Configure Multiple Transports and Traffic Engineering with Centralized Control Policy and App Route Policy

Contents

Introduction

This document describes how to configure centralized control policy and app route policy to achieve traffic engineering between sites. It might be considered as a specific design guideline for the particular use case as well.

Prerequisites

Requirements

There are no specific requirements for this document.

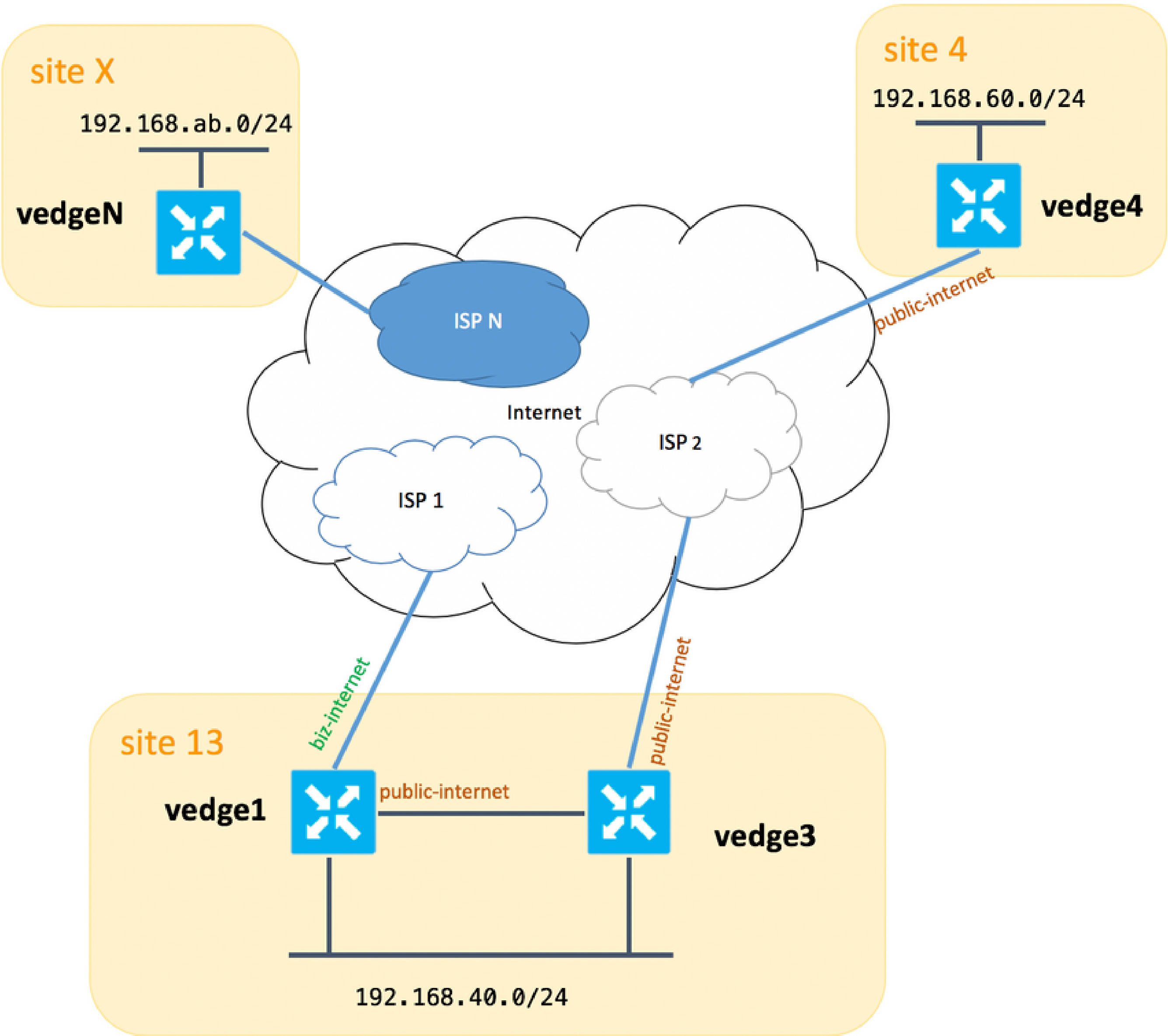
Components Used

This document is not restricted to specific software and hardware versions.

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, ensure that you understand the potential impact of any command.

Configuration

For the purpose of demonstration and a better understanding of the problem described later, please consider the topology shown in this image.



Please note, that in general between **vedge1** and **vedge3** you should have second link/subinterface for **biz-internet** TLOC extension as well, but here for sake of simplicity it was not configured.

Here are corresponding system settings for vEdges/vSmart (vedge2 represent all other sites):

hostname	site-id	system-ip
vedge1	13	192.168.30.4
vedge3	13	192.168.30.6
vedge4	4	192.168.30.7
vedgex	X	192.168.30.5
vsmart1	1	192.168.30.3

Here you can find transport side configurations for reference.

vedge1:

```
vedge1# show running-config vpn 0
vpn 0
 interface ge0/0
```

```
  description "ISP_1"
  ip address 192.168.109.4/24
  nat
   respond-to-ping
  !
  tunnel-interface
   encapsulation ipsec
   color biz-internet
   no allow-service bgp
   allow-service dhcp
   allow-service dns
   allow-service icmp
   allow-service sshd
   no allow-service netconf
   no allow-service ntp
   no allow-service ospf
   allow-service stun
  !
  no shutdown
 !
 interface ge0/3
  description "TLOC-extension via vedge3 to ISP_2"
  ip address 192.168.80.4/24
  tunnel-interface
   encapsulation ipsec
   color public-internet
   no allow-service bgp
   allow-service dhcp
   allow-service dns
   allow-service icmp
   no allow-service sshd
   no allow-service netconf
   no allow-service ntp
   no allow-service ospf
   allow-service stun
  !
  no shutdown
 !
 !
 ip route 0.0.0.0/0 192.168.80.6
 ip route 0.0.0.0/0 192.168.109.10
!
```

vedge3:

```
vpn 0
 interface ge0/0
  description "ISP_2"
  ip address 192.168.110.6/24
  nat
   respond-to-ping
  !
  tunnel-interface
   encapsulation ipsec
   color public-internet
   carrier carrier3
   no allow-service bgp
   allow-service dhcp
   allow-service dns
   allow-service icmp
   no allow-service sshd
   no allow-service netconf
   no allow-service ntp
```

```
   no allow-service ospf
   no allow-service stun
  !
  no shutdown
 !
 interface ge0/3
  ip address 192.168.80.6/24
  tloc-extension ge0/0
  no shutdown
 !
 ip route 0.0.0.0/0 192.168.110.10
```

vedge4:

```
vpn 0
 interface ge0/1
  ip address 192.168.103.7/24
  tunnel-interface
   encapsulation ipsec
   color public-internet
   no allow-service bgp
   allow-service dhcp
   allow-service dns
   allow-service icmp
   no allow-service sshd
   no allow-service netconf
   no allow-service ntp
   allow-service ospf
   no allow-service stun
  !
  no shutdown
 !
 ip route 0.0.0.0/0 192.168.103.10
!
```

Problem

The user wants to achieve these goals:

Internet service provides **ISP 2** should be preferred to communicate between **site 13** and **site 4** because of some reasons. For example, It's quite a common use case and a scenario when connection/connectivity quality within an ISP between its own clients is very good, but toward the rest of the Internet connectivity quality does not meet company's SLA because of some troubles or congestion on an ISP uplink and and hence this ISP (**ISP 2** in our case) should be avoided in general.

Site **site 13** should prefer **public-internet** uplink to connect to site **site 4**, but still, maintain redundancy and should be able to reach **site 4** if **public-internet** fails.

Site **site 4** should still maintain best-effort connectivity with all other sites directly (hence you can't use **restrict** keyword here on **vedge4** to achieve that goal).

Site **site 13** should use the better quality link with **biz-internet** colorto reach all other sites (represented by **site X** on topology diagram).

Another reason might be cost/pricing issues when traffic within ISP is free of charge, but much more expensive when traffic exiting a provider network (autonomous system).

Some users who are not experienced with SD-WAN approach and get used to **classic** routing may start to configure static routing to force traffic from **vedge1** to **vedge4** public interface address via TLOC-extension interface between **vedge1** and **vedge3**, but it won't give the desired result and can create confusion because:

Management plane traffic (e.g. ping, traceroute utility packet) follows the desired route.

At the same time, SD-WAN data plane tunnels (IPsec or gre transport tunnels) ignores routing table information and form connections based on TLOCs **colors.**

Since a static route has no intelligence, if **public-internet** TLOC is down on **vedge3** (uplink to ISP 2),then **vedge1** won't notice this and connectivity to **vedge4** fails despite the fact that **vedge1** still has **biz-internet** available.

Hence this approach should be avoided and not usable.

Solution

1. Use of centralized control policy to set a preference for **public-internet** TLOC on the vSmart controller when announcing corresponding OMP routes to **vedge4.** It helps to archive the desired traffic path from **site 4**to **site 13**.

2. To achieve desired traffic path in reverse direction from **site 13** to **site 4** you can't use centralized control policy because **vedge4** has only one TLOC available, hence you can't set a preference to anything, but you can use app route policy to achieve this result for egress traffic from **site 13**.

Here is how centralized control policy may look like on vSmart controller to prefer **public-internet** TLOC to reach **site 13**:

```
policy
 control-policy S4_S13_via_PUB
  sequence 10
   match tloc
    color   public-internet
    site-id 13
   !
   action accept
    set
     preference 333
    !
   !
  !
  default-action accept
 !
!
```

And here is an example of app route policy to prefer **public-internet** uplink as an exit point for egress traffic from **site 13** to **site 4** :

```
policy
 app-route-policy S13_S4_via_PUB
  vpn-list CORP_VPNs
```

```
   sequence 10
    match
     destination-data-prefix-list SITE4_PREFIX
    !
    action
     count                      COUNT_PKT
     sla-class SLA_CL1 preferred-color public-internet
    !
   !
  !
 !
policy
 lists
  site-list S13
   site-id 13
  !
  site-list S40
   site-id 4
  !
  data-prefix-list SITE4_PREFIX
   ip-prefix 192.168.60.0/24
  !
  vpn-list CORP_VPNs
   vpn 40
  !
 !
 sla-class SLA_CL1
  loss    1
  latency 100
  jitter  100
 !
```

Policies should be applied appropriately on vSmart controller:

```
apply-policy
 site-list S13
  app-route-policy S13_S4_via_PUB
 !
 site-list S4
  control-policy S4_S13_via_PUB out
 !
!
```

Please remember also that app-route policies can't be configured as a localized policy and should be applied on vSmart only.

Verify

Please note app route policy won't be applied to vEdge locally generated traffic, hence to verify if traffic flows steered according to the desired path it's recommended to generate some traffic from LAN segments of corresponding sites. As a high-level testing scenario case you can use iperf to generate traffic between hosts in LAN segments of **site 13** and **site 4** and then check an interface statistics. For example, in my case, there was no traffic besides system generated and hence you can see that major amount of traffic passed through ge0/3 interface towards TLOC extension on **vedge3**:

```
vedge1# show interface statistics
```

```
PPPOE  PPPOE  DOT1X  DOT1X
                 AF    RX                    RX      RX     TX                     TX      TX
RX     RX     TX     TX     TX     RX     TX     RX
VPN  INTERFACE    TYPE  PACKETS  RX OCTETS   ERRORS  DROPS  PACKETS  TX OCTETS   ERRORS  DROPS
PPS    Kbps    PPS    Kbps    PKTS   PKTS   PKTS   PKTS
------------------------------------------------------------------------------------------------
----------------------------------------------------------
0    ge0/0        ipv4  1832     394791      0       167    1934     894680      0       0
26     49     40     229     -      -      0      0
0    ge0/2        ipv4  0        0           0       0      0        0           0       0
0      0      0      0       -      -      0      0
0    ge0/3        ipv4  3053034  4131607715  0       27     2486248  3239661783  0       0
51933  563383  41588  432832  -      -      0      0
0    ge0/4        ipv4  0        0           0       0      0        0           0       0
0      0      0      0       -      -      0      0
```

Troubleshoot

First of all, ensure that corresponding BFD sessions are established (do not use **restrict** keyword anywhere):

```
vedge1# show interface statistics


PPPOE  PPPOE  DOT1X  DOT1X
                 AF    RX                    RX      RX     TX                     TX      TX
RX     RX     TX     TX     TX     RX     TX     RX
VPN  INTERFACE    TYPE  PACKETS  RX OCTETS   ERRORS  DROPS  PACKETS  TX OCTETS   ERRORS  DROPS
PPS    Kbps    PPS    Kbps    PKTS   PKTS   PKTS   PKTS
------------------------------------------------------------------------------------------------
----------------------------------------------------------
0    ge0/0        ipv4  1832     394791      0       167    1934     894680      0       0
26     49     40     229     -      -      0      0
0    ge0/2        ipv4  0        0           0       0      0        0           0       0
0      0      0      0       -      -      0      0
0    ge0/3        ipv4  3053034  4131607715  0       27     2486248  3239661783  0       0
51933  563383  41588  432832  -      -      0      0
0    ge0/4        ipv4  0        0           0       0      0        0           0       0
0      0      0      0       -      -      0      0

vedge1# show interface statistics


PPPOE  PPPOE  DOT1X  DOT1X
                 AF    RX                    RX      RX     TX                     TX      TX
RX     RX     TX     TX     TX     RX     TX     RX
VPN  INTERFACE    TYPE  PACKETS  RX OCTETS   ERRORS  DROPS  PACKETS  TX OCTETS   ERRORS  DROPS
PPS    Kbps    PPS    Kbps    PKTS   PKTS   PKTS   PKTS
------------------------------------------------------------------------------------------------
----------------------------------------------------------
0    ge0/0        ipv4  1832     394791      0       167    1934     894680      0       0
26     49     40     229     -      -      0      0
0    ge0/2        ipv4  0        0           0       0      0        0           0       0
0      0      0      0       -      -      0      0
0    ge0/3        ipv4  3053034  4131607715  0       27     2486248  3239661783  0       0
51933  563383  41588  432832  -      -      0      0
0    ge0/4        ipv4  0        0           0       0      0        0           0       0
0      0      0      0       -      -      0      0

vedge1# show interface statistics
```

```
PPPOE  PPPOE  DOT1X  DOT1X
                  AF    RX                  RX      RX      TX                    TX      TX
RX     RX      TX      TX      TX      RX      TX      RX
VPN  INTERFACE    TYPE  PACKETS  RX OCTETS   ERRORS  DROPS  PACKETS  TX OCTETS   ERRORS  DROPS
PPS    Kbps    PPS     Kbps     PKTS    PKTS    PKTS    PKTS
-------------------------------------------------------------------------------------------------
-----------------------------------------------------------
0    ge0/0        ipv4  1832     394791      0       167    1934     894680      0       0
26     49      40     229     -       -       0       0
0    ge0/2        ipv4  0        0           0       0      0        0           0       0
0      0       0      0       -       -       0       0
0    ge0/3        ipv4  3053034  4131607715  0       27     2486248  3239661783  0       0
51933  563383  41588  432832  -       -       0       0
0    ge0/4        ipv4  0        0           0       0      0        0           0       0
0      0       0      0       -       -       0       0
```

If you can't achieve the desired result with traffic engineering, then check that policies were applied properly:

1. On **vedge4** you should check that for prefixes originated from **site 13** appropriate TLOC was selected:

```
vedge4# show omp routes 192.168.40.0/24 detail

---------------------------------------------------
omp route entries for vpn 40 route 192.168.40.0/24
---------------------------------------------------
            RECEIVED FROM:
peer            192.168.30.3
path-id         72
label           1002
status          R
loss-reason     tloc-preference
lost-to-peer    192.168.30.3
lost-to-path-id 74
    Attributes:
     originator       192.168.30.4
     type             installed
     tloc             192.168.30.4, biz-internet, ipsec
     ultimate-tloc    not set
     domain-id        not set
     overlay-id        1
     site-id          13
     preference       not set
     tag              not set
     origin-proto     connected
     origin-metric    0
     as-path          not set
     unknown-attr-len not set
            RECEIVED FROM:
peer            192.168.30.3
path-id         73
label           1002
status          C,I,R
loss-reason     not set
lost-to-peer    not set
lost-to-path-id not set
    Attributes:
     originator       192.168.30.4
     type             installed
```

```
     tloc              192.168.30.4, public-internet, ipsec
     ultimate-tloc     not set
     domain-id         not set
     overlay-id         1
     site-id           13
     preference        not set
     tag               not set
     origin-proto      connected
     origin-metric     0
     as-path           not set
     unknown-attr-len not set
            RECEIVED FROM:
peer            192.168.30.3
path-id         74
label           1002
status          C,I,R
loss-reason     not set
lost-to-peer    not set
lost-to-path-id not set
    Attributes:
     originator        192.168.30.6
     type              installed
     tloc              192.168.30.6, public-internet, ipsec
     ultimate-tloc     not set
     domain-id         not set
     overlay-id         1
     site-id           13
     preference        not set
     tag               not set
     origin-proto      connected
     origin-metric     0
     as-path           not set
     unknown-attr-len not set
```

2. On **vedge1** and **vedge3** ensure that appropriate policy from vSmart is installed and packets are matched and counted:

```
vedge1# show policy from-vsmart
from-vsmart sla-class SLA_CL1
 loss    1
 latency 100
 jitter  100
from-vsmart app-route-policy S13_S4_via_PUB
 vpn-list CORP_VPNs
  sequence 10
   match
    destination-data-prefix-list SITE4_PREFIX
   action
    count                         COUNT_PKT
    backup-sla-preferred-color biz-internet
    sla-class       SLA_CL1
    no sla-class strict
    sla-class preferred-color public-internet
from-vsmart lists vpn-list CORP_VPNs
 vpn 40
from-vsmart lists data-prefix-list SITE4_PREFIX
 ip-prefix 192.168.60.0/24

vedge1# show policy app-route-policy-filter

                COUNTER
```

```
NAME       NAME   NAME     PACKETS   BYTES
-------------------------------------------------
S13_S4_via_PUB CORP_VPNs  COUNT_PKT    81126791  110610503611
```

Besides that you should see much more packets sent via **public-internet** color from **site 13** (during my testing there was no traffic via **biz-internet** TLOC):

```
vedge1# show app-route stats remote-system-ip 192.168.30.7
app-route statistics 192.168.80.4 192.168.103.7 ipsec 12386 12366
 remote-system-ip 192.168.30.7
 local-color      public-internet
 remote-color     public-internet
 mean-loss        0
 mean-latency     1
 mean-jitter      0
 sla-class-index  0,1

       TOTAL          AVERAGE  AVERAGE  TX DATA  RX DATA
INDEX  PACKETS  LOSS  LATENCY  JITTER   PKTS     PKTS
----------------------------------------------------------
0      600      0     0        0        0        0
1      600      0     1        0        5061061  6731986
2      600      0     0        0        3187291  3619658
3      600      0     0        0        0        0
4      600      0     2        0        9230960  12707216
5      600      0     1        0        9950840  4541723

app-route statistics 192.168.109.4 192.168.103.7 ipsec 12346 12366
 remote-system-ip 192.168.30.7
 local-color      biz-internet
 remote-color     public-internet
 mean-loss        0
 mean-latency     0
 mean-jitter      0
 sla-class-index  0,1

       TOTAL          AVERAGE  AVERAGE  TX DATA  RX DATA
INDEX  PACKETS  LOSS  LATENCY  JITTER   PKTS     PKTS
----------------------------------------------------------
0      600      0     0        0        0        0
1      600      0     1        0        0        0
2      600      0     0        0        0        0
3      600      0     0        0        0        0
4      600      0     2        0        0        0
5      600      0     0        0        0        0
```

Related Information

- https://sdwan-docs.cisco.com/Product_Documentation/Software_Features/Release_18.3/07Policy_Applications/01Application-Aware_Routing/01Configuring_Application-Aware_Routing
- https://sdwan-docs.cisco.com/Product_Documentation/Software_Features/Release_18.3/02System_and_Interfaces/06Configuring_Network_Interfaces
- https://sdwan-docs.cisco.com/Product_Documentation/Command_Reference/Configuration_Comman

ds/color