grained optimization are required

***PBR (based on DSCP value)******+ AutoMesh + LDP tunneling (between PE nodes)***

* + - * This technique is the best fit to meet Customer' business requirements and capacity constraint where strong optimization and efficient bandwidth utilization are required. Referring to documentations related to TE technology auto-bw and auto-tunnel features are not interoperable. I guess manual provision of TE tunnels is required. Scalability and operational complexity are the two drawbacks I was able to outline with this technique, although further cons may be found with this setup. It might be worthy to know customer growth and expansion plan before deciding between full mesh and manual provisioning.
      * Business Drivers:
        + Cost efficiency
        + Elasticity
        + Business Continuity
      * Technical & Functional requirements:
        + Network consolidation & virtualization
        + Adaptable & Responsive Design
        + Resiliency/Reliability
      * In this design:
        + Auto-Mesh TE tunnel with PBTS between the LERs (PEs) along with LDP tunneling
        + Admission control at PE ingress using PBR based on DSCP value along with hop by hop QoS policy on each core link
        + Diagram:



* + - * After implement these functional requirements we’re able to achieve the following objectives :
        + Reduce convergence time with FRR and tuned backoff timers in IGP (ISIS)
        + Two label in the label-stack to deliver Customer flow end to end
      * *Drawback:* 
        + *No ECMP for IPv4 & VPNv4 traffic*
        + *VPN traffic is remarked to EXP0 as PBTS requires EXP value to steer remarked traffic with EXP bit into the corresponding forward class thus DS-TE tunnel. Therefore DSCP Mapping or Remarking is not backward-compatible with PBTS.*
        + *Since PBTS is compatible with EXP value then you must remark ingress traffic :*

*From PE which implies PE-CE link must carry the label therefore we must extend label forwarding to CE.*

* + - * Config
        + MPLS TE - PBR related Configuration

*!*

*ipv4 access-list AUTO\_TUNNEL\_PE\_MESH*

*10 permit ipv4 <subnet> <wilcard\_mask> any*

*!*

*rsvp*

*interface <interface>*

*bandwidth rdm bc0 9700000 bc1 4000000*

*!*

*!*

*mpls traffic-eng*

*interface <interface>*

*bfd fast-detect*

*auto-tunnel backup*

*!*

*!*

*bfd lsp head down-action reoptimize timeout 180*

*logging events all*

*logging events frr-protection*

*auto-tunnel mesh*

*group 100*

*attribute-set DIFFSRV\_PE\_TUNNEL\_MESH\_CLASS\_0*

*destination-list AUTO\_TUNNEL\_PE\_MESH*

*!*

*group 110*

*attribute-set DIFFSRV\_PE\_TUNNEL\_MESH\_CLASS\_1*

*destination-list AUTO\_TUNNEL\_PE\_MESH*

*!*

*group 120*

*attribute-set DIFFSRV\_PE\_TUNNEL\_MESH\_CLASS\_2*

*destination-list AUTO\_TUNNEL\_PE\_MESH*

*!*

*group 130*

*attribute-set DIFFSRV\_PE\_TUNNEL\_MESH\_CLASS\_3*

*destination-list AUTO\_TUNNEL\_PE\_MESH*

*!*

*group 140*

*attribute-set DIFFSRV\_PE\_TUNNEL\_MESH\_CLASS\_4*

*destination-list AUTO\_TUNNEL\_PE\_MESH*

*!*

*group 150*

*attribute-set DIFFSRV\_PE\_TUNNEL\_MESH\_CLASS\_5*

*destination-list AUTO\_TUNNEL\_PE\_MESH*

*!*

*tunnel-id min 100 max 999*

*!*

*auto-tunnel backup*

*timers removal unused 20*

*tunnel-id min 18000 max 19023*

*!*

*reoptimize 3600*

*fast-reroute timers promotion 0*

*attribute-set auto-mesh DIFFSRV\_PE\_TUNNEL\_MESH\_CLASS\_0*

*signalled-bandwidth 100 class-type 0*

*autoroute announce*

*fast-reroute*

*record-route*

*!*

*attribute-set auto-mesh DIFFSRV\_PE\_TUNNEL\_MESH\_CLASS\_1*

*priority 6 6*

*signalled-bandwidth 100 class-type 0*

*autoroute announce*

*fast-reroute*

*record-route*

*forward-class 1*

*!*

*attribute-set auto-mesh DIFFSRV\_PE\_TUNNEL\_MESH\_CLASS\_2*

*priority 4 4*

*signalled-bandwidth 100 class-type 0*

*autoroute announce*

*fast-reroute*

*record-route*

*forward-class 2*

*!*

*attribute-set auto-mesh DIFFSRV\_PE\_TUNNEL\_MESH\_CLASS\_3*

*priority 3 3*

*signalled-bandwidth 100 class-type 0*

*autoroute announce*

*fast-reroute*

*record-route*

*forward-class 3*

*!*

*attribute-set auto-mesh DIFFSRV\_PE\_TUNNEL\_MESH\_CLASS\_4*

*priority 5 5*

*signalled-bandwidth 1000 class-type 0*

*autoroute announce*

*fast-reroute*

*record-route*

*forward-class 4*

*!*

*attribute-set auto-mesh DIFFSRV\_PE\_TUNNEL\_MESH\_CLASS\_5*

*priority 1 1*

*signalled-bandwidth 2000 class-type 1*

*autoroute announce*

*fast-reroute*

*record-route*

*forward-class 5*

*!*

*auto-bw collect frequency 5*

*reoptimize timers delay cleanup 10*

*reoptimize timers delay installation 10*

*reoptimize timers delay after-frr 5*

*ds-te mode ietf*

*ds-te te-classes*

*te-class 0 class-type 0 priority 7*

*te-class 1 class-type 0 priority 6*

*te-class 2 class-type 0 priority 5*

*te-class 3 class-type 0 priority 2*

*te-class 4 class-type 1 priority 0*

*te-class 5 class-type 0 priority 1*

*te-class 6 unused*

*te-class 7 unused*

*!*

*bfd minimum-interval 100*

*bfd multiplier 3*

*!*

*mpls ldp*

*log*

*neighbor*

*nsr*

*graceful-restart*

*!*

*igp sync delay on-session-up 5*

*router-id <router-id>*

*address-family ipv4*

*traffic-eng*

*auto-tunnel mesh*

*group all*

*!*

*!*

*!*

!

* + - Outputs results

*lax-n7k-ce1# traceroute 172.16.10.2 source 172.20.10.2 vrf FINANCIAL*

*traceroute to 172.16.10.2 (172.16.10.2) from 172.20.10.2 (172.20.10.2), 30 hops max, 40 byte packets*

*1 172.20.192.25 (172.20.192.25) 1.019 ms 0.818 ms 1.042 ms*

*2 10.72.11.41 (10.72.11.41) 1.66 ms 1.54 ms 1.651 ms*

*[Label=24171 E=0 TTL=1 S=0, Label=24010 E=0 TTL=1 S=1]*

*3 10.72.11.14 (10.72.11.14) 1.684 ms 1.541 ms 1.618 ms*

*[Label=24115 E=0 TTL=1 S=0, Label=24010 E=0 TTL=2 S=1]*

*4 10.72.11.138 (10.72.11.138) 1.736 ms 1.573 ms 1.522 ms*

*[Label=24010 E=0 TTL=1 S=1]*

*5 172.16.192.30 (172.16.192.30) 2.014 ms 1.75 ms 1.765 ms*

*6 172.16.10.2 (172.16.10.2) 2.037 ms 1.981 ms 2.001 ms*

*nyc-n7k-ce1# traceroute 172.20.10.2 source 172.16.10.2 vrf FINANCIAL*

*traceroute to 172.20.10.2 (172.20.10.2) from 172.16.10.2 (172.16.10.2), 30 hops max, 40 byte packets*

*1 172.16.192.89 (172.16.192.89) 1.289 ms 0.896 ms 0.929 ms*

*2 10.72.11.109 (10.72.11.109) 1.622 ms 1.545 ms 1.58 ms*

*[Label=24134 E=0 TTL=1 S=0, Label=24008 E=0 TTL=1 S=1]*

*3 10.72.11.17 (10.72.11.17) 1.714 ms 1.63 ms 1.599 ms*

*[Label=24165 E=0 TTL=1 S=0, Label=24008 E=0 TTL=2 S=1]*

*4 10.72.11.58 (10.72.11.58) 1.727 ms 1.614 ms 1.592 ms*

*[Label=24008 E=0 TTL=1 S=1]*

*5 172.20.10.2 (172.20.10.2) 1.838 ms 1.841 ms 172.20.192.30 (172.20.192.30) 1.894 ms*

* + ***PBR (based on EXP bit value)*** *+* ***AutoMesh + LDP tunneling & LDP***
    - This technique is the best fit to meet Customer' business requirements and capacity constraint where strong differentiation and fine-grained optimization are required. LDP enabled on the links between PE and P routers, auto-mesh tunneling between P routers with LDP tunneling between PE routers. Admission Control and MPLS DiffServ will differentiate Customer traffic respectively at the ingress and on the uplinks towards the P routers. We will also rely on CEF for traffic load balancing on the LDP links.
    - Business Drivers:
      * Cost efficiency
      * Elasticity
      * Business Continuity
    - Technical & Functional requirements:
      * Network consolidation & virtualization
      * Adaptable & Responsive Design
      * Resiliency/Reliability
    - In this setup:
      * Auto-Mesh TE tunnel with PBTS between the LSRs along with LDP tunneling
      * LDP on the links between the LER and LSR
      * Admission control at PE ingress using Short-Pipe model and PBR at P ingress along with hop by hop QoS policy on each core link
      * Diagram updated:



* 1. After implement these functional requirements we’re able to achieve the following goals:
     + ECMP for IPv4 & VPNv4 traffic
     + Reduce convergence time with FRR and tuned backoff timers in IGP (ISIS)
     + Enforce the EXP value of VPN packet across the MPLS Core
     + Scale MPLS –TE topology for future expansion

* + Knob:
    - To make this works you have to specifically add each TE RID in the access-list used to automate the TE Tunnel creation. The odd behavior is if the PE RIDs are also included in the access-list P nodes will attempt to build LSP towards all RID even though the attaching interfaces are not part of MPLS Traffic-Eng Process thus these TE tunnels will stay down. Since we’re using DS-TE for class-based forwarding & constrained-based routing each TE tunnel is tied to a unique EXP value therefore if the next-hop (using auto-route) is not installed in the RIB table packets will get dropped.

So instead of using this line:

*!*

*ipv4 access-list AUTO\_TUNNEL\_P\_MESH*

*10 permit ipv4 10.72.2.0 0.0.0.255 any*

*!*

You have to change it to :

*!*

*ipv4 access-list AUTO\_TUNNEL\_P\_MESH*

*10 permit ipv4 host 10.72.2.1 any*

*20 permit ipv4 host 10.72.2.2 any*

*30 permit ipv4 host 10.72.2.3 any*

*40 permit ipv4 host 10.72.2.4 any*

*!*

* + *Drawback:* 
    - LDP sessions required between TE nodes to prevent packet drops. As per PHP mechanism LDP label is swapped on ingress P and the remote P will push the corresponding LDP label. Without LDP session the remote P node will receive a packet with an unknown BGP label thus drop the packet or blackhole the traffic.
    - *Since PBTS is compatible with EXP value then you must remark ingress traffic :*
      * *From P which means you must remark customer traffic at ingress PE then run admission control for PBTS on each ingress P*
  1. Config
     + MPLS TE - PBR Configuration

*!*

*ipv4 access-list AUTO\_TUNNEL\_P\_MESH*

*10 permit ipv4 host <RID\_P\_1\_NODE> any*

*20 permit ipv4 host <RID\_P\_2\_NODE> any*

*. . .*

*!*

*rsvp*

*interface <interface\_connected\_to\_p\_node>*

*bandwidth rdm bc0 9700000 bc1 4000000*

*!*

*!*

*mpls traffic-eng*

*interface <interface\_connected\_to\_p\_node>*

*bfd fast-detect*

*auto-tunnel backup*

*!*

*!*

*bfd lsp head down-action reoptimize timeout 180*

*logging events all*

*logging events frr-protection*

*auto-tunnel mesh*

*group 100*

*attribute-set DIFFSRV\_P\_TUNNEL\_MESH\_CLASS\_0*

*destination-list AUTO\_TUNNEL\_P\_MESH*

*!*

*group 110*

*attribute-set DIFFSRV\_P\_TUNNEL\_MESH\_CLASS\_1*

*destination-list AUTO\_TUNNEL\_P\_MESH*

*!*

*group 120*

*attribute-set DIFFSRV\_P\_TUNNEL\_MESH\_CLASS\_2*

*destination-list AUTO\_TUNNEL\_P\_MESH*

*!*

*group 130*

*attribute-set DIFFSRV\_P\_TUNNEL\_MESH\_CLASS\_3*

*destination-list AUTO\_TUNNEL\_P\_MESH*

*!*

*group 140*

*attribute-set DIFFSRV\_P\_TUNNEL\_MESH\_CLASS\_4*

*destination-list AUTO\_TUNNEL\_P\_MESH*

*!*

*group 150*

*attribute-set DIFFSRV\_P\_TUNNEL\_MESH\_CLASS\_5*

*destination-list AUTO\_TUNNEL\_P\_MESH*

*!*

*tunnel-id min 100 max 999*

*!*

*auto-tunnel backup*

*timers removal unused 20*

*tunnel-id min 18000 max 19023*

*!*

*reoptimize 3600*

*fast-reroute timers promotion 0*

*attribute-set auto-mesh DIFFSRV\_P\_TUNNEL\_MESH\_CLASS\_0*

*signalled-bandwidth 100 class-type 0*

*autoroute announce*

*fast-reroute*

*record-route*

*!*

*attribute-set auto-mesh DIFFSRV\_P\_TUNNEL\_MESH\_CLASS\_1*

*priority 6 6*

*signalled-bandwidth 100 class-type 0*

*autoroute announce*

*fast-reroute*

*record-route*

*forward-class 1*

*!*

*attribute-set auto-mesh DIFFSRV\_P\_TUNNEL\_MESH\_CLASS\_2*

*priority 4 4*

*signalled-bandwidth 100 class-type 0*

*autoroute announce*

*fast-reroute*

*record-route*

*forward-class 2*

*!*

*attribute-set auto-mesh DIFFSRV\_P\_TUNNEL\_MESH\_CLASS\_3*

*priority 3 3*

*signalled-bandwidth 100 class-type 0*

*autoroute announce*

*fast-reroute*

*record-route*

*forward-class 3*

*!*

*attribute-set auto-mesh DIFFSRV\_P\_TUNNEL\_MESH\_CLASS\_4*

*priority 5 5*

*signalled-bandwidth 1000 class-type 0*

*autoroute announce*

*fast-reroute*

*record-route*

*forward-class 4*

*!*

*attribute-set auto-mesh DIFFSRV\_P\_TUNNEL\_MESH\_CLASS\_5*

*priority 1 1*

*signalled-bandwidth 2000 class-type 1*

*autoroute announce*

*fast-reroute*

*record-route*

*forward-class 5*

*!*

*auto-bw collect frequency 5*

*reoptimize timers delay cleanup 10*

*reoptimize timers delay installation 10*

*reoptimize timers delay after-frr 5*

*ds-te mode ietf*

*ds-te te-classes*

*te-class 0 class-type 0 priority 7*

*te-class 1 class-type 0 priority 6*

*te-class 2 class-type 0 priority 5*

*te-class 3 class-type 0 priority 2*

*te-class 4 class-type 1 priority 0*

*te-class 5 class-type 0 priority 1*

*te-class 6 unused*

*te-class 7 unused*

*!*

*bfd minimum-interval 100*

*bfd multiplier 3*

*!*

*mpls ldp*

*address-family ipv4*

*traffic-eng*

*auto-tunnel mesh*

*group all*

*interface <interface\_toward\_pe\_node>*

*!*

*!*

*commit*

*!*

*end*

*!*

* + Outputs results

*lax-n7k-ce1# traceroute 172.17.222.101 source 172.23.222.101 vrf FINANCIAL*

*traceroute to 172.17.222.101 (172.17.222.101) from 172.23.222.101 (172.23.222.101), 30 hops max, 40 byte packets*

*1 172.20.192.25 (172.20.192.25) 1.163 ms 1.068 ms 0.88 ms*

*2 10.72.11.57 (10.72.11.57) 1.665 ms 10.72.11.41 (10.72.11.41) 1.872 ms 10.72.11.57 (10.72.11.57) 1.666 ms*

*[Label=24249 E=4 TTL=1 S=0, Label=24038 E=4 TTL=1 S=1]*

*[Label=24134 E=4 TTL=1 S=0, Label=24038 E=4 TTL=1 S=1]*

*[Label=24249 E=4 TTL=1 S=0, Label=24038 E=4 TTL=1 S=1]*

*3 10.72.11.10 (10.72.11.10) 1.857 ms 1.737 ms 10.72.11.18 (10.72.11.18) 1.722 ms*

*[Label=24193 E=4 TTL=1 S=0, Label=24038 E=4 TTL=2 S=1]*

*[Label=24193 E=4 TTL=1 S=0, Label=24038 E=4 TTL=2 S=1]*

*[Label=24193 E=4 TTL=1 S=0, Label=24038 E=4 TTL=2 S=1]*

*4 10.72.11.106 (10.72.11.106) 1.82 ms 10.72.11.138 (10.72.11.138) 1.94 ms 10.72.11.106 (10.72.11.106) 1.733 ms*

*[Label=24038 E=4 TTL=1 S=1]*

*[Label=24038 E=4 TTL=1 S=1]*

*[Label=24038 E=4 TTL=1 S=1]*

*5 172.17.222.101 (172.17.222.101) 2.186 ms 1.98 ms 2.099 ms*

*nyc-n7k-ce1# traceroute 172.23.222.101 source 172.17.222.101 vrf FINANCIAL*

*traceroute to 172.23.222.101 (172.23.222.101) from 172.17.222.101 (172.17.222.101), 30 hops max, 40 byte packets*

*1 172.16.192.89 (172.16.192.89) 1.336 ms 0.992 ms 0.889 ms*

*2 10.72.11.109 (10.72.11.109) 1.816 ms 10.72.11.141 (10.72.11.141) 1.86 ms 10.72.11.109 (10.72.11.109) 1.616 ms*

*[Label=24207 E=4 TTL=1 S=0, Label=24036 E=4 TTL=1 S=1]*

*[Label=24298 E=4 TTL=1 S=0, Label=24036 E=4 TTL=1 S=1]*

*[Label=24207 E=4 TTL=1 S=0, Label=24036 E=4 TTL=1 S=1]*

*3 10.72.11.21 (10.72.11.21) 2.22 ms 1.612 ms 10.72.11.17 (10.72.11.17) 1.783 ms*

*[Label=24260 E=4 TTL=1 S=0, Label=24036 E=4 TTL=2 S=1]*

*[Label=24260 E=4 TTL=1 S=0, Label=24036 E=4 TTL=2 S=1]*

*[Label=24260 E=4 TTL=1 S=0, Label=24036 E=4 TTL=2 S=1]*

*4 10.72.11.58 (10.72.11.58) 2.153 ms 10.72.11.42 (10.72.11.42) 1.774 ms 10.72.11.58 (10.72.11.58) 1.614 ms*

*[Label=24036 E=4 TTL=1 S=1]*

*[Label=24036 E=4 TTL=1 S=1]*

*[Label=24036 E=4 TTL=1 S=1]*

*5 172.23.222.101 (172.23.222.101) 1.822 ms 1.763 ms 1.747 ms*

* 1. This design as it solves the scalability pain points of MPLS TE, reduce the control plane stress in the core and simplifies the operational expenditure of this mpls core.