Advanced SD-WAN Policies Troubleshooting

And lessons learned from field escalations

Eugene Khabarov
SD-WAN Escalation Engineer, CCIE#51348
Catalyst Engineering BU

BRKENT-3797

Webex App

Questions?

Use the Webex app to chat with the speaker after the session

How

1. Find this session in the Cisco Events mobile app
2. Click “Join the Discussion”
3. Install the Webex app or go directly to the Webex space
4. Enter messages/questions in the Webex space

Webex spaces will be moderated by the speaker until February 28, 2025.

About me: Eugene Khabarov

and why I'm the right person to talk about SD-WAN policies troubleshooting

- Engineer's Degree in Computer Systems Networking and Telecommunications (VorSTU, Russia) 2003-2008
- 15+ years in IT as support engineer, network engineer, consulting engineer, architect
- CCIE #51348 since 2015
- Joined Cisco Systems Belgium as a TAC engineer in 2017
- EMEA SD-WAN TAC Team Lead 2019-2021
- Catalyst Engineering BU escalation engineer since 2021
- LinkedIn: https://www.linkedin.com/in/enk/
- GitHub: https://github.com/enk37/

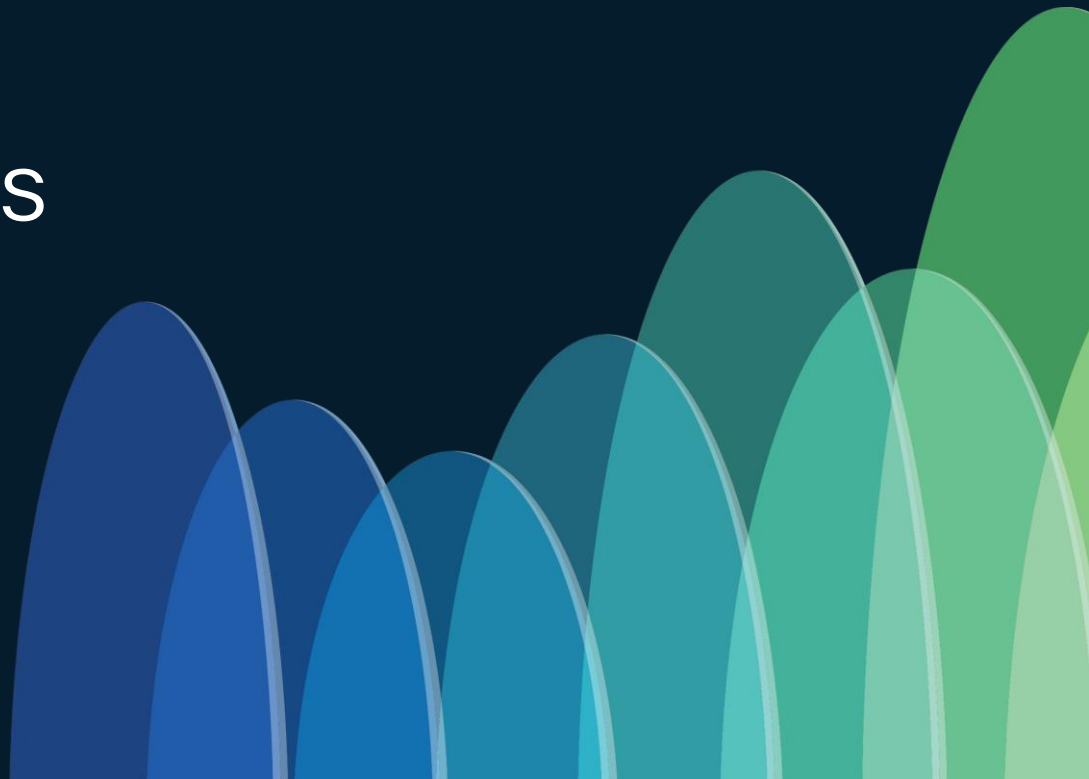
Baseline and Objectives

- Cisco SD-WAN basic level knowledge required at least because it is advanced level session, very technical
- The session main objectives:
  - to demonstrate useful policies troubleshooting tools and techniques
  - to share experience about some typical or interesting issues seen in the field
- It is not a comprehensive guide; there are always additional issues not covered here
- Consider the session as a "cookbook" for SD-WAN policy failures, but not a "Tour de Force."
- The session primarily focuses on centralized policies, but I will also briefly discuss localized policies.
- Main topics covered:
  - Policies troubleshooting workflow
  - Available troubleshooting toolset
  - Internal components of IOS-XE responsible for policies programming and implementation
  - Common pitfalls and challenges
- Heavily CLI based, old-school classic ☺
- Recommended prerequisite session: Advanced SD-WAN Routing Troubleshooting (BRKENT-3793)

Agenda

- Part 1: SD-WAN Polices Troubleshooting Basics
  - 1.1 SD-WAN Policies Quick Overview
  - 1.2 Troubleshooting SD-WAN policies from vManage perspective
  - 1.3 Centralized Control Policies troubleshooting workflow
  - 1.4 Centralized Data and AAR Policies troubleshooting workflow
- Part 2: Issues seen in the field
  - 2.1 Not-so-well-known failures with centralized control policies
  - 2.2 Interesting cases with centralized data and AAR policies

Part 1.
SD-WAN Policies Troubleshooting Basics



Before we begin, disclaimer: new Cisco Catalyst SD-WAN components naming

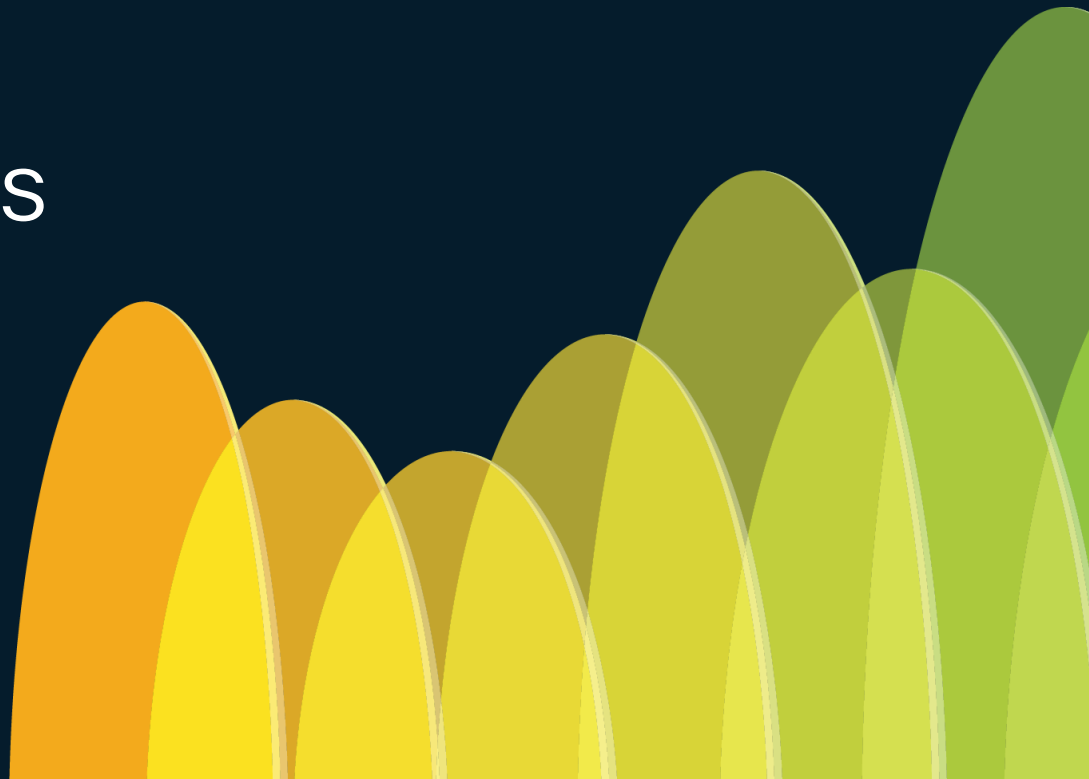
- vManage (NMS) == Catalyst SD-WAN Manager
- vBond (orchestrator) == Catalyst SD-WAN Validator
- vSmart == Catalyst SD-WAN Controller

I will stick to the legacy names vManage/vBond/vSmart in the slides

Why? Because I like the old names

But seriously, it is to avoid confusion because they are historically called so and in all CLI outputs we uses their original names. They will remain the same (vmanage/vbond/vsmart), there are no plans to change it.

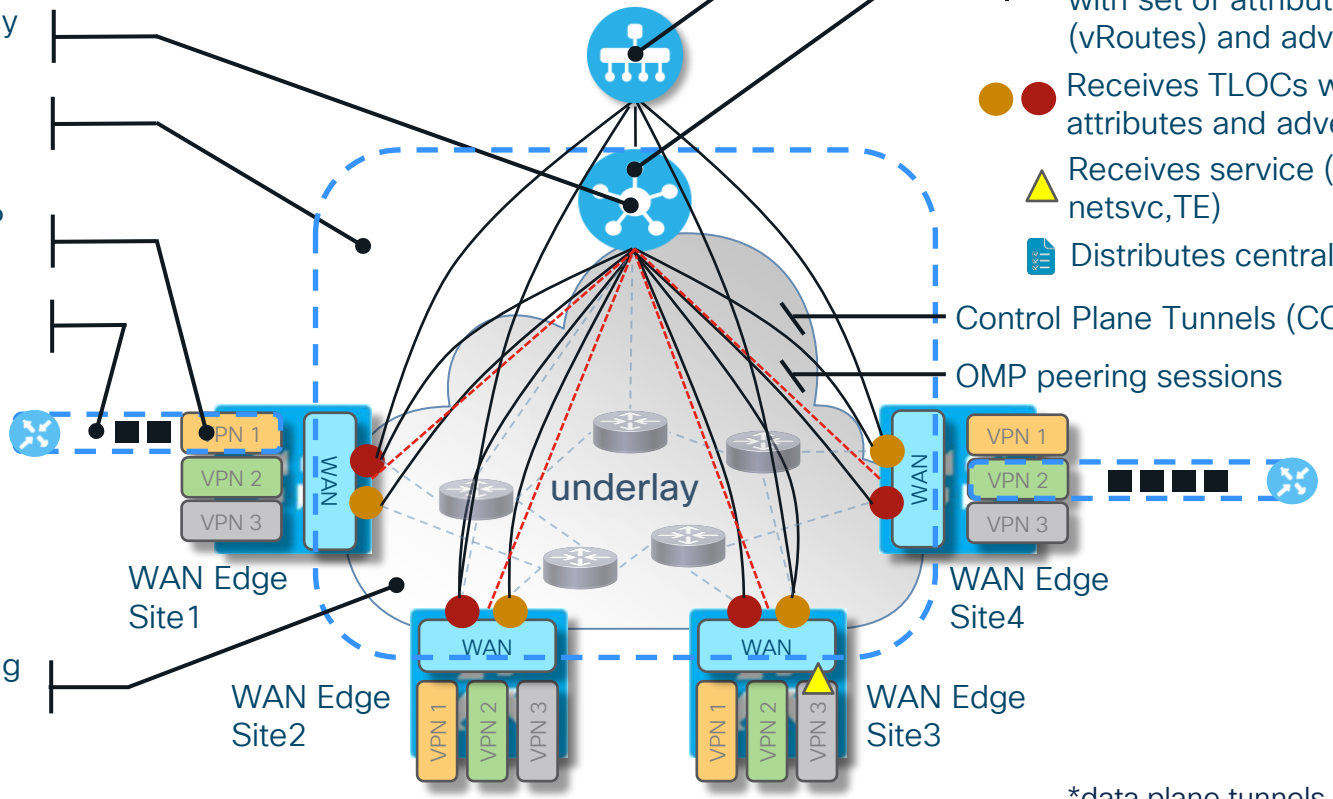
SD-WAN Policies Quick Overview



cisco Live!

Cisco Catalyst SD-WAN Overlay

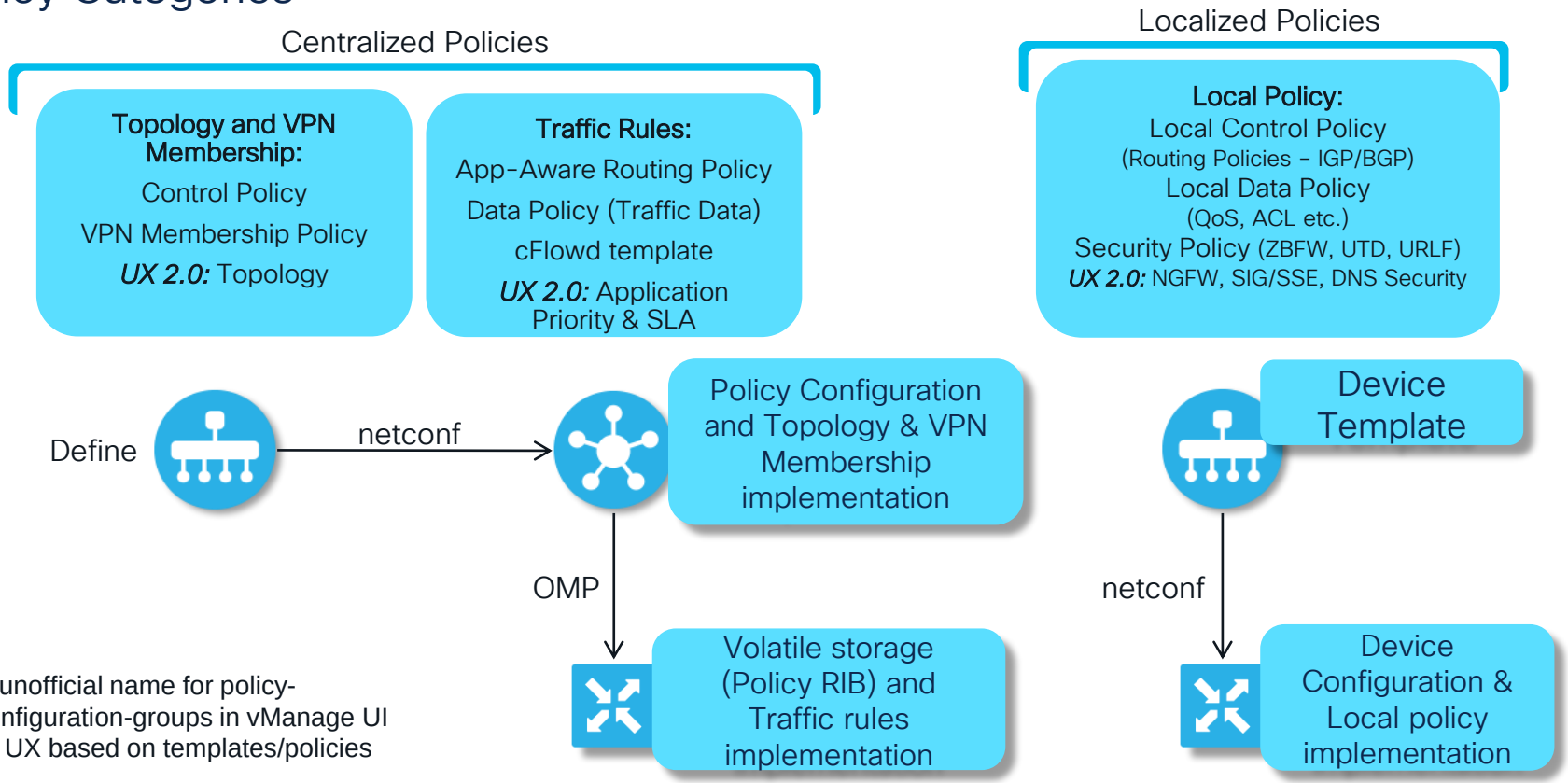
Fabric Components Quick Recap



*data plane tunnels are not shown

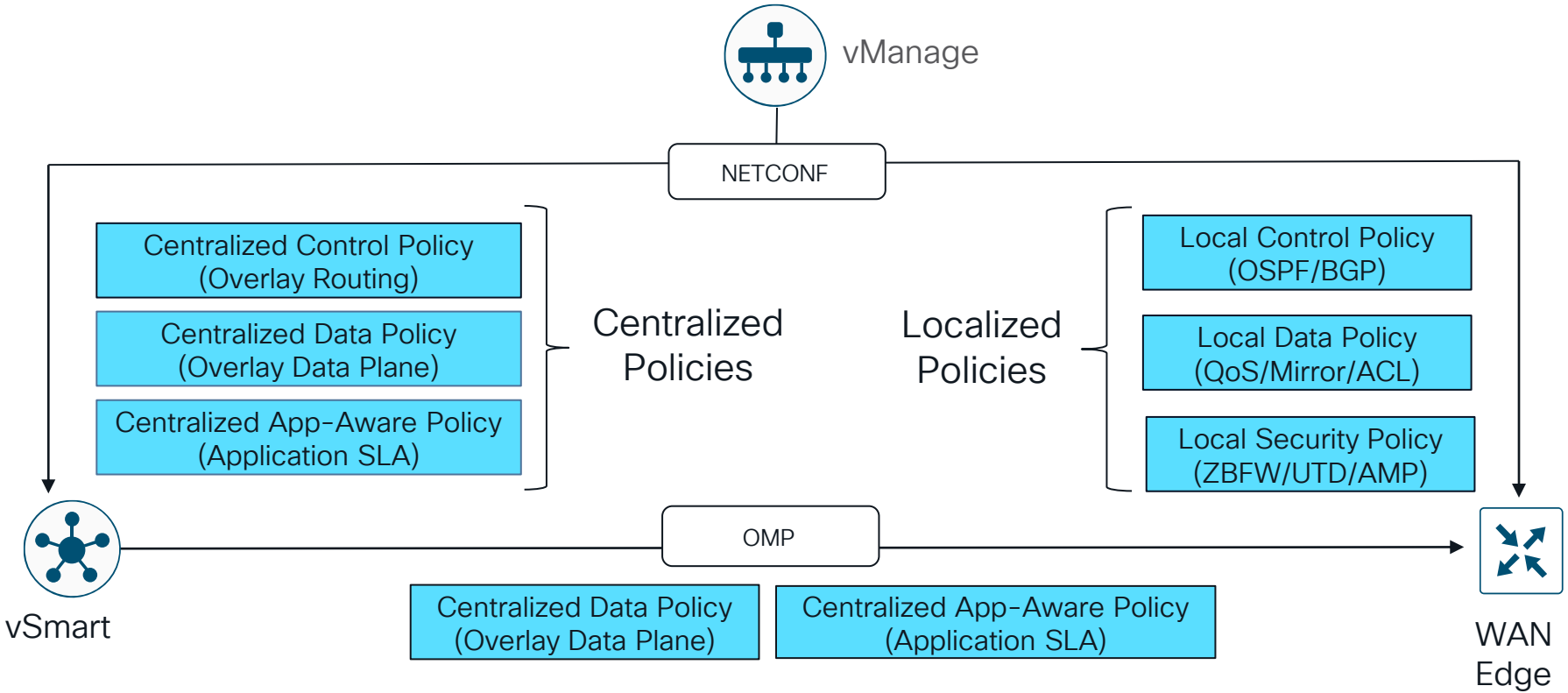
Cisco SD-WAN Policy Architecture

Policy Categories

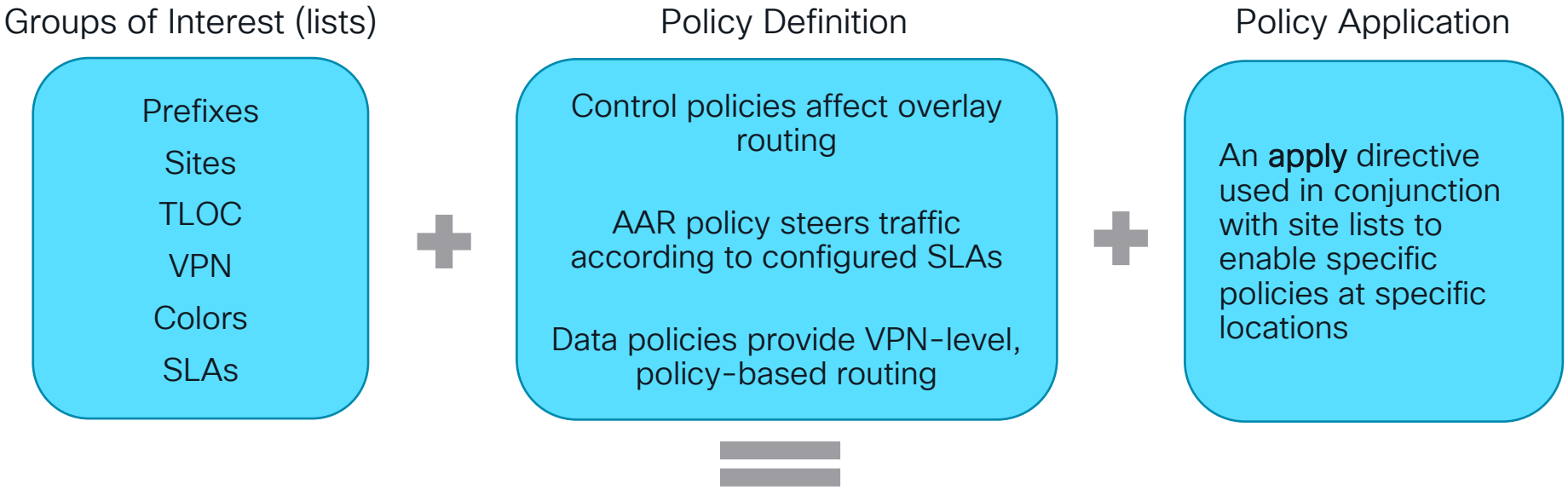


**UX 2.0* - unofficial name for policy-groups/configuration-groups in vManage UI vs classic UX based on templates/policies

Policies



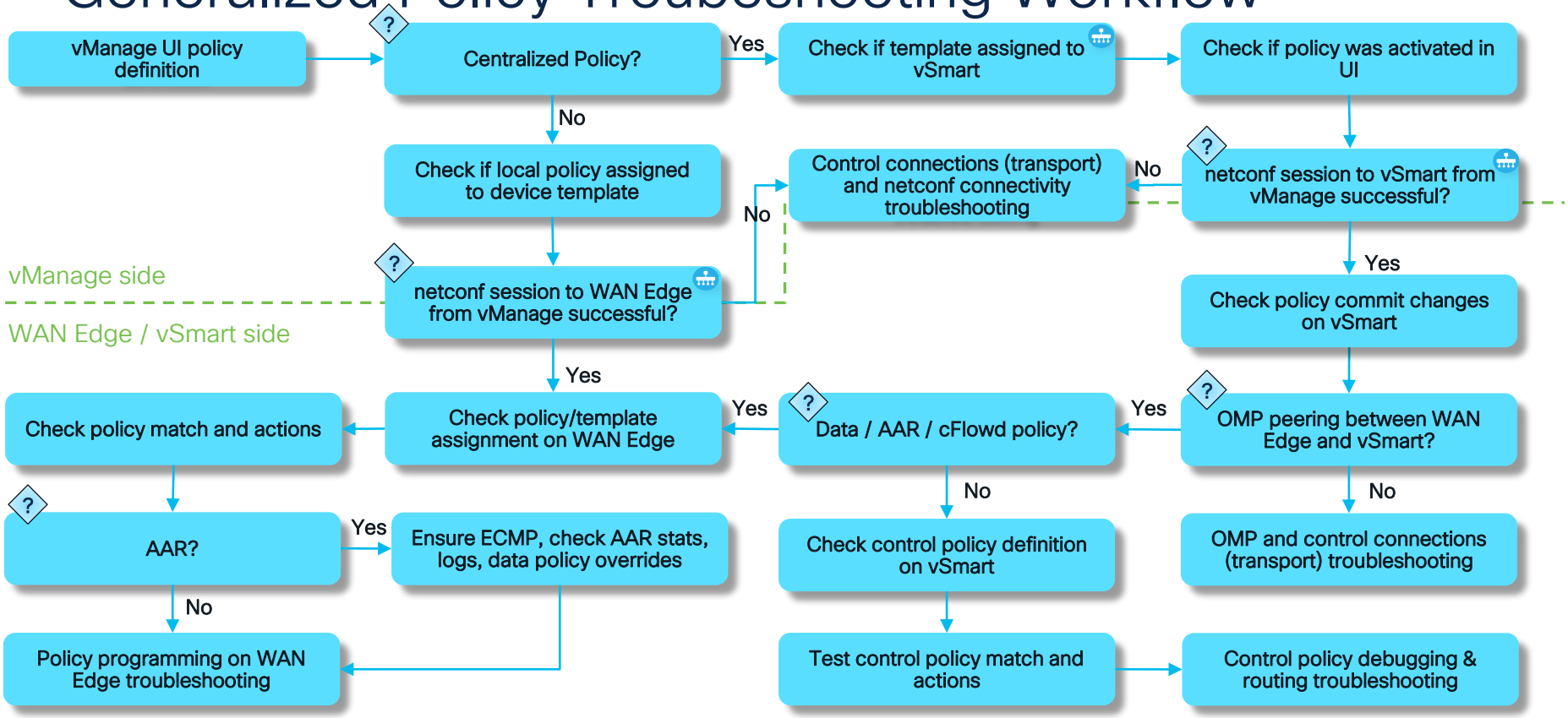
Building Blocks of Centralized Policies



Centralized policy definition is configured on vManage and enforced across the network (on a device or vSmart controller depending on type)

Generalized Policy Troubeshooting Workflow

For Your Reference



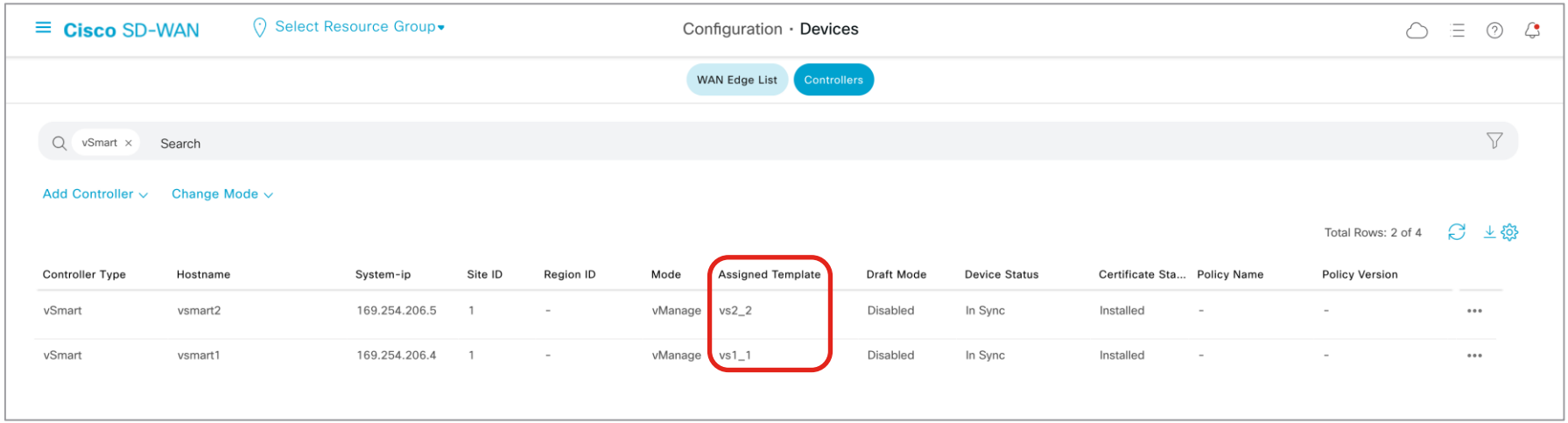
* vManage automatically verify this and warns us in case of problems

Troubleshooting SD-WAN policies from vManage perspective

cisco Live!

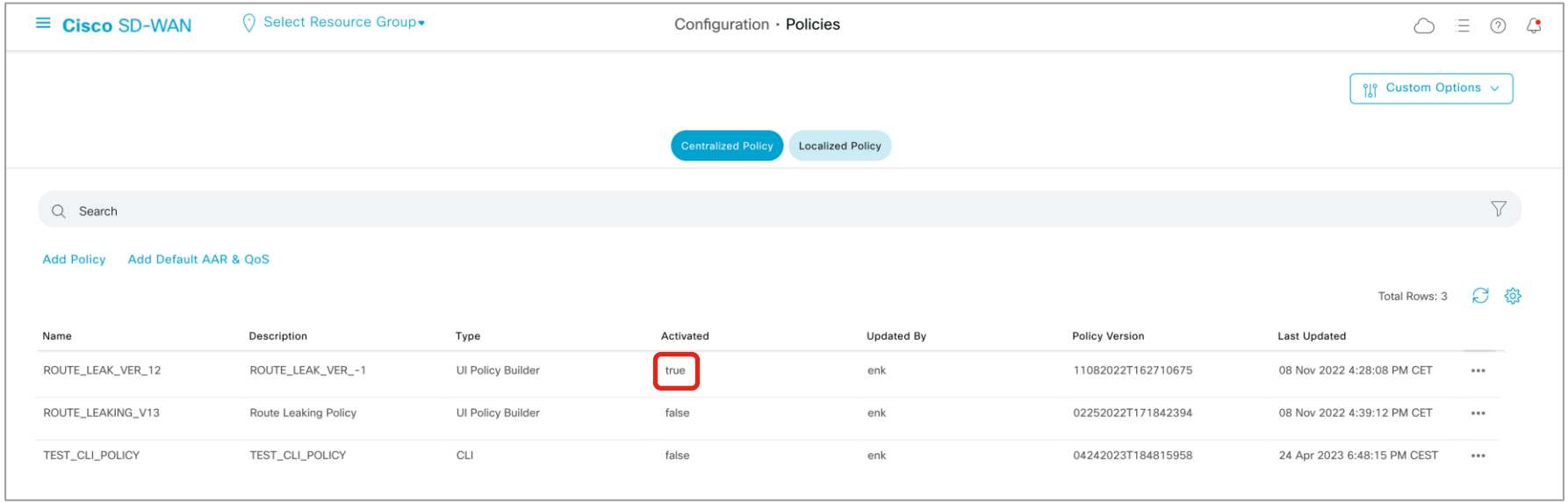
Centralized Policy: Check if template assigned to vSmart

Configuration -> Devices -> Controllers



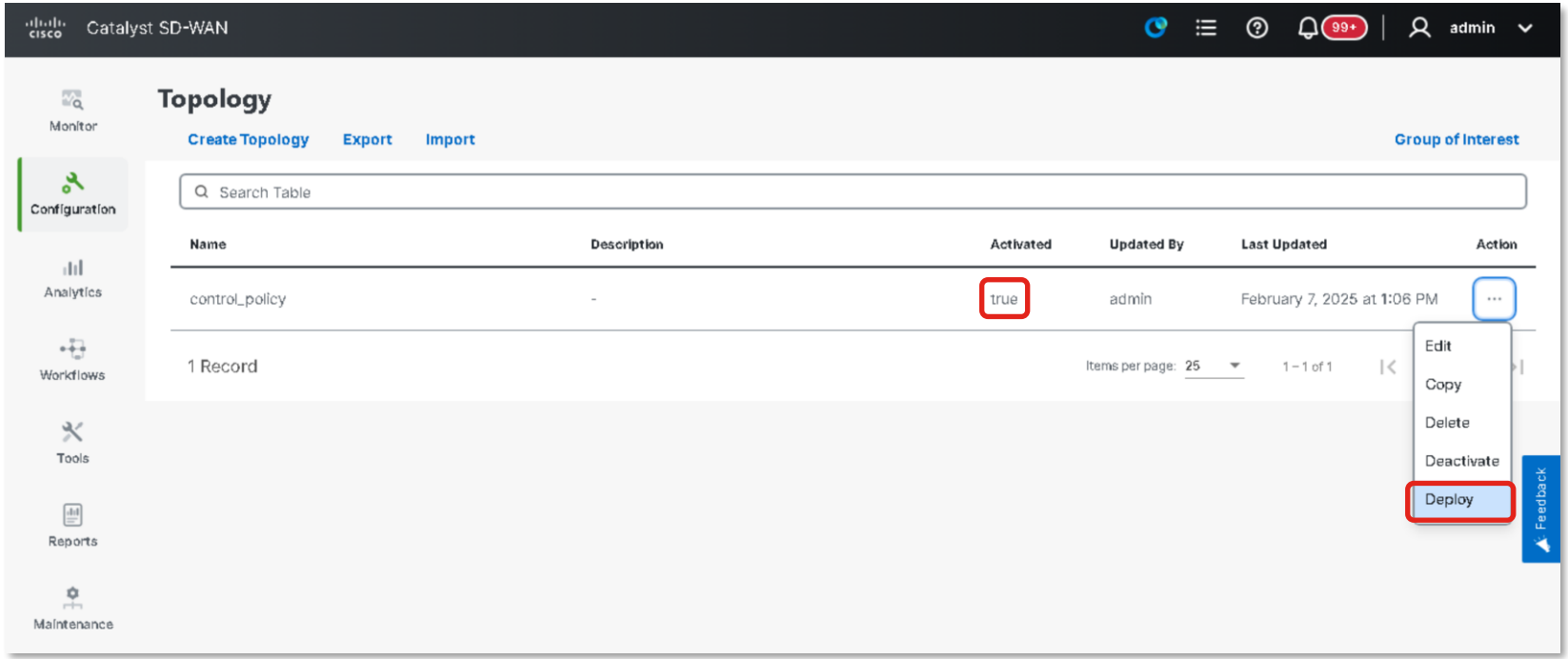
Centralized Policy: Check if policy was activated in UI

Configuration -> Policices -> Centralized Policy



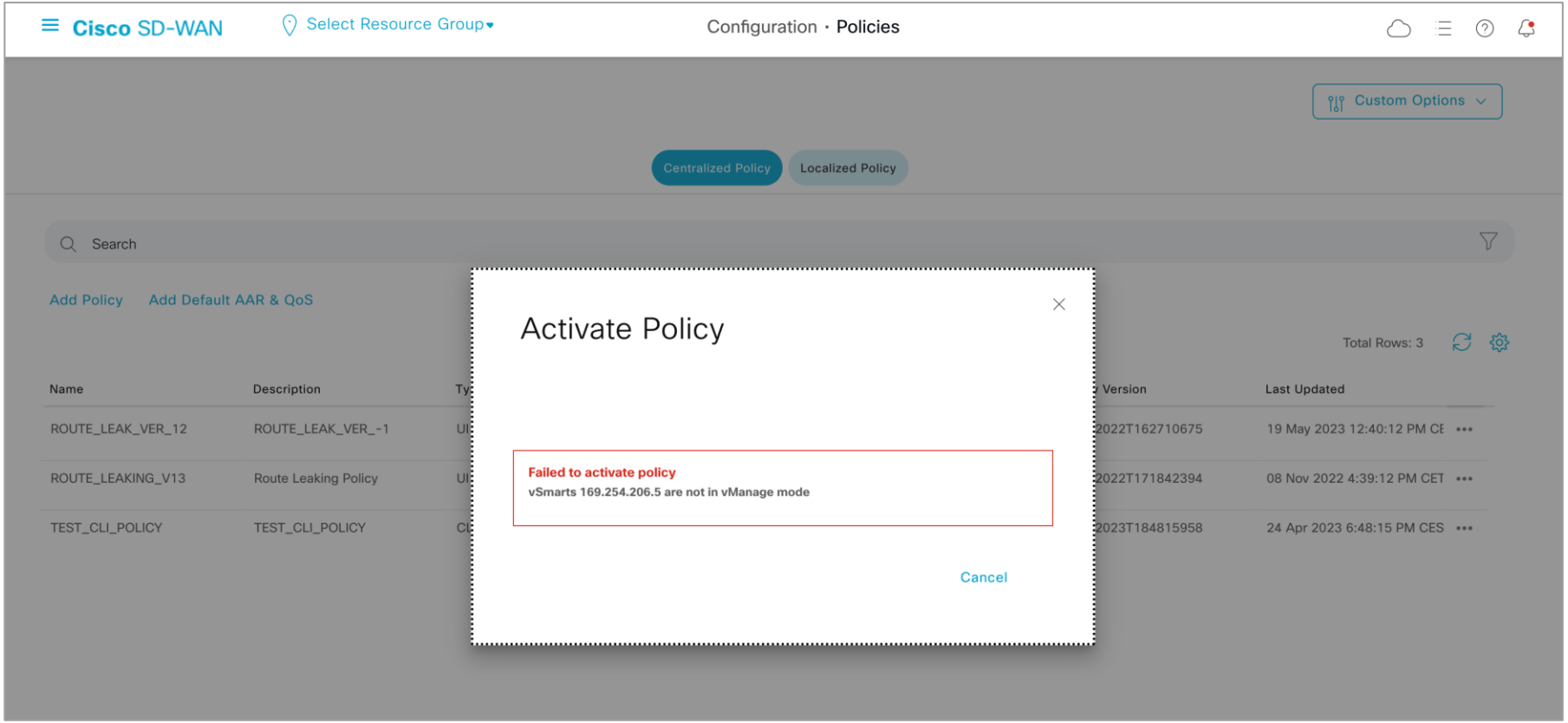
Centralized Policy: Check if policy was activated in UX 2.0

Configuration -> Topology

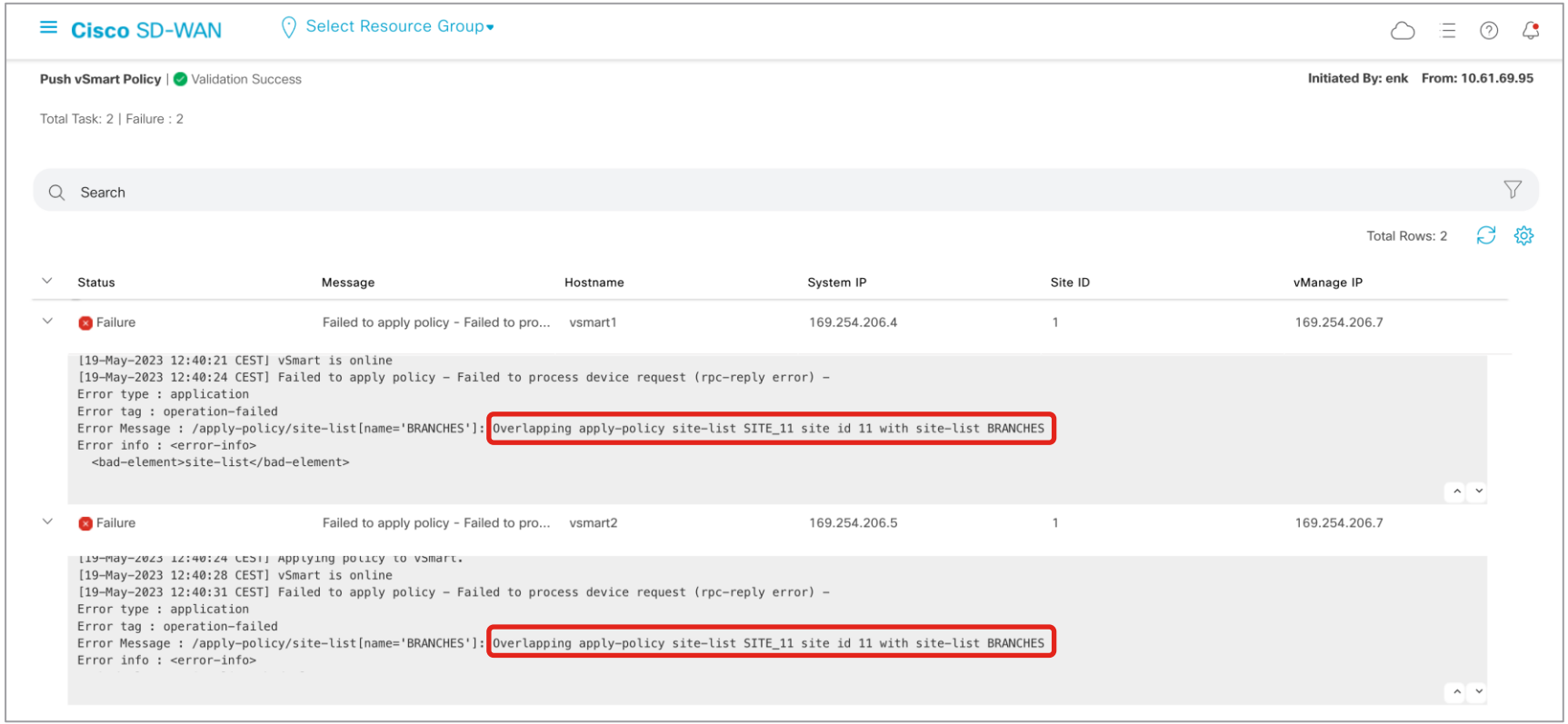


Typical catch and the difference with classic UI – policy changes must be deployed also after policy changed

Centralized Policy: Check if template assigned to vSmart

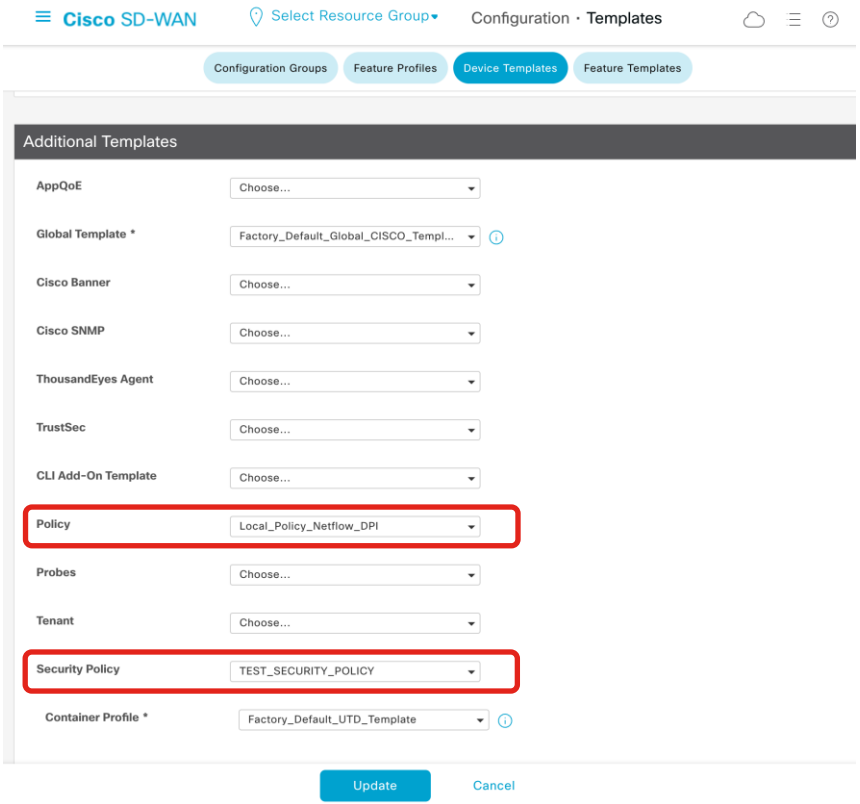


Centralized Policy: Policy Activation Issues



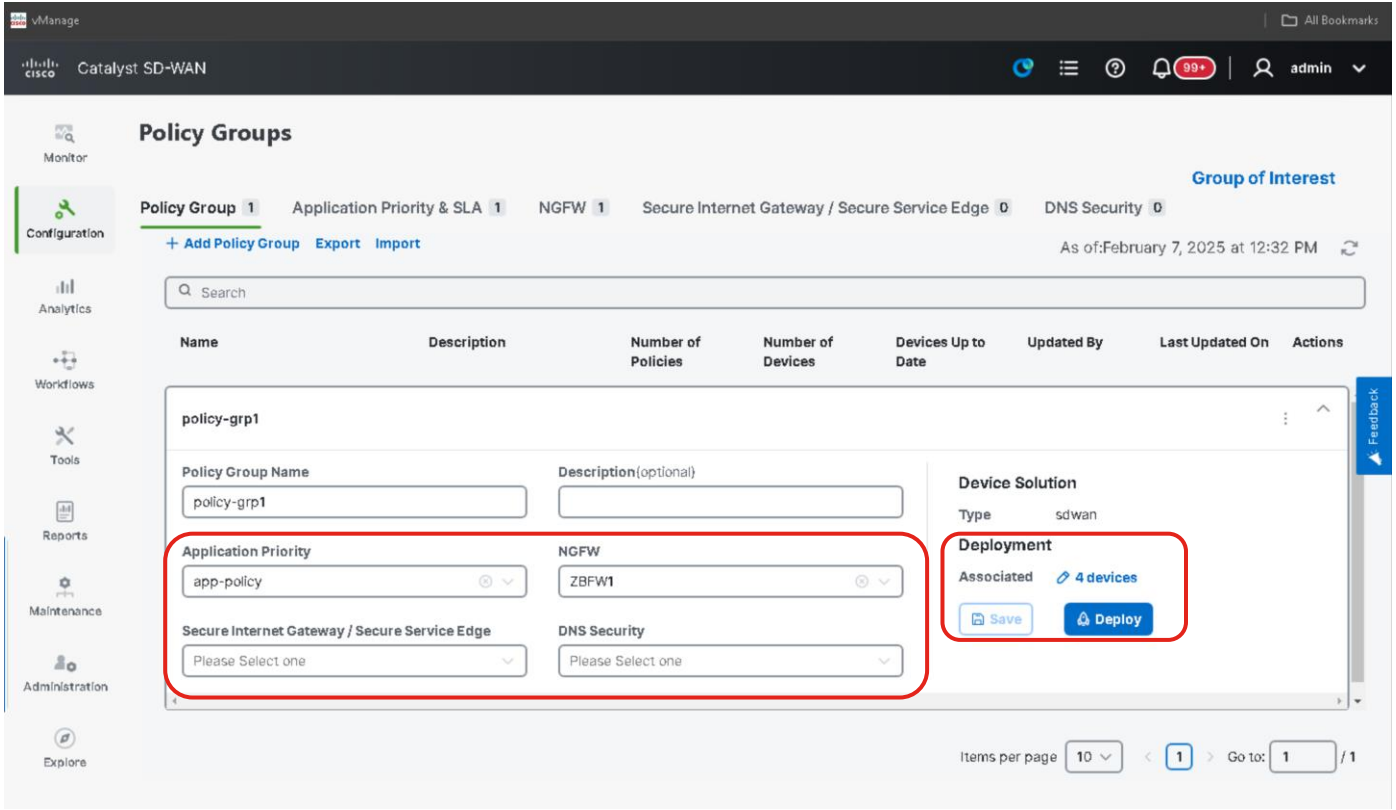
Localized Policy: Check if policy assigned to device template

Configuration -> Templates -> Device Template -> Additional Templates section



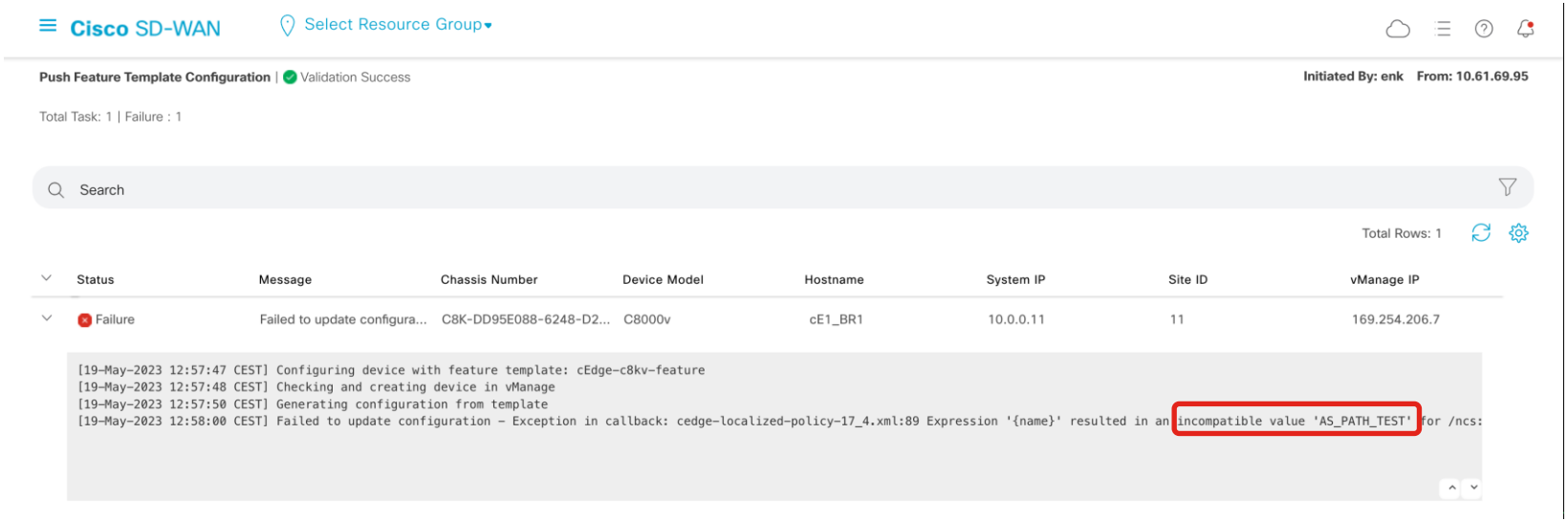
Localized Policy: Check if policy assigned to policy-group UX 2.0

Configuration -> Policy-Groups -> expand policy-group details



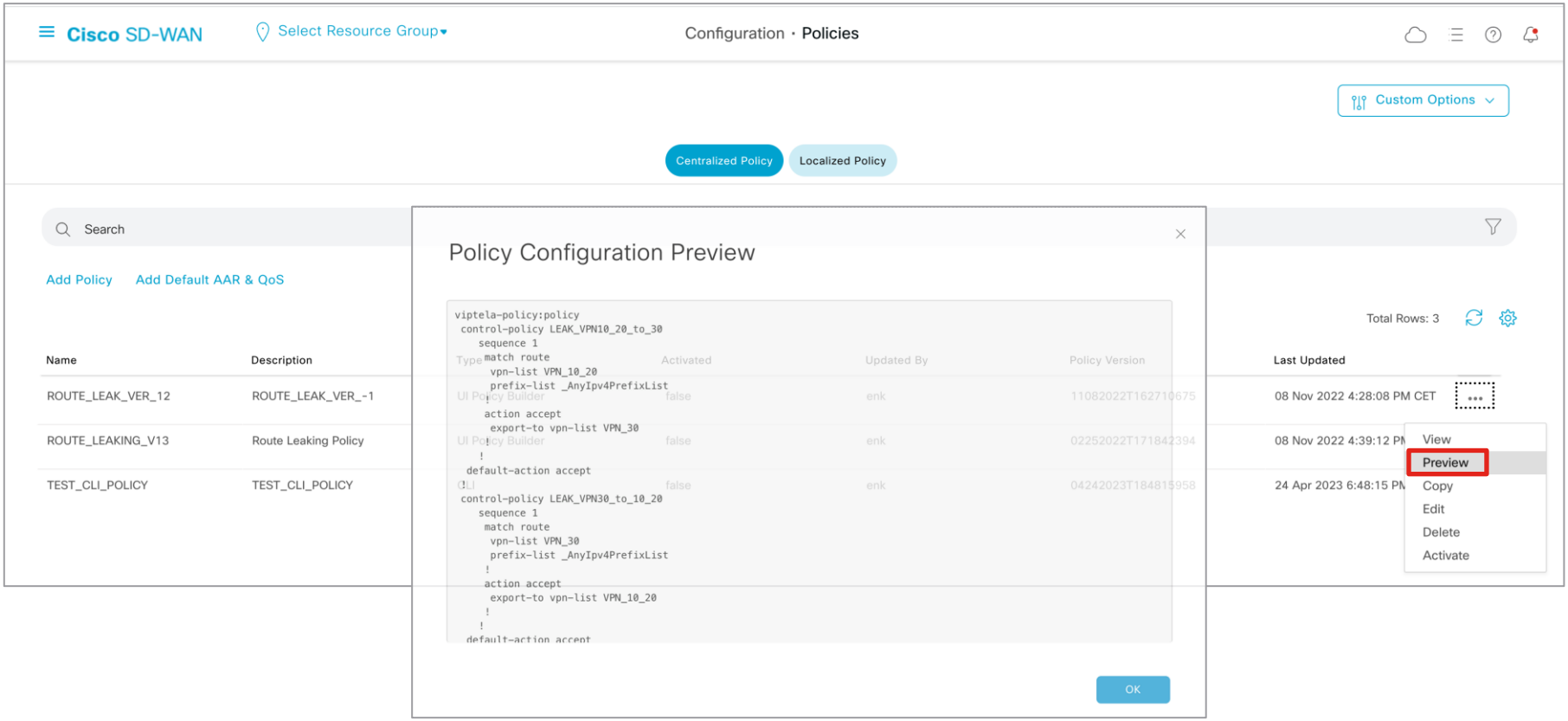
Localized Policy: Device Template Assignment Issues

Configuration -> Templates -> Device Template -> ... -> Attach Devices

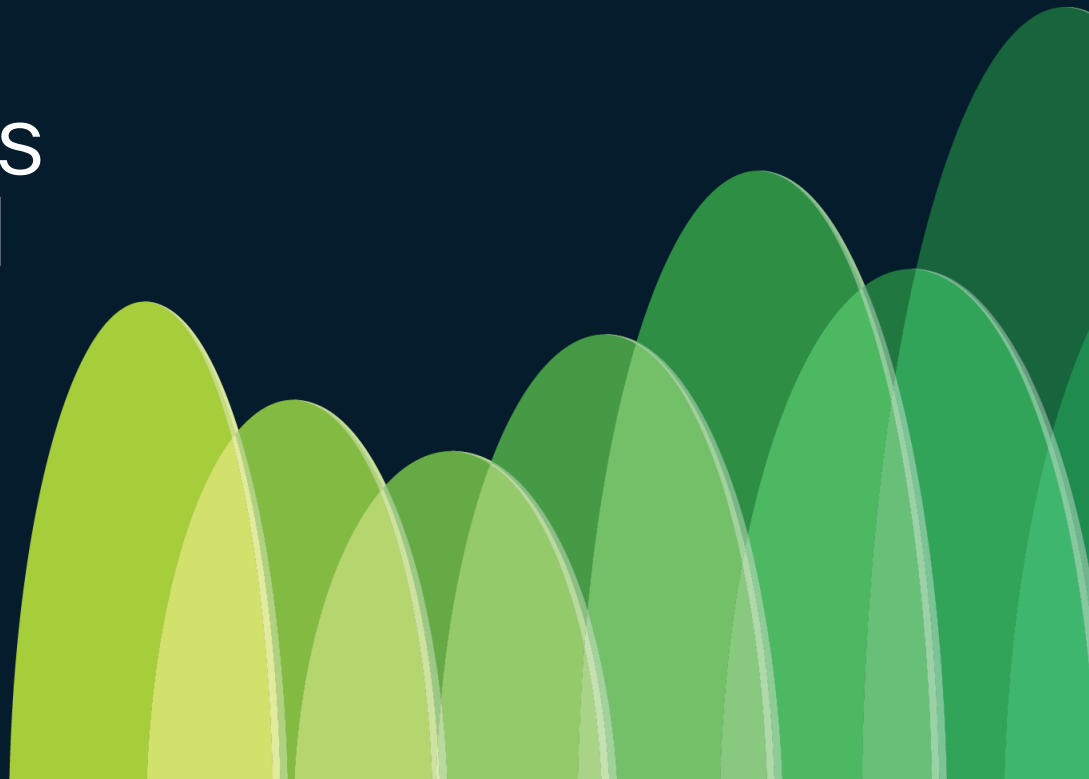


* Here is the reason that AS_PATH_TEST contains typo "^^*$"

Policy Preview in vManage

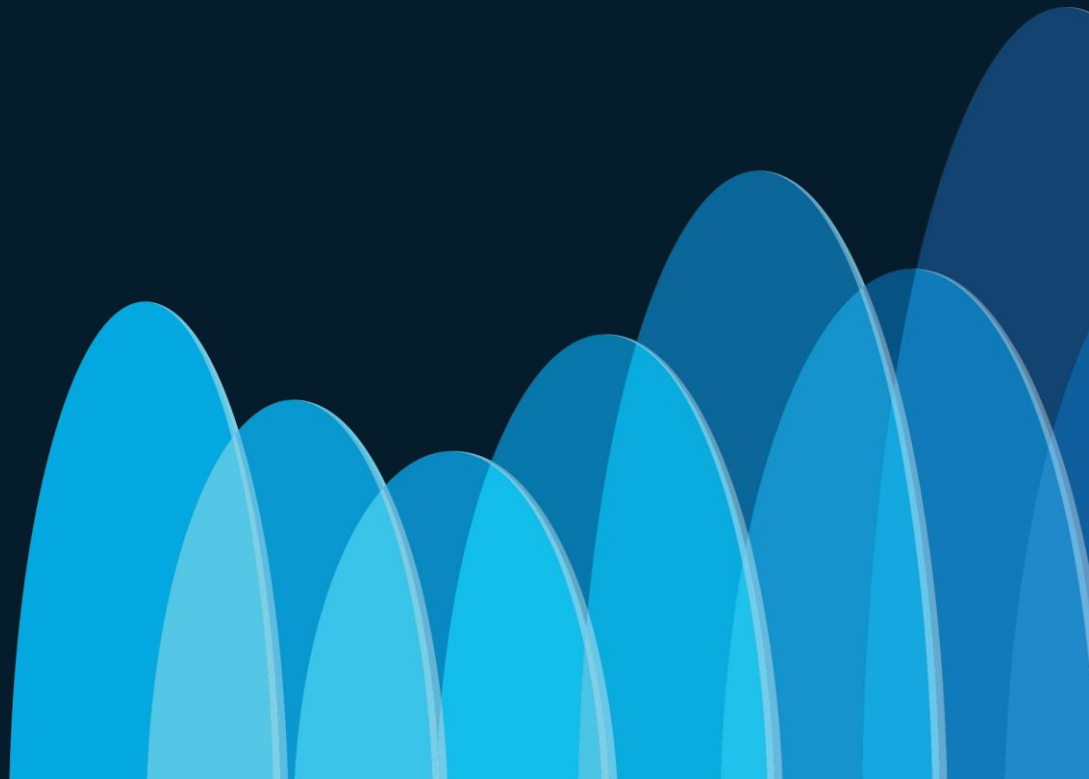


Troubleshooting SD-WAN policies from vSmart and WAN Edge perspective



cisco Live!

Centralized Control Policy Troubleshooting 101

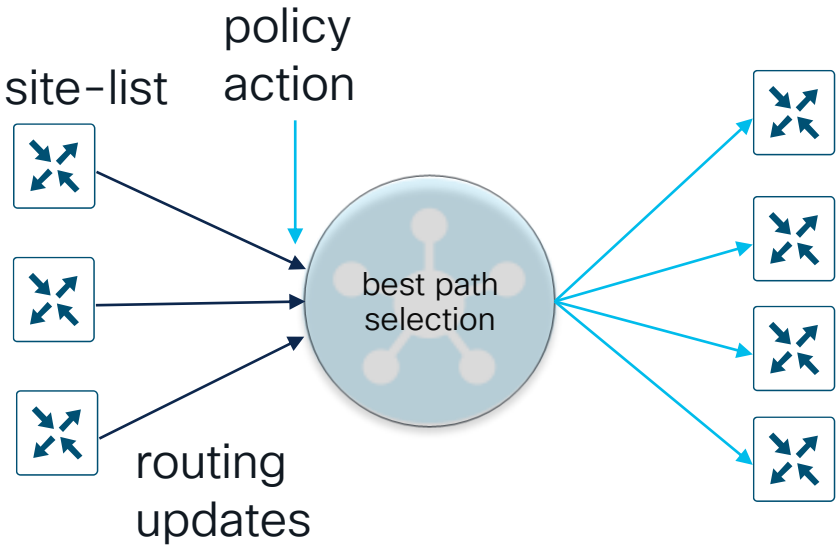


cisco Live!

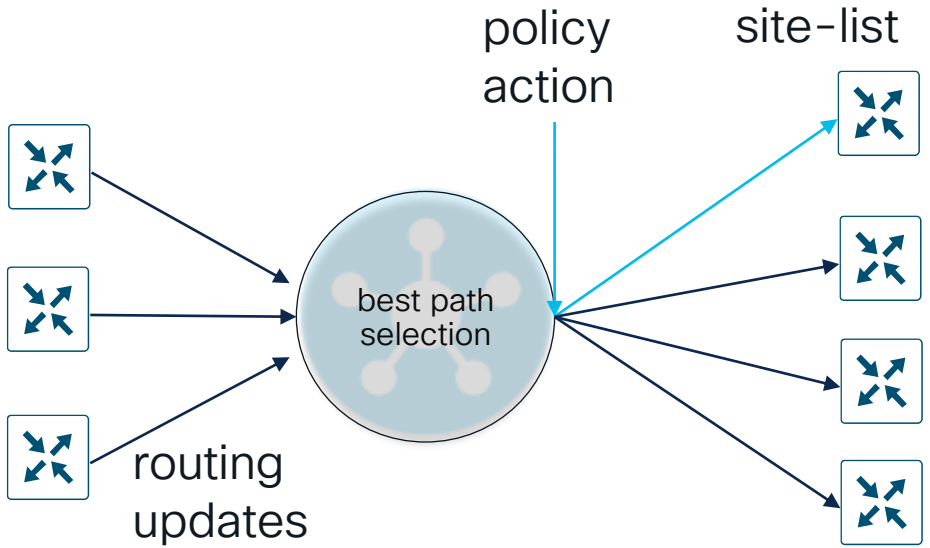
Centralized Control Policy Application

Most important concept to rememer for control policies troubleshooting

Policy applied in the inbound direction



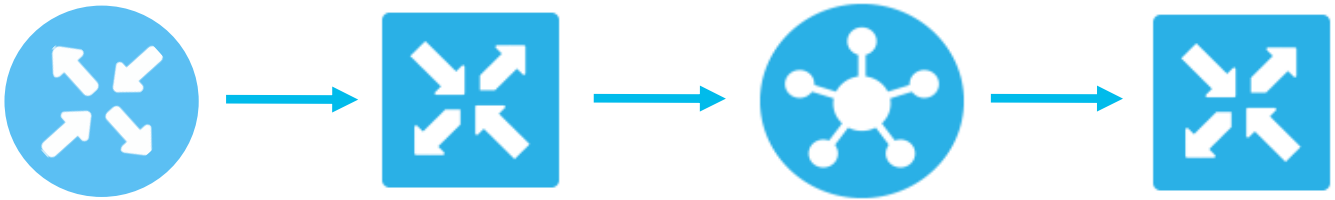
Policy applied in the outbound direction



Centralized Control Policies Troubleshooting Workflow (1)

1. Check policy commit changes:
   `show configuration commit changes <number>`
2. Check OMP peering between WAN Edge and vSmart to ensure policy can be applied on routing updates to/from WAN Edge:
   `show omp peers <system-ip>`
3. Check which control policy assigned and direction of assignment
   `show support omp peer peer-ip <system-ip> | include –pol`
4. Check that vManage UI polciy definition was sucessfully translated into CLI representation on vSmart (you should see the same things as in vManage policy preview):
   `show configuration commit changes`
   `show run apply-policy site-list <name> control-policy <name>`
   `show run policy list <name>`
   `show run policy control-policy <name>`
5. Check control policy match and actions (logic)
   `test policy match control-policy <name> <conditions>`
6. Proceed with overlay routing troubleshooting, more in BRKENT-3793

Recap: overlay routing troubleshooting: missing route(s) algorithm



*Routing info announcement direction

Check on WAN Edge:

1. RIB/FIB (**`show ip route/show [sdwan] ip fib`**)
2. OMP table if route is not in RIB (**`show [sdwan] omp route`**)
3. TLOC information presented (**`show [sdwan] omp tloc`**)
4. BFD session with remote TLOC (**`show [sdwan] bfd sessions`**) -> troubleshoot data plane tunnels
5. Local policy filtering on redistribution to/from OMP table (**`show sdwan run "sdwan omp", show sdwan run "policy", show run route-map`**)

Check on vSmart:

- OMP route and TLOC tables on vSmart (**`show omp route, show omp tloc`**)

Centralized Control Policies Troubleshooting Workflow (2)

- Last resort: start debugging on vSmart:
  - **`debug omp policy [level <high|low> peer-address <system-ip> prefix <IP prefix/length> direction <both|received|sent> vpn <number>]`**
  - Before 20.12 logs stored in **`/var/log/tmplog/vdebug`**
  - 20.12+ logs stored in **`/var/log/vdebug`**
    - Ensure to enable disk logging for debug messages:
      `vSmart1(config)# system logging disk enable priority debug`
  - To view them:
    - enter **`vshell`** and use **`tail -f <filename>`**
    - Or simply **`show log <filename> tail -f`**
    - Or **`monitor start <filename>`** and logs will be printed into your terminal

Centralized Control Policies Troubleshooting Workflow:

Commands usage examples

cisco Live!

Centralized Control Policies Troubleshooting Commands (1)

1. Check policy commit changes **`show configuration commit changes <number>`**:

```
vsmart1# show configuration commit changes 0
!
! Created by: vmanage-admin
! Date: 2023-04-24 19:22:02
! Client: netconf
!
policy
 lists
  site-list BRANCHES
   site-id 11-12
  !
  site-list SITE-30
   site-id 30
  !
  site-list SITE-40
   site-id 40
  !
  prefix-list DEFAULT
   ip-prefix 0.0.0.0/0
  !
 !
 control-policy MY-CONTROL-POLICY-v1
  sequence 1
   match tloc
    site-list SITE-30
   !
   action accept
   !
  !
  sequence 11
   match tloc
    site-list SITE-40
   !
   action accept
   !
  !
  sequence 21
   match route
    prefix-list DEFAULT
    site-list   SITE-30
   !
   action accept
    set
     preference 100
     service netsvc3 vpn 3
    !
   !
  !
  sequence 31
   match route
    prefix-list DEFAULT
    site-list   SITE-40
   !
   action accept
    set
     preference 50
     service IDP vpn 3
    !
   !
  !
  default-action reject
 !
!
apply-policy
 site-list BRANCHES
  control-policy MY-CONTROL-POLICY-v1 out
 !
!
```

Centralized Control Policies Troubleshooting Commands (2)

2. Check OMP peering between WAN Edge and vSmart to ensure policy can be applied on routing updates to/from WAN Edge:

`show omp peers <system-ip> [details]`

```
vsmart1# show omp peers 10.0.0.11
R -> routes received
I -> routes installed
S -> routes sent

                        DOMAIN    OVERLAY   SITE
PEER            TYPE    ID        ID        ID        STATE    UPTIME           R/I/S
-------------------------------------------------------------------------------------
10.0.0.1        vedge   1         1         30        up       1:18:11:17       18/0/94
```

Centralized Control Policies Troubleshooting Commands (3)

3. Check control policy assignment and direction of assignment

`show support omp peer peer-ip <system-ip> | include –pol`

Can be used to find which policies applied to a peer and which site-list it belons:

```
vsmart1# show support omp peer peer-ip 10.0.0.11 | include -pol
        site-pol: BRANCHES route-pol-in: None route-pol-out: MY-CONTROL-POLICY-v1 data-pol-in: None
           data-pol-out: None pfr-pol: None mem-pol: None cflowd:None
```

Centralized Control Policies Troubleshooting Commands (4)

4. Check if vManage UI policy definition was sucessfully translated into CLI representation on vSmart, **`policy`** section):

```
vsmart1# show running-config policy control-policy MY-CONTROL-POLICY-V1
policy
 control-policy REMOTE-TOPOLOGY-POLICY-PPC-rev1
  sequence 1
   match tloc
    site-list SITE-30
   !
   action accept
   !
  !
  sequence 11
   match tloc
    site-list SITE-40
   !
   action accept
   !
  !
  sequence 21
   match route
    prefix-list DEFAULT
    site-list   SITE-30
   !
   action accept
    set
     preference 100
     service netsvc3 vpn 3
    !
   !
  !
  sequence 31
   match route
    prefix-list DEFAULT
    site-list   SITE-40
   !
   action accept
    set
     preference 50
     service IDP vpn 3
    !
   !
  !
  default-action reject
 !
!
```

Centralized Control Policies Troubleshooting Commands (4 cont.)

... and **`apply-policy`** section:

```
vsmart1# show running-config apply-policy site-list BRANCHES
apply-policy
 site-list BRANCHES
  control-policy MY-CONTROL-POLICY-v1 out
 !
!

vsmart1# show running-config policy lists site-list BRANCHES
policy
 lists
  site-list BRANCHES
   site-id 11-12
  !
 !
!
```

Centralized Control Policies Troubleshooting Commands (5)

5. Check control policy match and actions

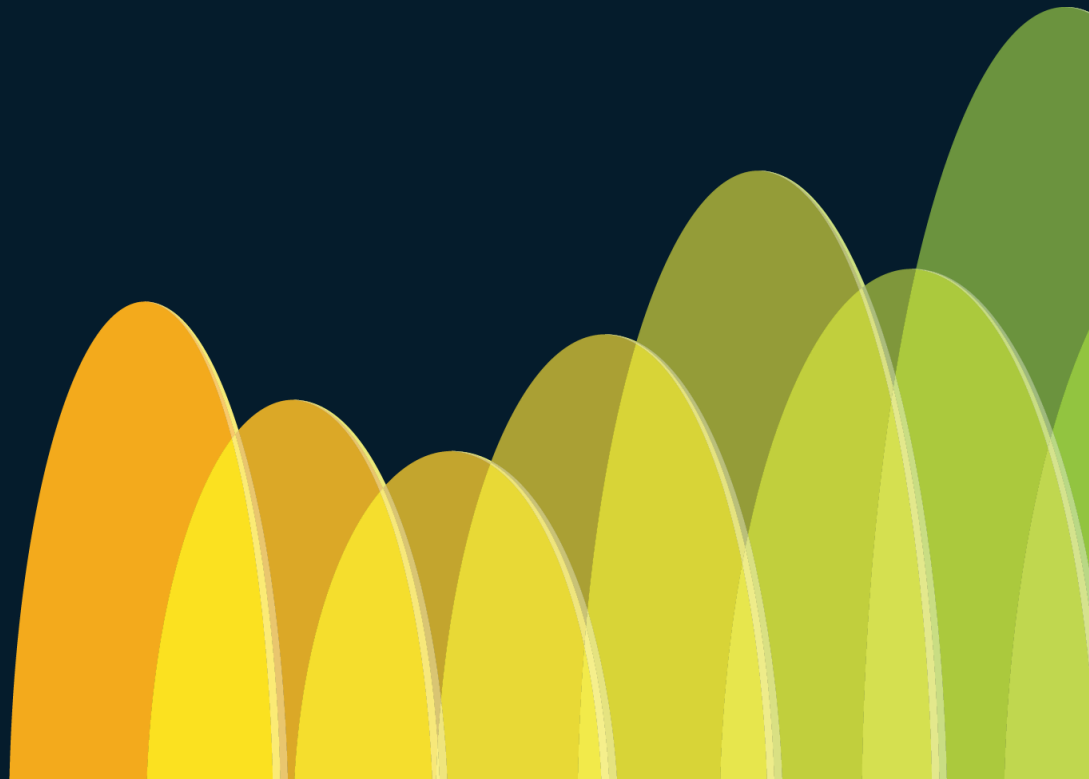
`test policy match control-policy <name> <conditions>`

- The command can be used to find matching sequence in a control policy on vSmart

```
vsmart1# test policy match control-policy MY-CONTROL-POLICY-V1 site-id 40 ipv4-prefix DEFAULT
Found: "site-id 40 ipv4-prefix-list DEFAULT" matches policy MY-COJNTROL-POLICY-v1 sequence 31
    sequence: 31
        match route [SITE-LIST PFX-LIST  (0x11) ]
            site-list: SITE-40 (0x7f15b90bfc00)
            IPv4 prefix-list: DEFAULT (0x7f15b90bfc80)
        action: accept
        set: [PREF SERVICE  (0x44) ]
            preference: 50
            service: 3 vpn: 3 tloc: :: : invalid : ipsec [none]
```

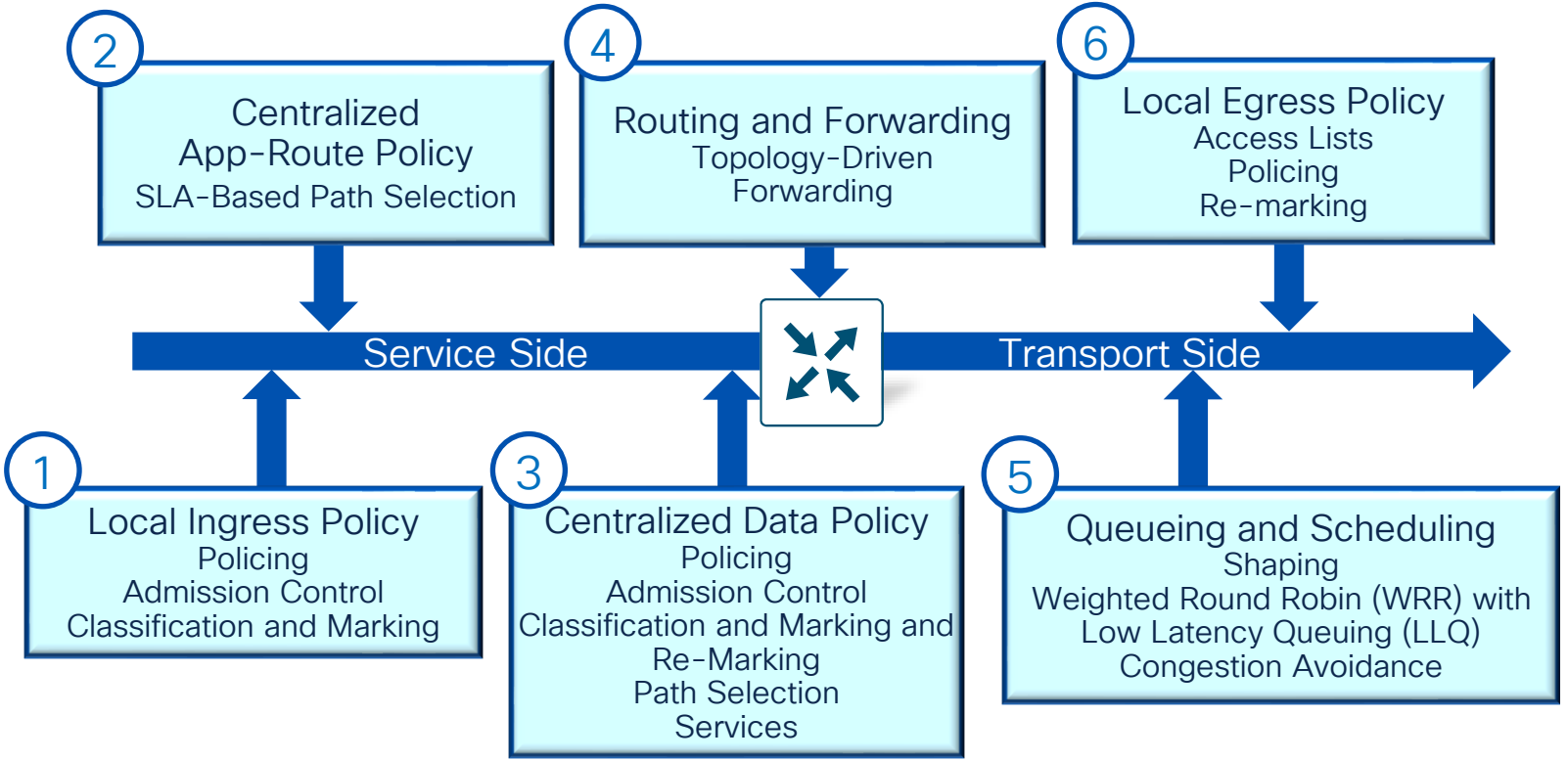
6. Examples on routing related policy troubleshooting will folow in Part 2.

Centralized Data and AAR Policies Troubleshooting



cisco Live!

Most important concept: Order of Operations

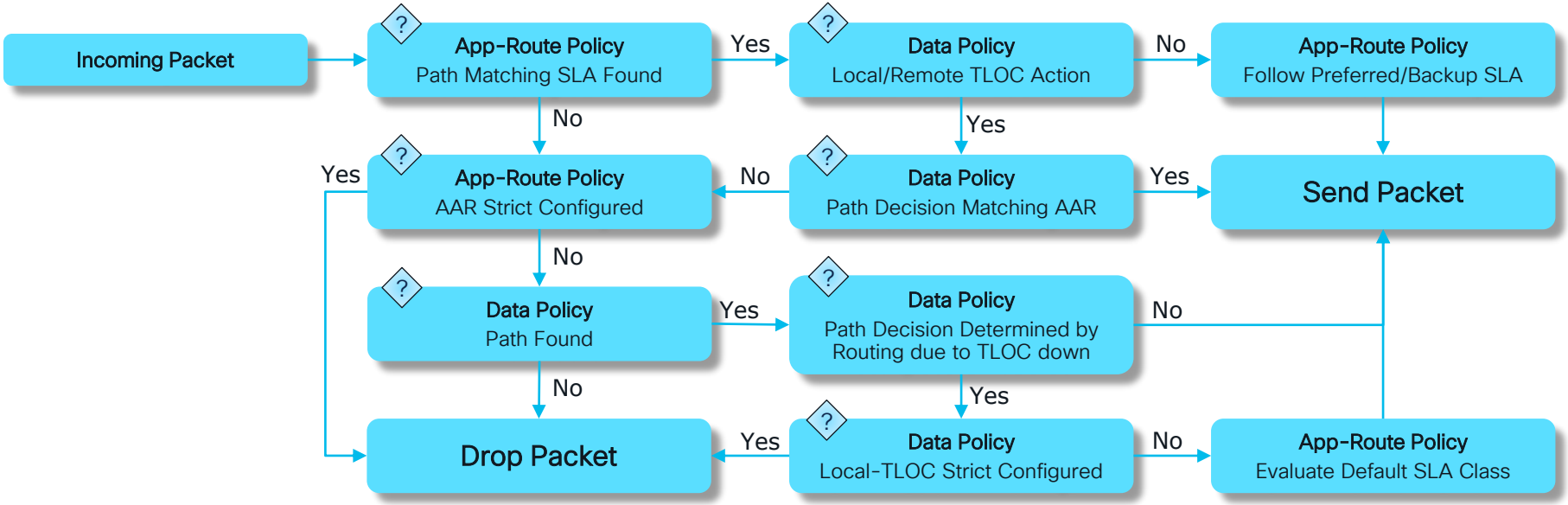


App-Aware Routing and Data Policy Overlap

Policy Processing when packet is subject to match in both policies

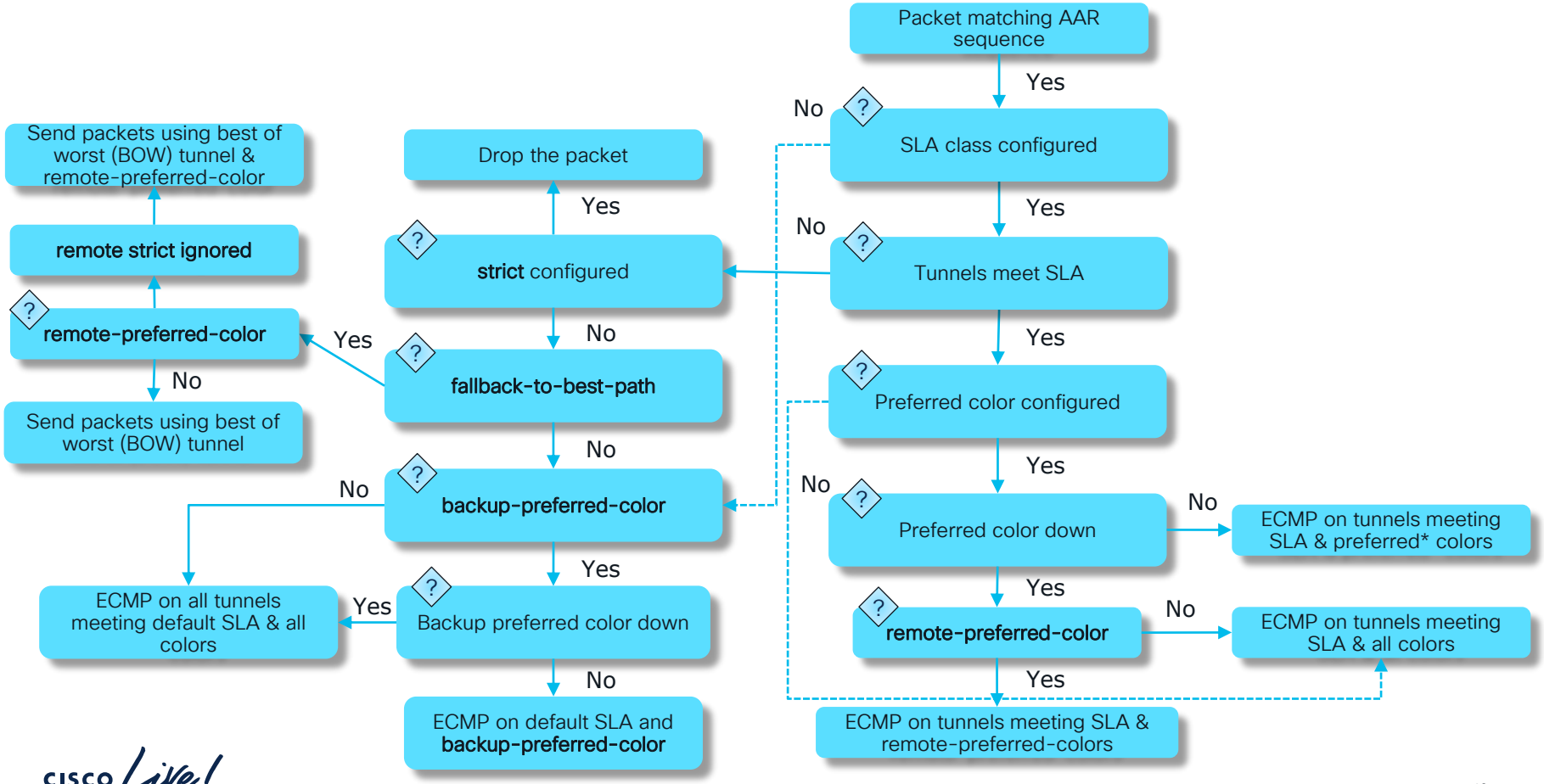
For Your Reference

Guiding Principle:
Data Policy Makes Final Decision with Consideration for AAR SLA Match



App-Aware Routing Tunnel Selection Flowchart

For Your Reference



App-Aware Routing Remote Preferred Color Feature

Key concepts about new feature **`remote-preferred-color`**:

- **`remote-preferred-color`** can coexist with (local) preferred color
- If none of tunnels meet SLA and best-of-worst (BOW) **`fallback-to-best-path`** confugured, some tunnels meet BOW criterias. Then **`remote-preferred-color`** will be respected for BOW
  - but **`remote-color-restrict`** will be ignored because intention is to use BOW and forward traffic and not to drop it
  - Hence **`remote-color-restrict`** applicable only in non-BOW scenarios

Data and AAR Policies Troubleshooting Workflow (1)

From vSmart perspective, it is similar to control policy workflow:

1. Check policy commit changes:
   `show configuration commit changes <number>`
2. Check OMP peering between WAN Edge and vSmart to ensure policy can be sent:
   `show omp peers <system-ip>`
3. Check AAR/Data policy assignment and direction of assignment
   `show support omp peer peer-ip <system-ip> | include –pol`
4. Check that vManage UI polciy definition was sucessfully translated into CLI representation on vSmart:
   `show run policy list <name>`
   `show run policy <data-policy|app-route-policy> <name>`
   `show run apply-policy site-list <data-policy|app-route-policy> <name>`
5. Check policy to XML translation (crafting)*:
   `show support omp peer peer-ip <system-ip>`

The same steps as for control policies

Data and AAR Policies Troubleshooting Workflow (2)

From WAN Edge perspective, ensure policy processing:

1. Check policy assignment on WAN Edge
   - for localized policies, part of a template:
     `show sdwan running-config "policy"`
   - for AAR, data policies and cFlow template received via OMP, volatile RIB:
     `show sdwan policy from-vsmart`
2. Ensure correct next-hop and egress interface selected according to a policy*:

```
show sdwan policy <service-path|tunnel-path> vpn <name>
interface <name> source-ip <ip-addr> dest-ip <ip-addr>
protocol <id> src/dst-port <number> app <name> [all]
```

* can be used also for centralized control policies or just routing verification

Data and AAR Policies Troubleshooting Workflow (3)

3. Ensure correct policy match occurs from WAN Edge perspective:

a) Configure policy counters:

```
action [accept|drop]
count <counter name>
```

- To display counters on the WAN Edge router, depends on type of policy:

```
show sdwan policy <app-route-policy-filter|data-policy-filter|access-list-counters>
```

b) Use logging action in a policy sequence (logs first packet in the flow only)

```
action [accept|drop]
log
```

c) Use policy troubleshooting tools like packet-trace (CLI) or NWPI (vManage UI)

```
debug platform condition ipv4 <address>/<mask> both
debug platform packet-trace packet <number of packets> [fia-trace]
debug platform condition [start|stop]
show platform packet-trace [summary|packet <number>]
```

Data and AAR Policies Troubleshooting Workflow (4)

4. Other useful Data and AAR policies troubleshooting commands:

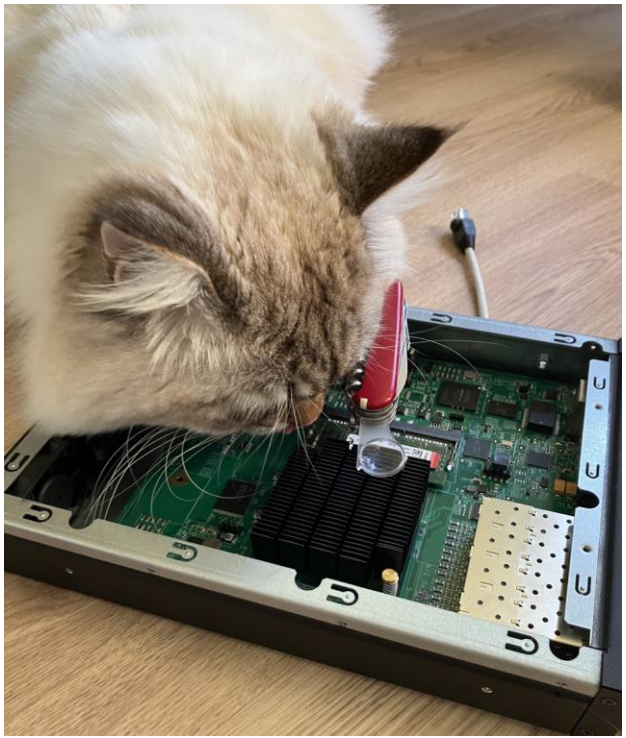
- Verify AAR SLA class statistics:
  `show sdwan app-route stats`

- Check traffic flows symmetry and path taken according to NeFlow data if "**`policy flow-visibility`**" or cFlowd template configured (also useful control polices):
  `show sdwan app-fwd cflowd flows`

- Verify DPI application classification if "**`policy app-visibility`**" enabled:
  `show sdwan app-fwd dpi flows`

Data and AAR Policies Troubleshooting Workflow (5)

5. Proceed to policy programming low-level verification.



*On the photo my cat trying to put himself into my shoes and catch a bug like a mouse ☺

Side note: how to generate synthetic traffic for testing?

Probem to solve: no user at a site to help with a testing

1. CLI command to trigger synthetic traffic, execution will trigger one probe.
2. Probe result will be reported using log. Use **`show logging`** to see it

CLI request syntax:

`request platform software sdwan synthetic-traffic probe vpn-id 1 url www.cisco.com [dscp <code> [dns <address of nameserver>]`

Example:

```
cEdge1#request sdwan synthetic-traffic probe vpn-id 1 url www.cisco.com

*Apr 11 02:05:34.302: %Cisco-SDWAN-Site25-cEdge-1-DBGD-6-INFO-1500002: Synthetic test probe
result for app: Def-Test, url: www.cisco.com, src_intf: GigabitEthernet7, latency: 253, loss:
0%, score: 6, count 1
```

Centralized Data and AAR Policies Troubleshooting:

Commands usage examples

cisco Live!

Data and AAR Policies Troubleshooting Commands

From vSmart perspective the same steps as for control policies (so we won't repeat them here), except additional step 5. Check policy XML translation (crafting):

```
vsmart1# show support omp peer peer-ip 10.0.0.11 | begin "Policy received" | until "Statistics"
        Policy received: Complete
       Forwarding policy len: 632
<data-policy>
  <name>VPN_1_NAT</name>
  <vpn-list>
    <name>VPN_1</name>
    <sequence>
      <seq-value>1</seq-value>
      <match>
        <source-data-prefix-list>LAN</source-data-prefix-list>
      </match>
      <action>
        <action-value>accept</action-value>
        <nat>
          <use-vpn>0</use-vpn>
        </nat>
      </action>
    </sequence>
  </vpn-list>
<direction>from-service</direction></data-policy><lists><vpn-list>
  <name>VPN_1</name>
  <vpn>
    <id>1</id>
  </vpn>
</vpn-list>
<data-prefix-list>
  <name>LAN</name>
  <ip-prefix>
    <ip>10.10.10.0/24</ip>
  </ip-prefix>
</data-prefix-list>
</lists>
        Statistics:
```

Data and AAR Policies Troubleshooting Commands (1)

From WAN Edge perspective

1. Check policy assignment on WAN Edge:

```
cE1_BR1#show sdwan policy from-vsmart
from-vsmart data-policy VPN_1_NAT
 direction from-service
 vpn-list VPN_1
  sequence 1
   match
    source-data-prefix-list LAN
   action accept
    nat use-vpn 0
    no nat fallback
  default-action drop
from-vsmart lists vpn-list VPN_1
 vpn 1
from-vsmart lists data-prefix-list LAN
 ip-prefix 10.10.10.0/24
```

Data and AAR Policies Troubleshooting Commands (2)

From WAN Edge perspective

2. Ensure correct egress interface and next-hop selected as a result of a policy:

```
cE1_BR1#show sdwan policy service-path vpn 1 interface GigabitEthernet 4 source-ip 10.10.10.10 dest-ip 1.1.1.1
protocol 17 dest-port 53
Next Hop: Remote
  Remote IP: 192.168.10.1, Interface GigabitEthernet3 Index: 9
```

Example of a problematic state:

```
cE1_BR1#show sdwan policy service-path vpn 1 interface GigabitEthernet 4 source-ip 10.10.1.10 dest-ip 1.1.1.1
protocol 17 dest-port 53 app dns
Next Hop: Blackhole
```

Data and AAR Policies Troubleshooting Commands (3a)

From WAN Edge perspective

3a. Ensure that correct policy match occurs from WAN Edge perspective

Using counters in a policy:

```
cE1_BR1#show sdwan policy from-vsmart
from-vsmart data-policy _VPN_1_TEST_COUNT
 direction from-service
 vpn-list VPN_1
  sequence 1
   match
    source-ip 0.0.0.0/0
   action accept
    count COUNT_PKTS
  default-action accept
from-vsmart lists vpn-list VPN_1
 vpn 1
```

```
cE1_BR1#ping vrf 1 192.168.4.196
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to
192.168.4.196, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip
min/avg/max = 1/1/1 ms
```

```
cE1_BR1#show sdwan policy data-policy-filter
data-policy-filter _VPN_1_TEST_COUNT
 data-policy-vpnlist VPN_1
  data-policy-counter COUNT_PKTS
   packets 5
   bytes   500
  data-policy-counter default_action_count
   packets 76652
   bytes   9023632
```

Data and AAR Policies Troubleshooting Commands (3b)

From WAN Edge perspective

3b. Ensure that correct policy match occurs from WAN Edge perspective

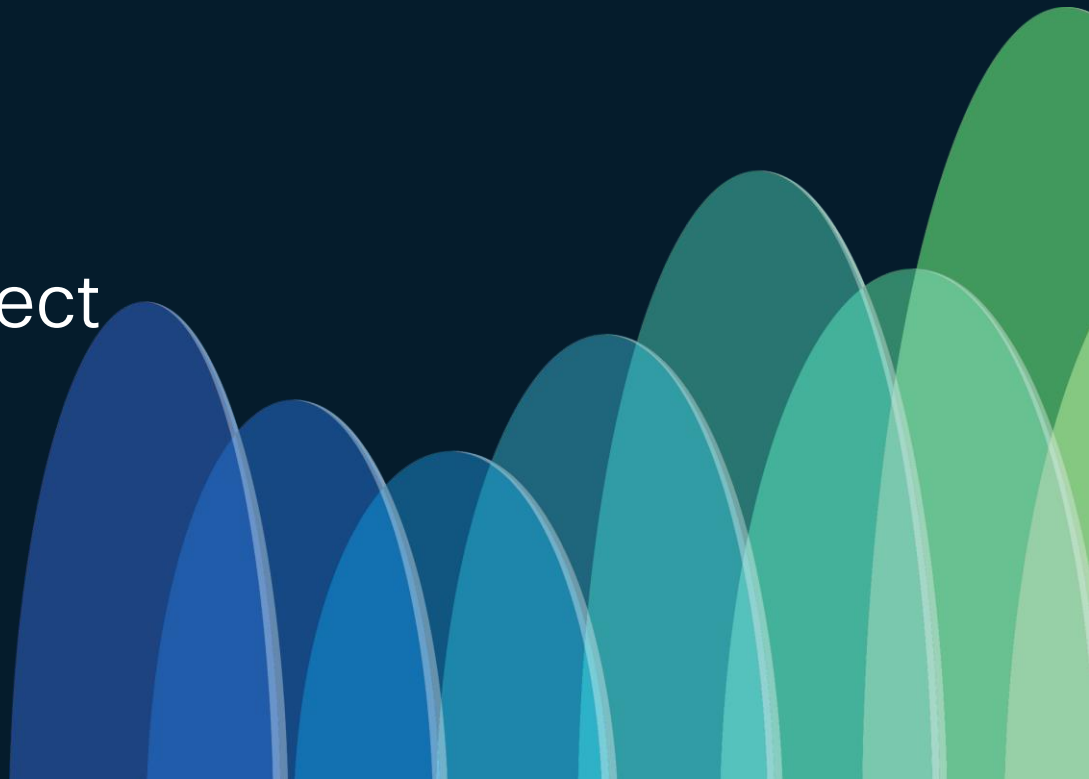
Using logging in a policy:

```
cE1_BR1#show sdwan policy from-vsmart
from-vsmart data-policy _VPN_1_TEST_LOG
 direction from-service
 vpn-list VPN_1
  sequence 1
   match
    source-ip 0.0.0.0/0
   action accept
    log
  default-action accept
from-vsmart lists vpn-list VPN_1
 vpn 1
```

```
cE1_BR1#ssh -l admin -vrf 1 192.168.5.197
Password:
Welcome to Ubuntu 20.04.6 LTS (GNU/Linux 5.4.0-
182-generic x86_64)
Last login: Thu Feb  6 14:33:46 2025 from
192.168.5.197
admin@testsrv:~$ exit
logout

[Connection to 192.168.5.197 closed by foreign
host]
```

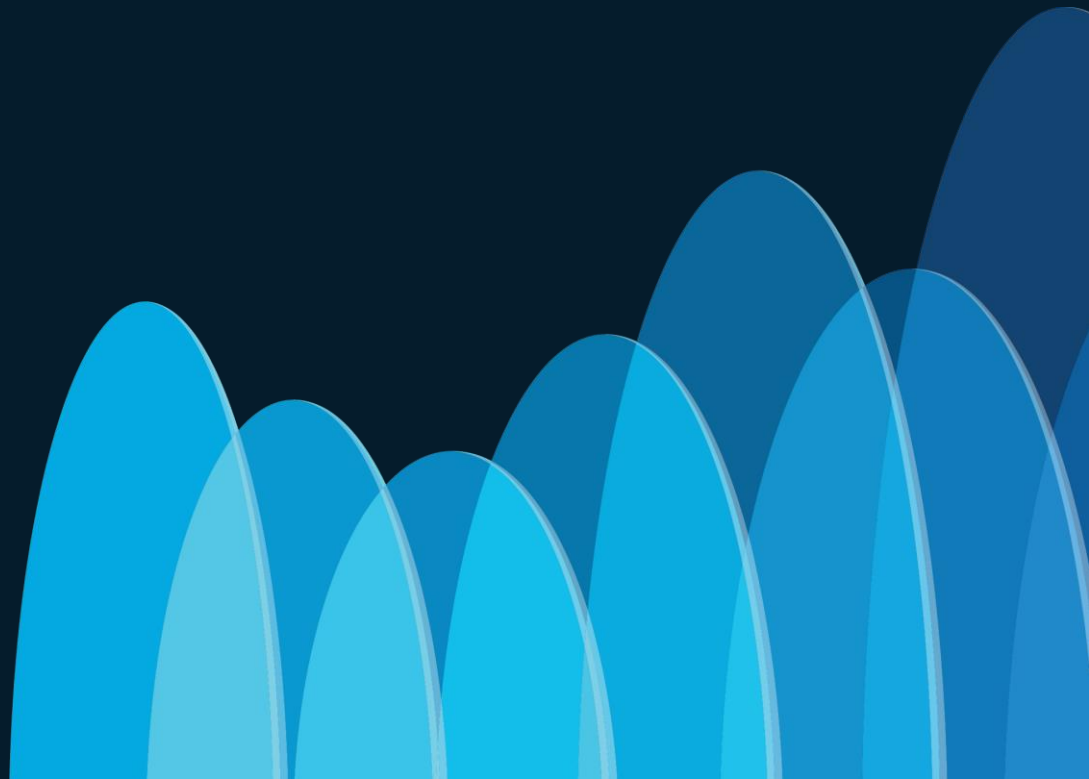
```
cE1_BR1#show logging | include dst: 192.168.5.197
Feb  6 14:48:16.902: %SDWAN-5-FPMD : FLOW LOG device-vpn: 1 tenant-vpn: 1 src: 192.168.4.11/16516 dst: 192.168.5.197/22
proto: 6 tos: 192 direction: from-service, policy: _VPN_1_TEST_LOG-VPN_1, sequence: 1, Result: accept Pkt count: 1
bytes: 52 Ingress-Intf: cpu Egress-intf: Unknown Tenant: Not-Applicable
```

SD-WAN Policy troubleshooting tools
(step 3c. ensure correct policy match and actions)



cisco Live!

Packet-trace
a.k.a
FIA-trace



cisco Live!

Enabling packet-trace

Set debug conditions (match filter) and enable packet-trace:

```
cEdge1#debug platform condition <ipv4|ipv6|mac|mpls> <address/mask | access-list name> both
cEdge1#debug platform packet-trace packet <number of packets> [fia-trace]
cEdge1#debug platform condition start
```

Optionally, dump a packet data in a hex format:

```
cEdge1#debug platform packet-trace copy packet both size <…>
```

If you want to trace only internally dropped packets, check QFP statistics first:

```
cE1_BR1#show platform hardware qfp active statistics drop detail
Last clearing of QFP drops statistics : never

-------------------------------------------------------------------------------
   ID  Global Drop Stats                          Packets                Octets
-------------------------------------------------------------------------------
   62  IpTtlExceeded                                 28                  1748
   56  IpsecInput                                    14                  2402
   19  Ipv4NoRoute                                 4909                786205
  483  SdwanDataPolicyDrop                          650                 78230
  479  SdwanImplicitAclDrop                      261280              44905782
```

Then enable trace only for the specific drop code ID:

```
cEdge1#debug platform packet-trace drop code <id>
```

Using packet-trace (1)

You can also check overall statistics and number of packets captured:

```
cEdge1# show platform packet-trace statistics
Packets Summary
  Matched  1165
  Traced   1024
Packets Received
  Ingress  1085
  Inject   80
    Count       Code  Cause
    80          3     QFP IPv4/v6 nexthop lookup
Packets Processed
  Forward  928
  Punt     5
  Drop     34
  Consume  237
```

To stop packet-trace and clear all conditions (filters):

```
cEdge1# debug platform condition stop
cEdge1# clear platform condition all
```

Using packet-trace (2)

To show captured packets summary:

```
cEdge1#show platform packet-trace summary
Pkt    Input                     Output                    State  Reason
0      Gi2                       Gi3                       FWD
1      Tu3                       Gi2                       FWD
2      INJ.3                     Gi2                       FWD
3      internal0/0/recycle:0     Gi2                       FWD
4      Gi2                       Tu3                       DROP   525 (NoStatsUpdate)
5      internal0/0/recycle:0     Gi2                       FWD
```

BFD return packet dropped, expected

Details of specific packet:

```
cEdge1# show platform packet-trace packet <packet number>
```

Packet-trace output example (1)

```
cEdge1#show platform packet-trace packet 0
Packet: 0           CBUG ID: 35949496
Summary
  Input     : GigabitEthernet2
  Output    : GigabitEthernet3
  State     : FWD
  Timestamp
    Start   : 1214211941024994 ns (02/24/2020 11:03:14.435466 UTC)
    Stop    : 1214211941530105 ns (02/24/2020 11:03:14.435971 UTC)
Path Trace
  Feature: IPV4(Input)
    Input       : GigabitEthernet2
    Output      : <unknown>
    Source      : 192.168.11.254
    Destination : 192.168.17.254
    Protocol    : 1 (ICMP)
<remove>
  Feature: SDWAN ACL IN
    Interface    : GigabitEthernet2
    CG           : 3
    Seq          : 21
    Policy Flags : 0x100
    Action : SET_FWD_CLASS 3 Prec3
  Feature: SDWAN_ACL_IN
    Entry       : Input - 0x81845740
    Input       : GigabitEthernet2
    Output      : <unknown>
    Lapsed time : 815733 ns
<removed>
```

Local policy in action here

SD-WAN ACL matches flow and assign to QoS class ”Prec3”

Packet-trace output example (2)

```
<skipped>
Feature: NBAR
    Packet number in flow: N/A
    Classification state: Final
    Classification name: ping
<skipped>
Feature: SDWAN App Route Policy
    VRF           : 1
    CG            : 1
    Seq           : 65535
    SLA           : __all_tunnels__ (0)
    Policy Flags : 0x2
    SLA Strict    : No
    Preferred Color : 0x0 none
<removed>
Feature: SDWAN OCE
    Hash Value  : 0xaf6f0c4e
    Encap       : ipsec
    SLA         : 0
    SDWAN VPN   : 1
    SDWAN Proto : IPV4
    Out Label   : 1001
    Local Color : biz-internet
    Remote Color: biz-internet
    FTM Tunnel ID:15
    SDWAN Session Info
      SRC IP      : 172.16.11.254
      SRC Port    : 12346
      DST IP      : 172.16.17.254
      DST Port    : 12346
      Remote System IP : 172.16.255.17
```

NBAR classification is completed
Application is recognized

This flow does not match any app-route policies, so it's load balanced to all available tunnels

Forwarding decision

Packet-trace output example (3)

```
<removed>
 Feature: SDWAN QoS Output
    Fwd Class     : 3
    QoS Queue     : 3
    DSCP Rewrite : No
    CoS  Rewrite : No
<removed>
 Feature: IPSec
    Result     : IPSEC_RESULT_SA
    Action     : ENCRYPT
    SA Handle : 45
    Peer Addr : 172.16.17.254
    Local Addr: 172.16.11.254
<removed>
 Feature: MARMOT_SPA_D_TRANSMIT_PKT
    Entry        : Output - 0x817f19f4
    Input        : GigabitEthernet2
    Output       : GigabitEthernet3
    Lapsed time : 217840 ns
```

QoS queueing in queue3

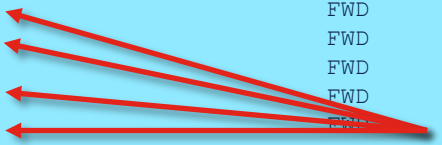
Encrypting

Packet is transmitted on an interface

Troubleshooting with packet-trace: example (1)

Green: Service interface [VPN 1]
Blue: “mpls” interface [VPN 0]
Red: “biz-internet” interface [VPN 0]

```
cEdge1#show platform packet-trace summary
Pkt    Input                Output               State  Reason
0      Gi2                  Gi1                  FWD
1      Gi2                  Gi1                  FWD
2      Tu1                  Gi2                  FWD
3      Gi2                  Gi1                  FWD
4      Gi2                  Gi3                  FWD
5      Tu3                  Gi2                  FWD
6      Tu3                  Gi2                  FWD
```



Why traffic follows Gig1 at the beginning and then switched to Gig3 ?

Troubleshooting with packet-trace: example (2)

First let's check the very first packet in the flow:

```
cEdge1#show platform packet-trace packet 0
Packet: 0           CBUG ID: 423
Summary
  Input     : GigabitEthernet2
  Output    : GigabitEthernet1
  State     : FWD
<removed>
 Feature: NBAR
    Packet number in flow: 1
    Classification state: Not final
<removed>
 Feature: SDWAN App Route Policy
    VRF          : 1
    Seq          : 1
    SLA          : __all_tunnels__ (0)
    Policy Flags : 0x0
    SLA Strict   : No
    Preferred Color : 0x0 none
```

NBAR application recognition does not have enough information to recognize the application yet

SDWAN App route policy will consider all available tunnels

Troubleshooting with packet-trace: example(3)

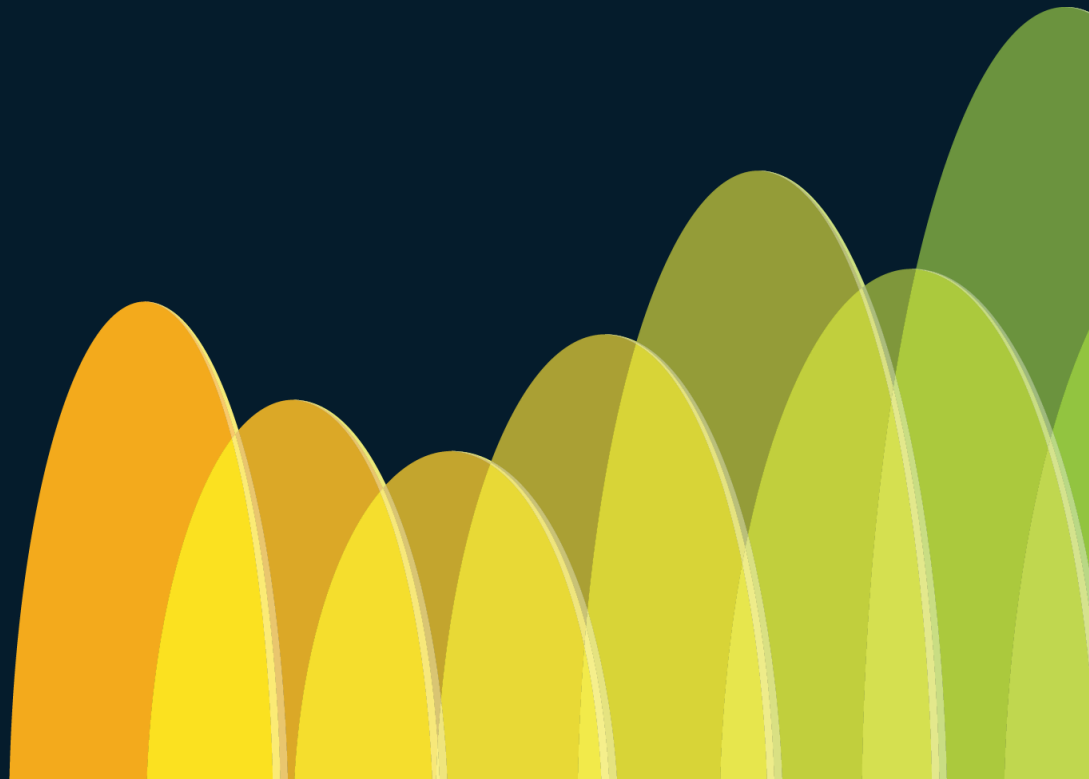
Then let's check some later packet from the service side:

```
cEdge1#show platform packet-trace packet 4
Packet: 4           CBUG ID: 423
Summary
  Input     : GigabitEthernet2
  Output    : GigabitEthernet3
  State     : FWD
<removed>
 Feature: NBAR
    Packet number in flow: 5
    Classification state: Final
    Classification name: ssh
    Classification ID: [IANA-L4:22]
<removed>
Feature: SDWAN App Route Policy
    VRF           : 1
    Seq           : 10
    SLA           : TEST1 (1)
    Policy Flags : 0x1
    SLA Strict    : Yes
    Preferred Color : 0x10 biz-internet
    Tunnel Match Reason : MATCHED_SLA_AND_PREF_COLOR
```

NBAR application recognition has discovered SSH application finally

SDWAN App route policy will optimize the flow towards biz-internet tunnel

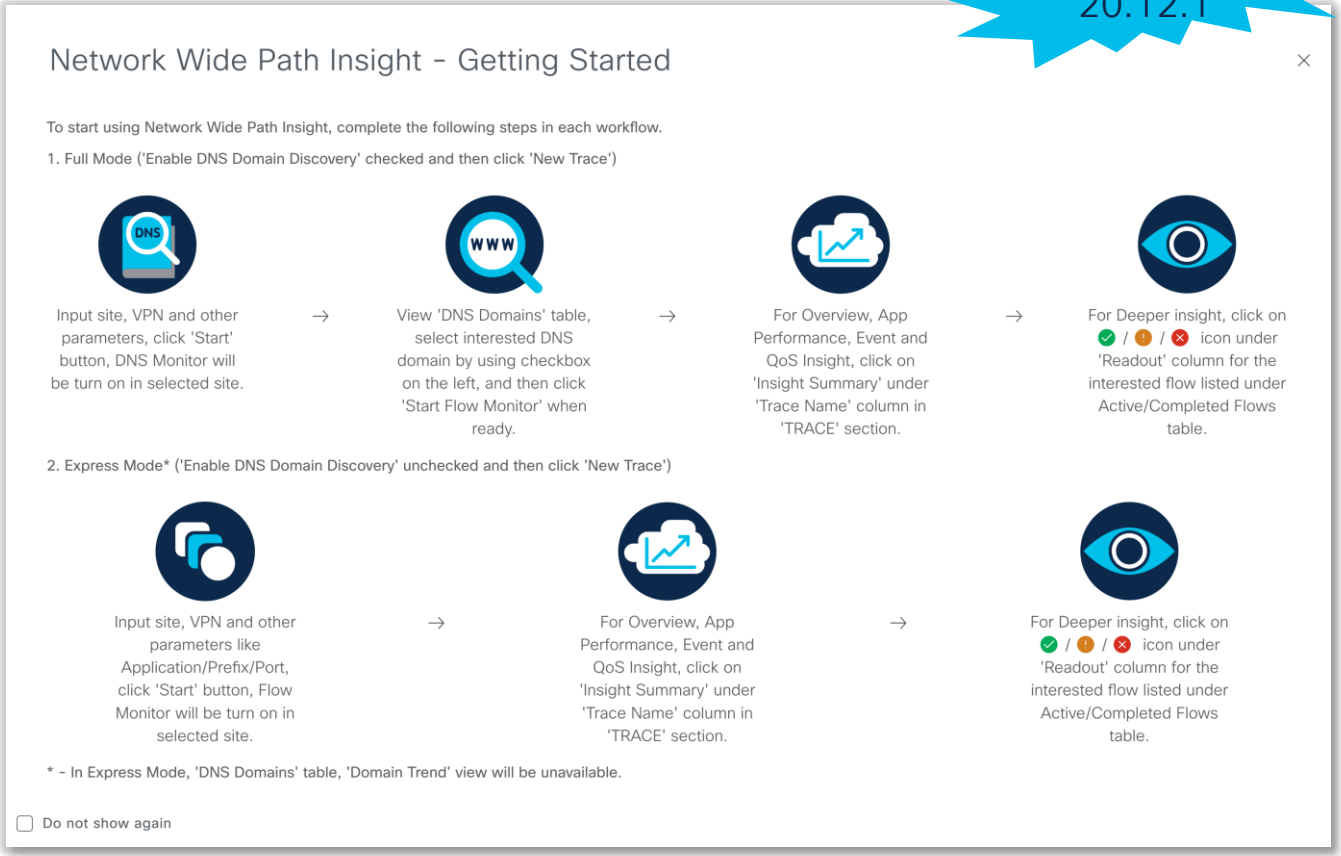
Network Wide Path Insight (NWPI)



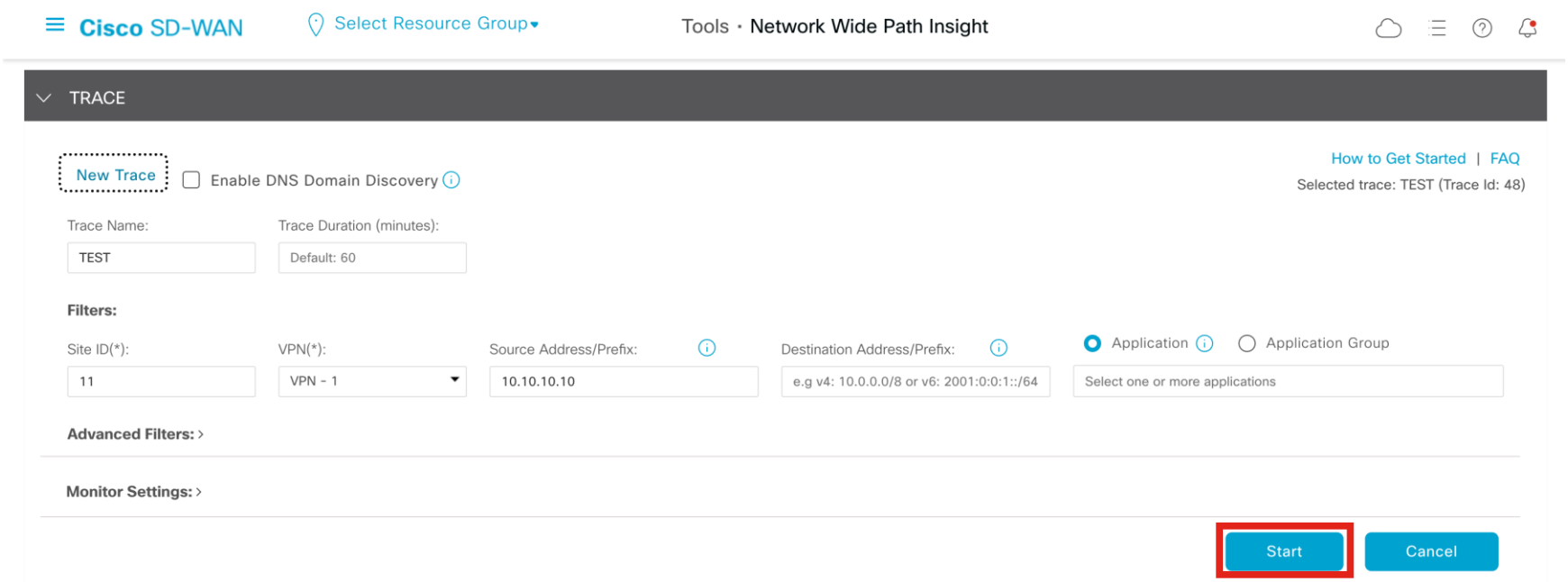
cisco Live!

How to Get Started with NWPI

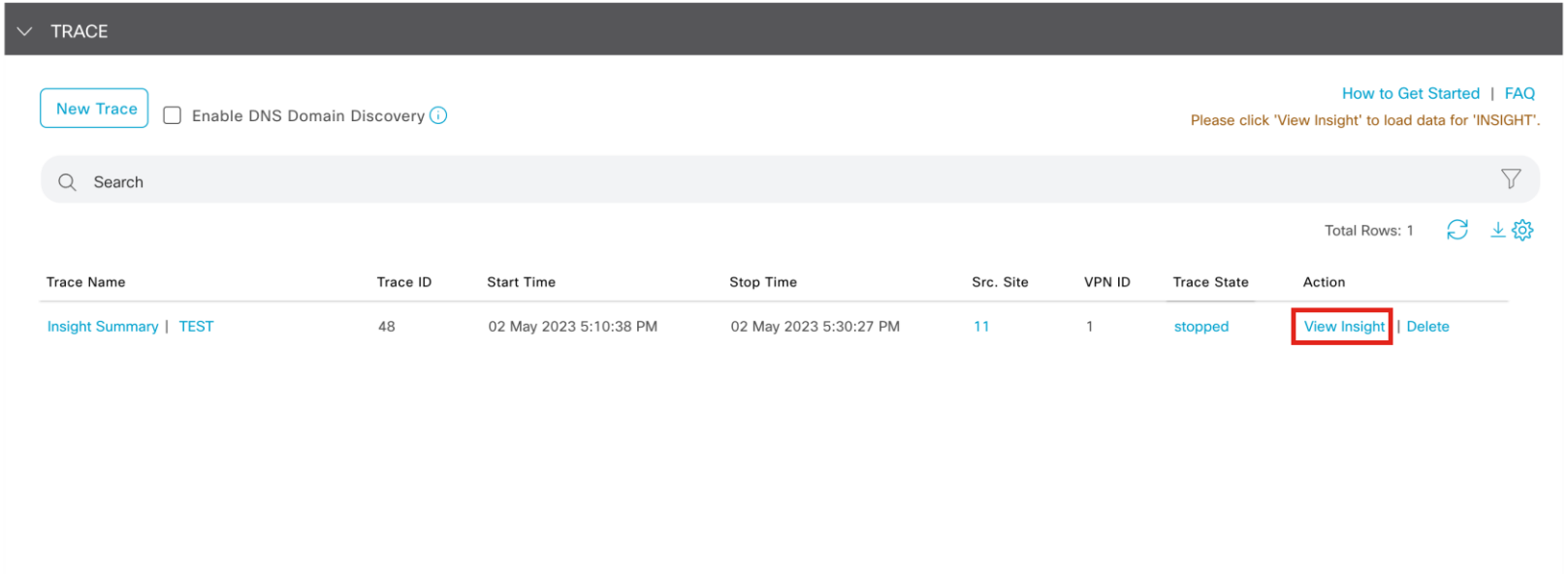
vManage 20.12.1



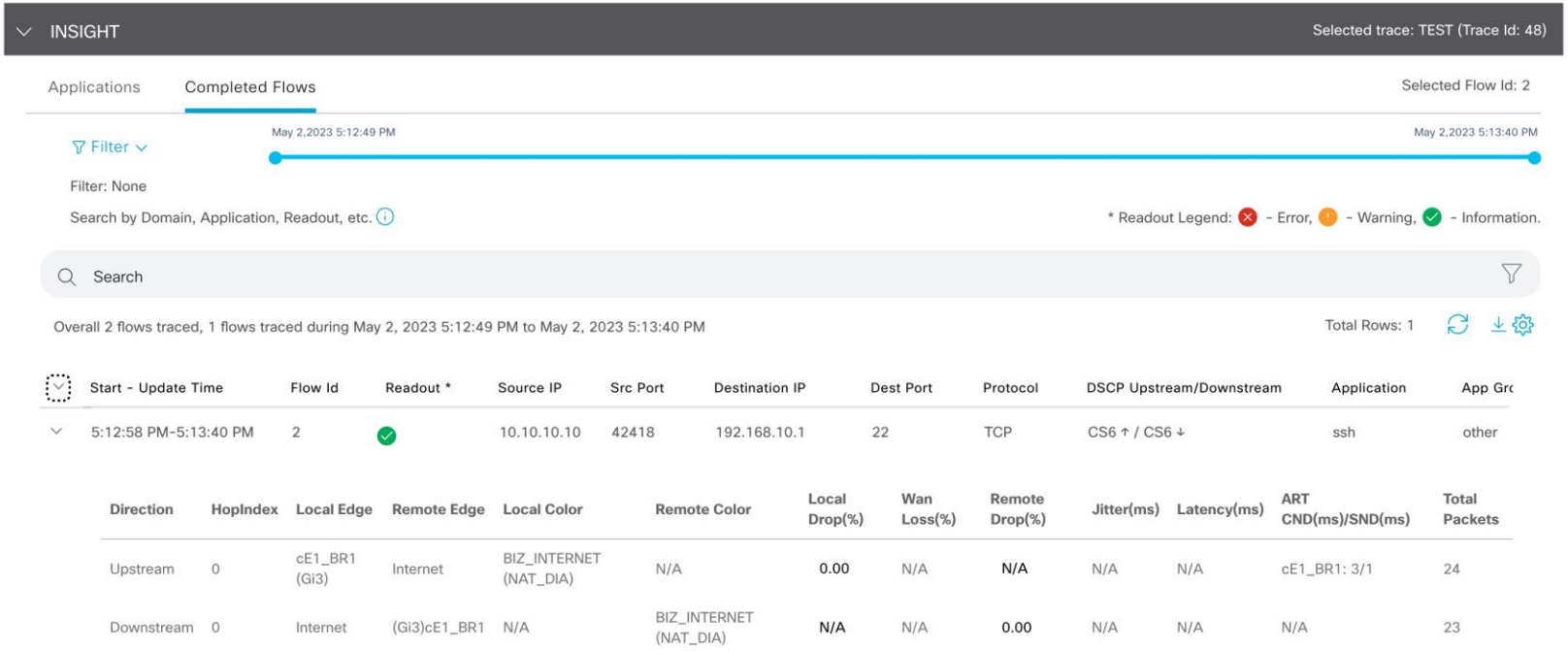
Enabling NWPI



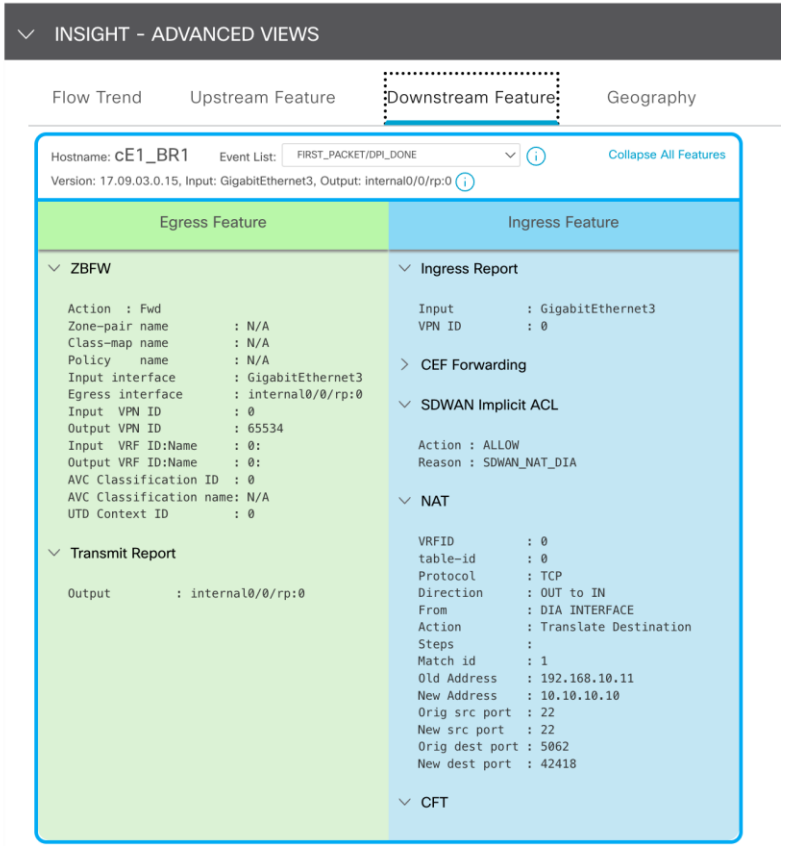
Using NWPI



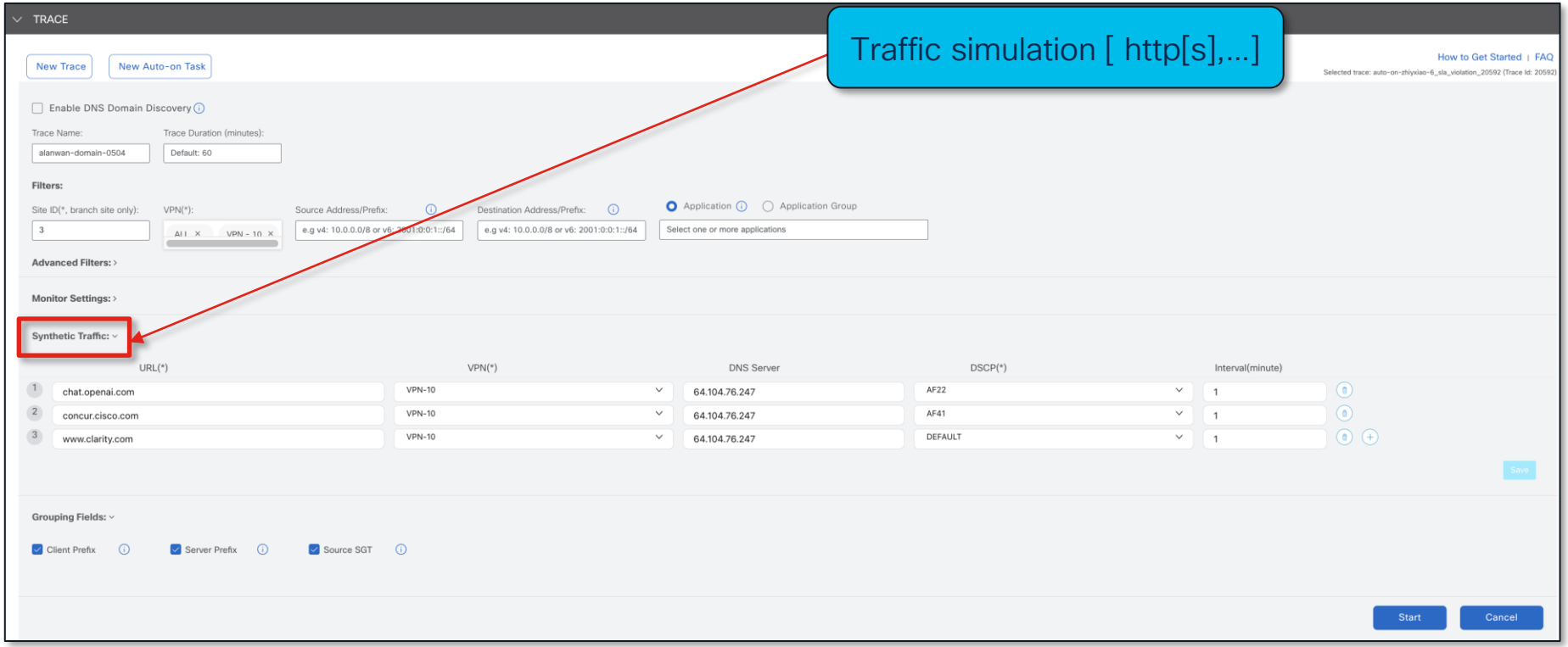
Using NWPI (2)



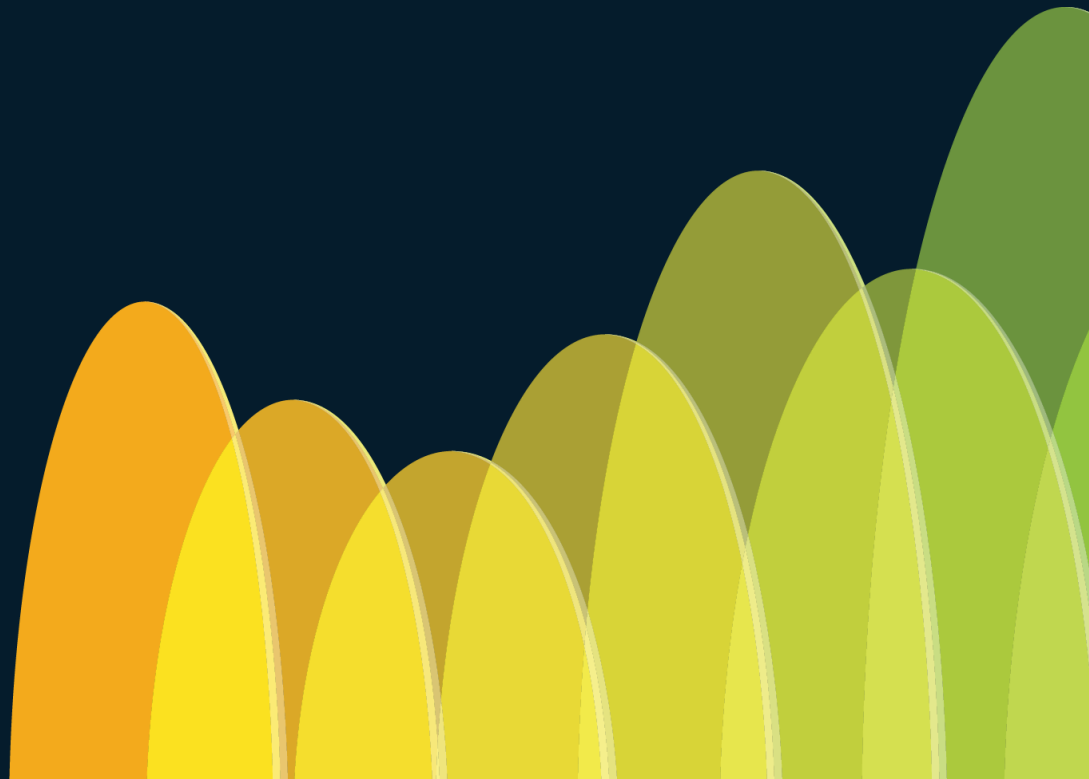
Using NWPI (3)



What's new in 20.12/17.12: Synthetic Traffic



Other useful commands for AAR and Data Policies troubleshooting (step 4)



Other Useful Commands for AAR troubleshooting (1)

```
cE1_BR1#show sdwan app-route stats remote-color biz-internet remote-system-ip 169.254.206.37 summary
Generating output, this might take time, please wait ...
app-route statistics 192.168.10.11 192.168.10.37 ipsec 12346 12406
remote-system-ip 169.254.206.37
local-color      biz-internet
remote-color     public-internet
mean-loss        54
mean-latency     43
mean-jitter      21
sla-class-index  0
                                                                  IPV6 TX  IPV6 RX
       TOTAL           AVERAGE  AVERAGE  TX DATA  RX DATA       DATA     DATA
INDEX  PACKETS  LOSS   LATENCY  JITTER   PKTS     PKTS          PKTS     PKTS
-------------------------------------------------------------------------------
0      132      66     43       19       3380     0             0        0
1      132      76     45       23       3277     0             0        0
2      133      62     38       11       3958     1             0        0
3      134      75     43       20       5233     1             0        0
4      133      82     51       35       4003     0             0        0
5      132      74     43       20       4070     1             0        0
```

Other Useful Commands for AAR troubleshooting (2)

```
cE1_BR1#show sdwan app-fwd cflowd flows vpn 4
Generating output, this might take time, please
wait ...
app-fwd cflowd flows vpn 4 src-ip 192.168.5.197
dest-ip 192.168.4.196 src-port 22 dest-port 37748
dscp 4 ip-proto 6
 tcp-cntrl-bits           24
 icmp-opcode              0
 total-pkts               6
 total-bytes              2064
 start-time               "Fri Dec 22 15:35:11
2023"
 egress-intf-name         GigabitEthernet4
 ingress-intf-name        GigabitEthernet3
 application              unknown
 family                   network-service
 drop-cause               "No Drop"
 drop-octets              0
 drop-packets             0
 sla-not-met              0
 color-not-met            0
 queue-id                 2
 initiator                2
 tos                      0
 dscp-output              0
 sampler-id               0
 fec-d-pkts               0
 fec-r-pkts               0
 pkt-dup-d-pkts-orig      0
 pkt-dup-d-pkts-dup       0
 pkt-dup-r-pkts           0
 pkt-cxp-d-pkts           0
 category                 0
 service-area             0
 cxp-path-type            0
 region-id                0
 ssl-read-bytes           0
 ssl-written-bytes        0
 ssl-en-read-bytes        0
 ssl-en-written-bytes     0
 ssl-de-read-bytes        0
 ssl-de-written-bytes     0
 ssl-service-type         0
 ssl-traffic-type         0
 ssl-policy-action        0
 appqoe-action            0
 appqoe-sn-ip             0.0.0.0
 appqoe-pass-reason       0
 appqoe-dre-input-bytes   0
 appqoe-dre-input-packets 0
 appqoe-flags             0
```

Other Useful Commands for AAR troubleshooting (3)

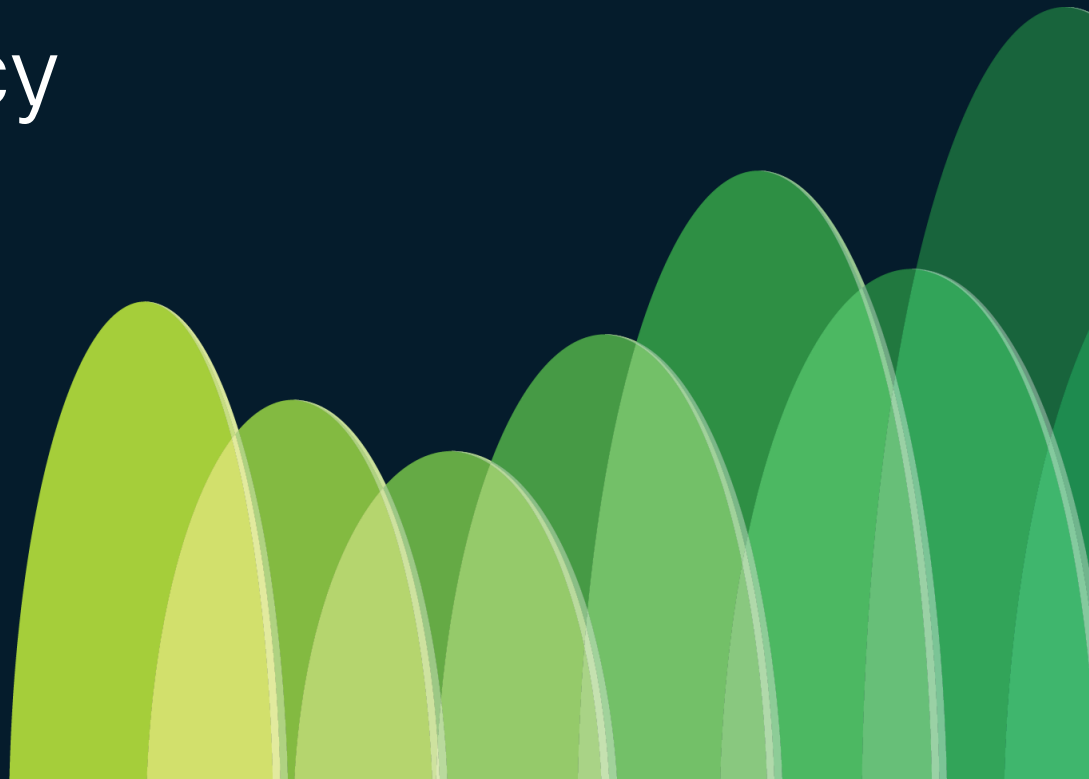
```
cE1_BR1#show sdwan app-fwd dpi flows vpn 4
Generating output, this might take time, please wait
app-fwd cflowd flows vpn 4 src-ip 192.168.5.197 dest-ip
192.168.4.196 src-port 36470 dest-port 22 dscp 10 ip-
proto 6
 tcp-cntrl-bits           24
 icmp-opcode              0
 total-pkts               61
 total-bytes              4080
 start-time               "Tue Jan 16 15:26:56 2024"
 egress-intf-name         GigabitEthernet4
 ingress-intf-name        GigabitEthernet3
 application              ssh
 family                   terminal
 drop-cause               "No Drop"
 drop-octets              0
 drop-packets             0
 sla-not-met              0
 color-not-met            0
 queue-id                 2
 initiator                1
 tos                      40
 dscp-output              10
 sampler-id               0
 fec-d-pkts               0
 fec-r-pkts               0
 pkt-dup-d-pkts-orig      0
 pkt-dup-d-pkts-dup       0
 pkt-dup-r-pkts           0
 pkt-cxp-d-pkts           0
 category                 0
 service-area             0
 cxp-path-type            0
 region-id                0
 ssl-read-bytes           0
 ssl-written-bytes        0
 ssl-en-read-bytes        0
 ssl-en-written-bytes     0
 ssl-de-read-bytes        0
 ssl-de-written-bytes     0
 ssl-service-type         0
 ssl-traffic-type         0
 ssl-policy-action        0
 appqoe-action            0
 appqoe-sn-ip             0.0.0.0
 appqoe-pass-reason       0
 appqoe-dre-input-bytes   0
 appqoe-dre-input-packets 0
 appqoe-flags             0
```

Policy programming low-level verification (step 5)

Down the rabbit hole. Are you ready?

*Not a stock photo. A rabbit hole near Cisco's Brussles office in Belgium

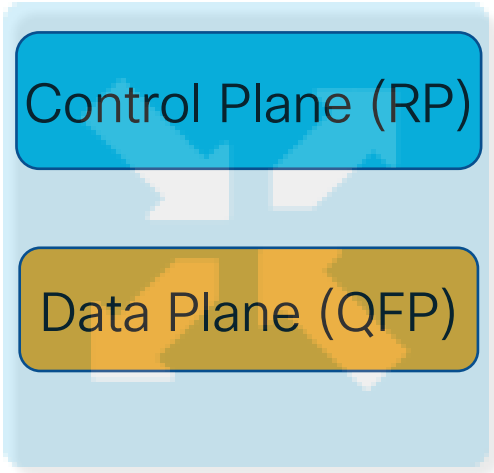
Centralized Policy Installation Workflow from IOS-XE Perspective

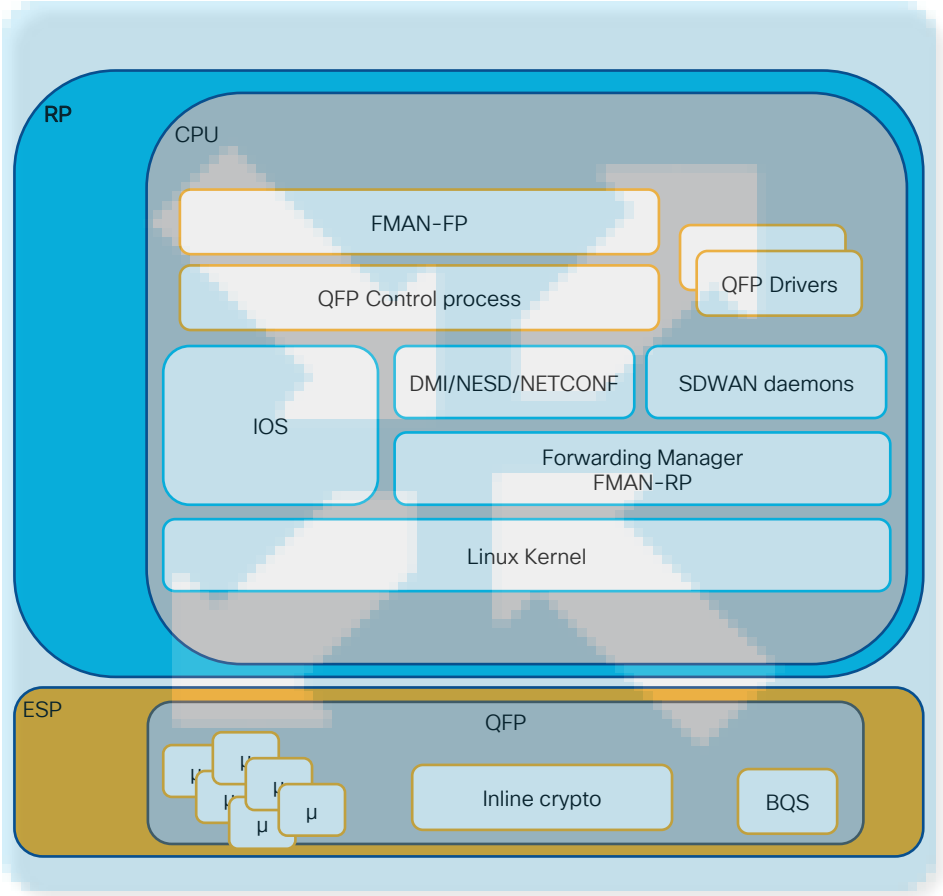


cisco Live!

WAN Edge running IOS-XE “cEdge”

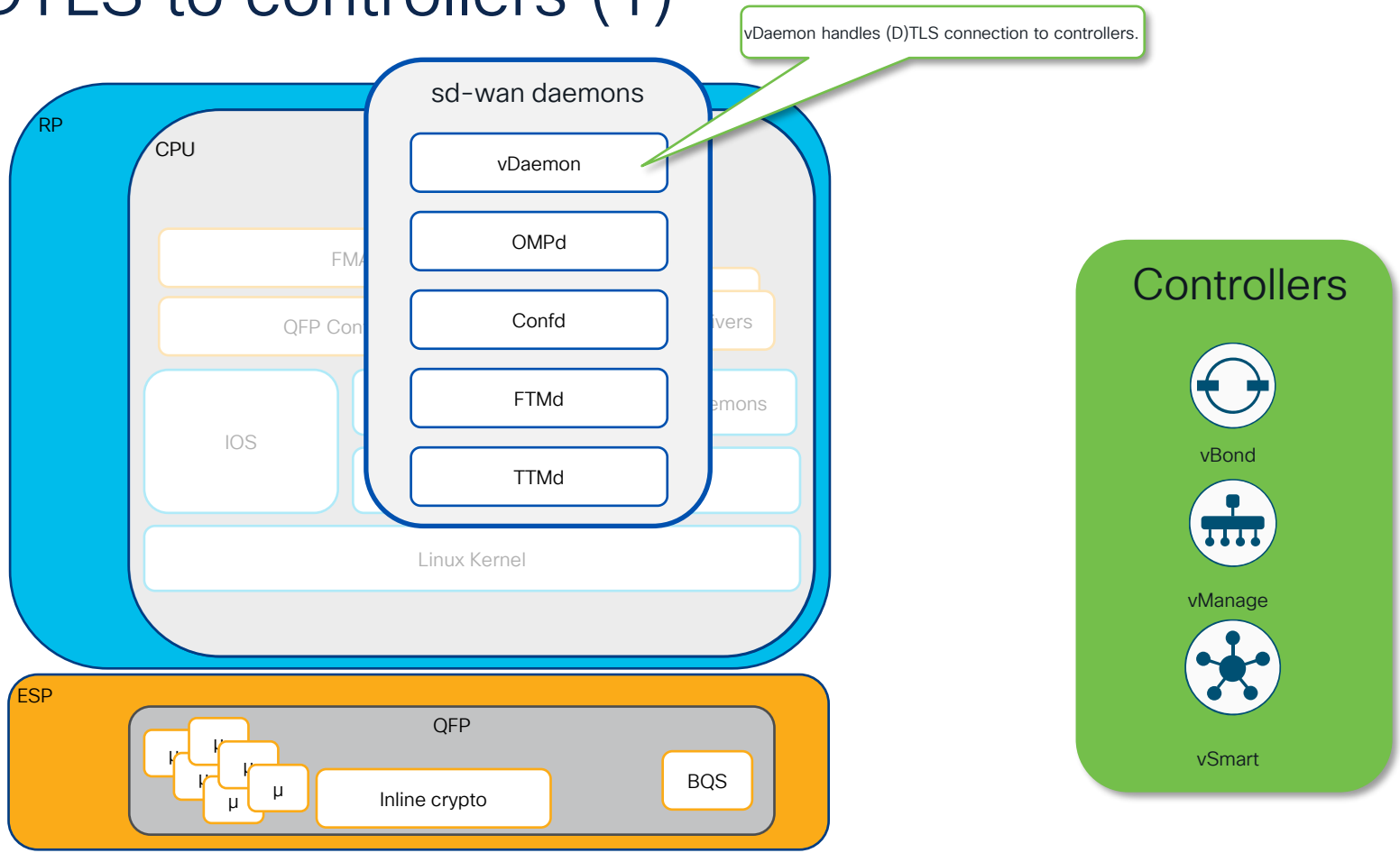
Generalized Software Architecture



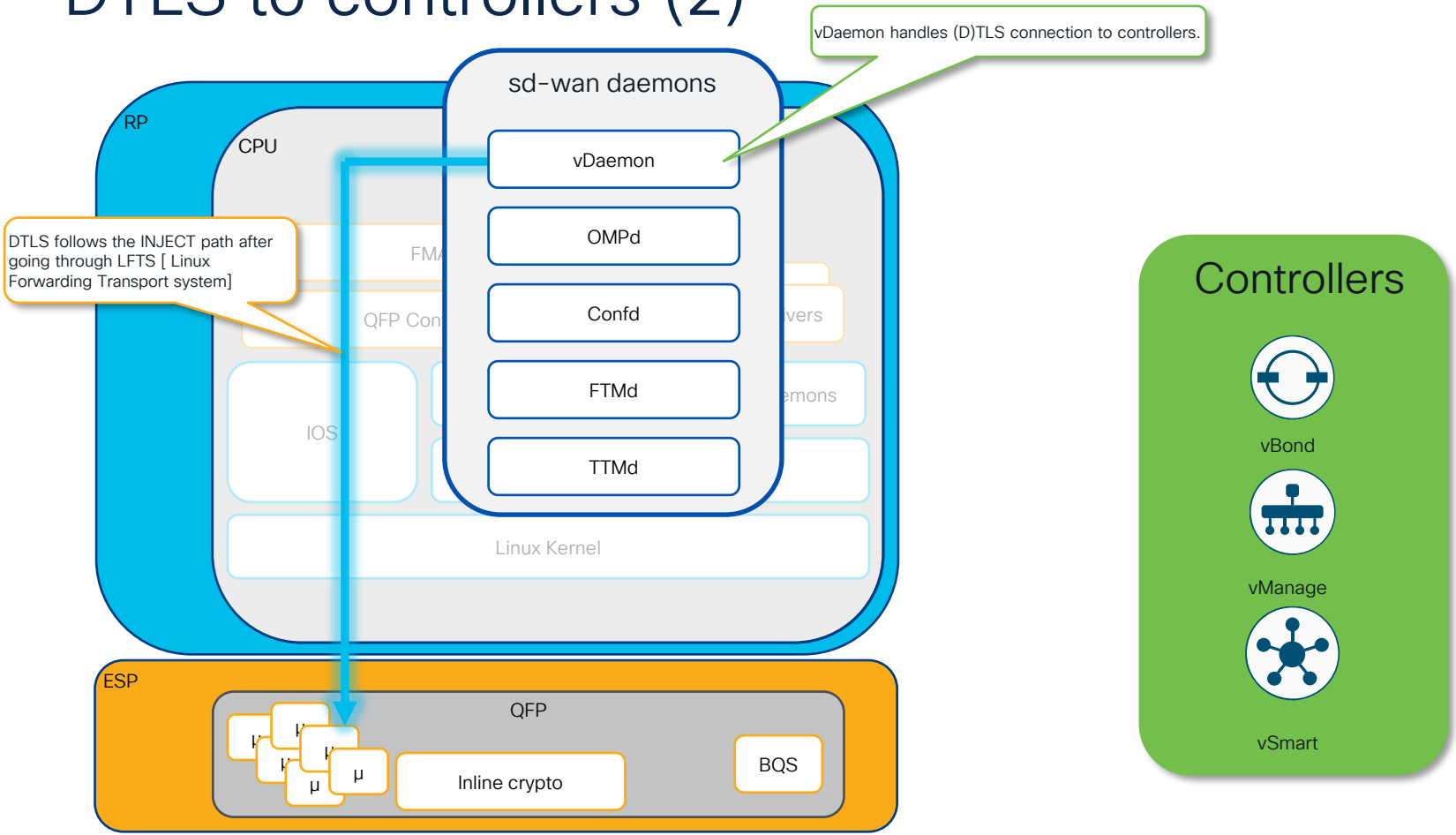


- All platforms share similar architecture
- Key differences:
  - the **data plane (QFP)** is either dedicated CPU/linecard or a Linux software process
  - Crypto implemented either inline or via external crypto accelerator/hardware/ASIC

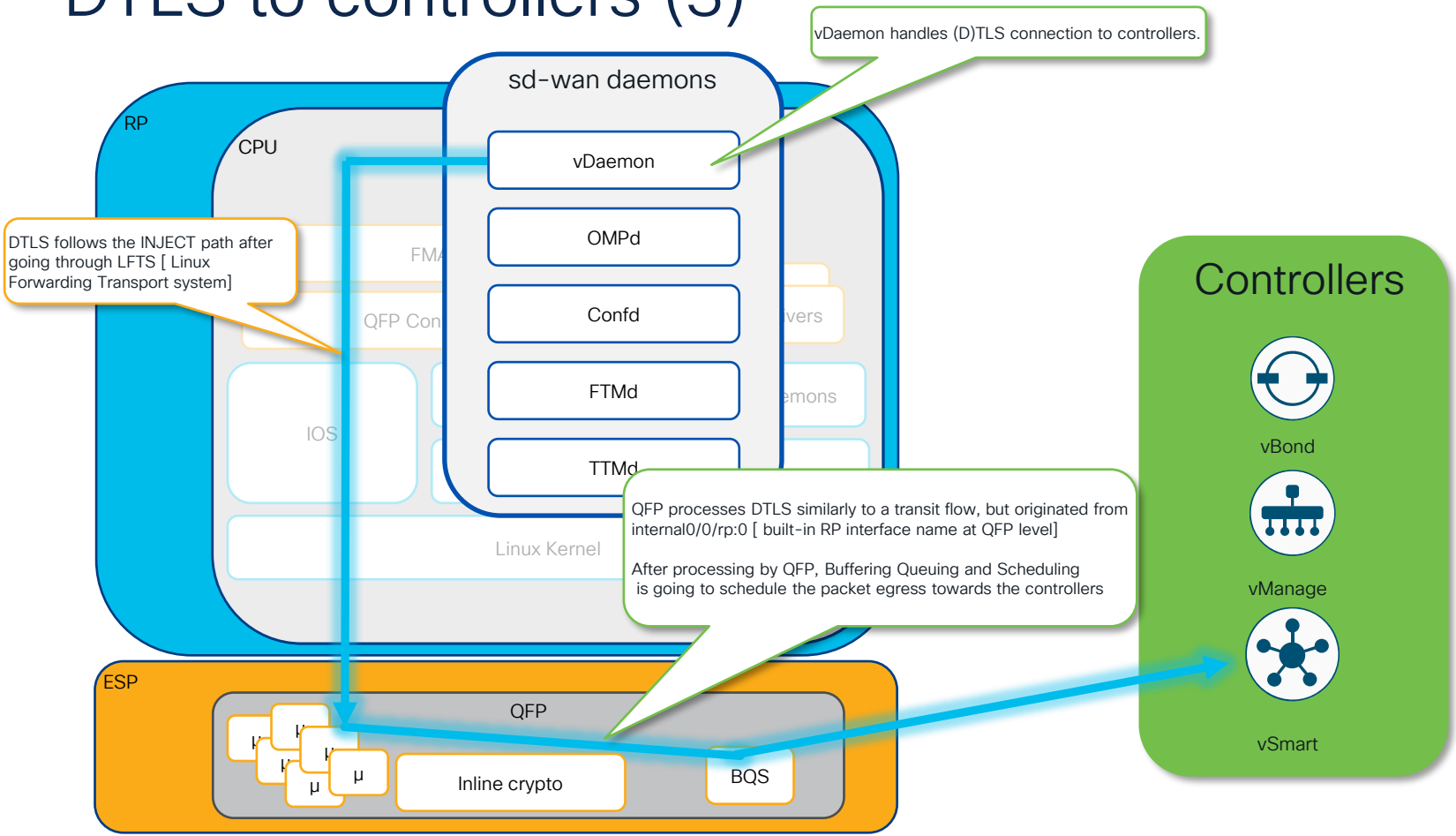
DTLS to controllers (1)



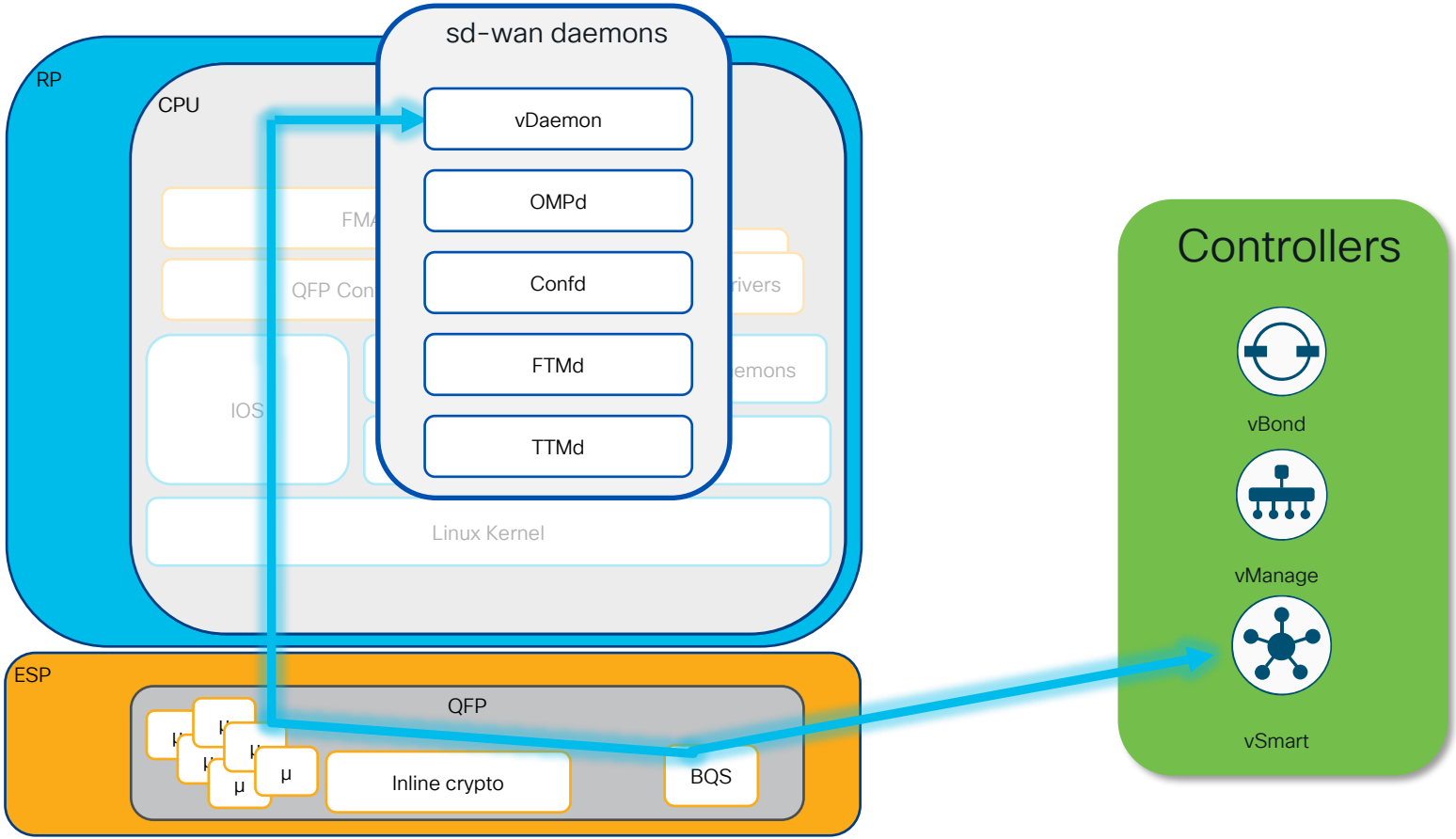
DTLS to controllers (2)



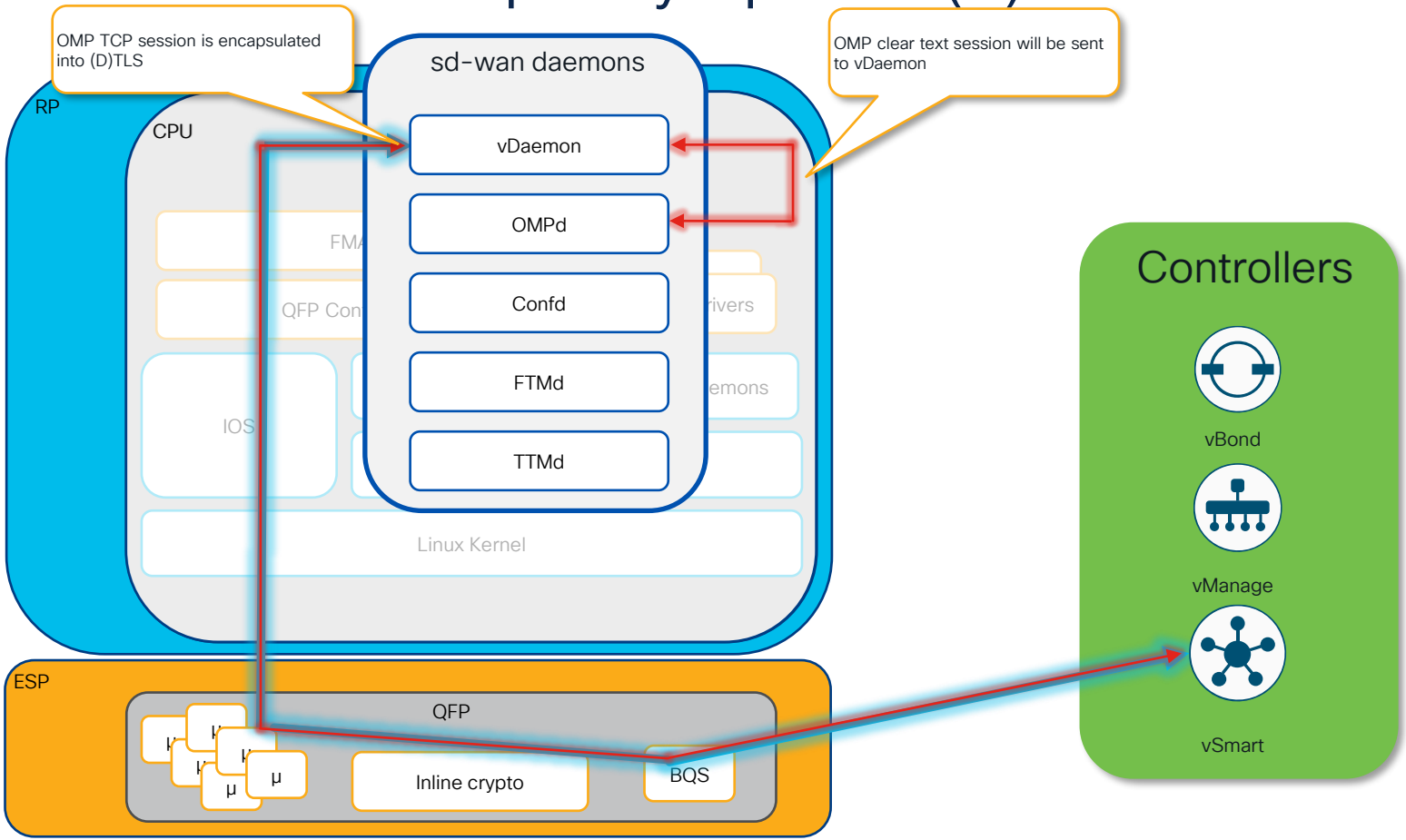
DTLS to controllers (3)



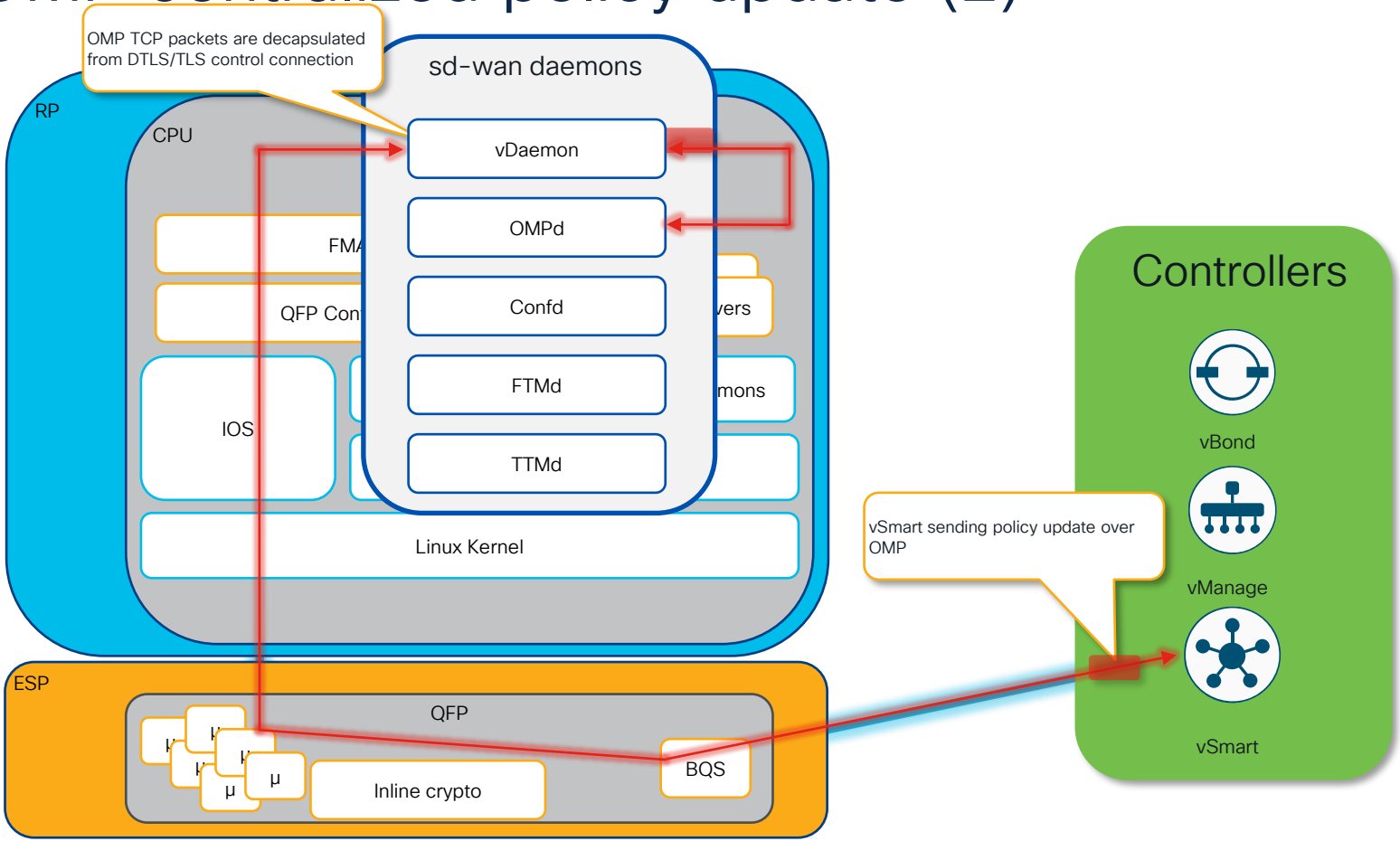
DTLS to controllers (4)



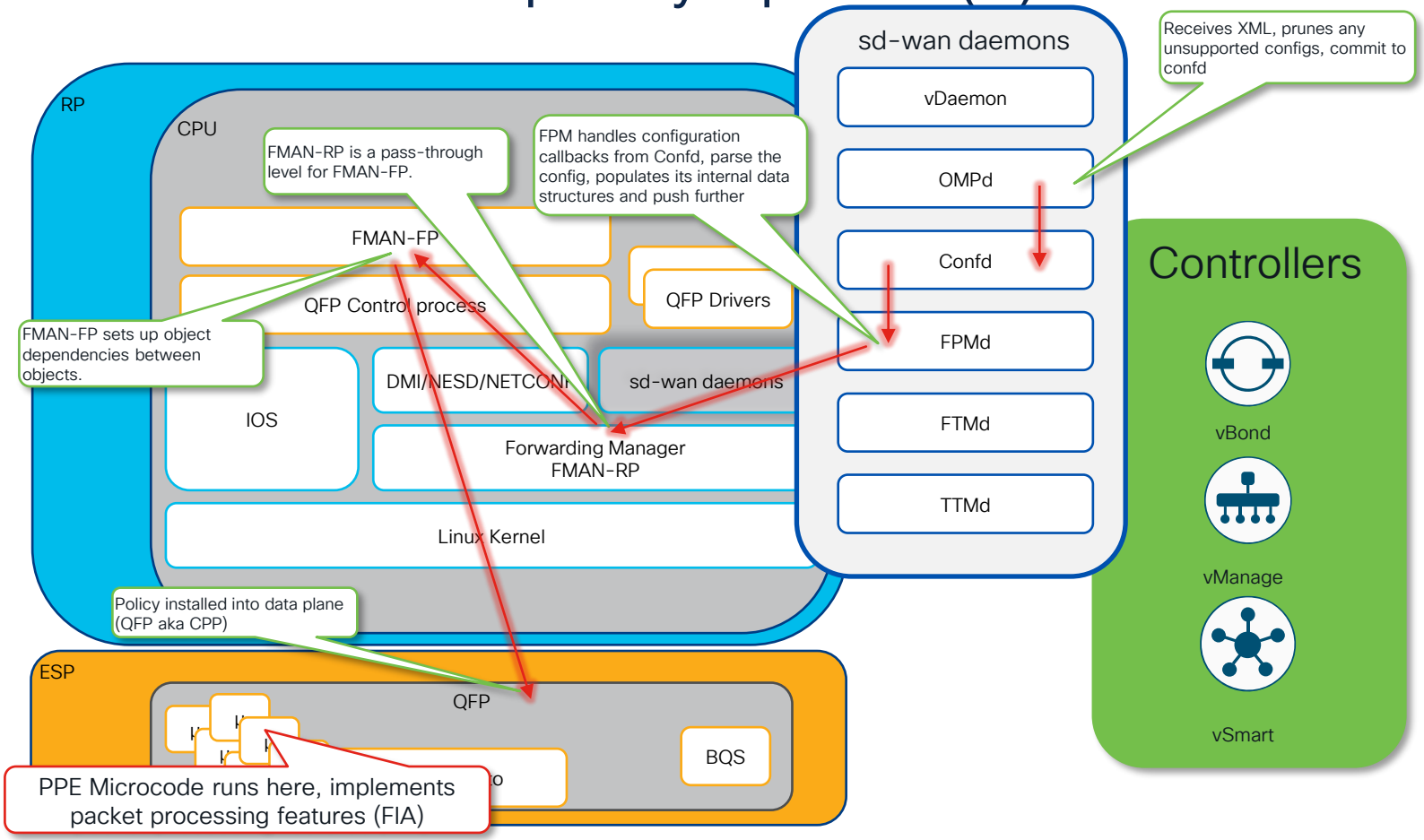
OMP centralized policy update (1)



OMP centralized policy update (2)



OMP centralized policy update (3)



Key points of this section

- All platforms using IOS-XE have a similar architecture.
- The Route Processor (RP), which runs the Linux kernel and multiple processes, handles the control plane and inter-process communications. RP programs data plane.
- The Data Plane, known as Quantum Flow Processor (QFP), uses the Feature Invocation Array (FIA) for traffic processing, including data & AAR policies, ACL, QoS marking, security policies and so on.
- The same troubleshooting tools apply across different platforms for both control plane and transit data traffic.

Back on track: Policy programming low-level verification (step 5)

Policy programming verification on WAN Edge

Why? Certainly for fun! Put yourself into shoes of TAC engineer ☺

When? For example, policy does not work and log message like below was seen:

```
%FMFP-3-OBJ_DWNLD_TO_DP_STUCK: AOM download to Data Plane is stuck for more
than 1800 seconds for obj[180464] type[711] pending-issue Req-create Issued-
none 'class class_name NEW-POLICY-seq-10 class_key 12:10’
```

Only few basic terms and concepts you need to know and remember (simplified):

- AOM is Asynchronous Object Manager, control plane thing that allows processes to continue with other tasks without waiting for the IPC operation to finish
- AOM state “Done” == Good and “Pending” == ”Bad”
- Class-group == Policy, Class == Policy Sequence, just a fancy terms for the known things
- Whole policy should be reflected starting from OMP, via Forwarding Policy Manager Daemon (FPMD) to Forwarding Manager (FMAN) and then to QFP (data plane)
- FMAN-RP is just passthrough level for policy objects, hence nothing to check there, check rather FMAN-FP

Recap: Data Policy as per FPMD view

I will use simple DIA policy for demonstration with just single sequence:

```
cE1_BR1#show sdwan policy from-vsmart
from-vsmart data-policy VPN_1_NAT
 direction from-service
 vpn-list VPN_1
  sequence 1
   match
    source-data-prefix-list LAN
   action accept
    nat use-vpn 0
    no nat fallback
  default-action drop
from-vsmart lists vpn-list VPN_1
 vpn 1
from-vsmart lists data-prefix-list LAN
 ip-prefix 10.10.10.0/24
```

OMP policy processing troubleshooting

If FPMD related output (**`show sdwan policy from-vsmart`**) has some issues already (wrong or incomplete policy, no policy at all), then OMP policy processing to be debugged:

- Set logging marker:
  - **`set logging marker MY_DEBUG`**
- Enable debugs IOS-XE release < 17.10
  - **`debug platform software sdwan omp policy level high`**
- Enable debugs IOS-XE release >= 17.10
  - **`set platform software trace ompd R0 ompd-policy verbose`**
  - **`set platform software trace ompd R0 ompd-event verbose`**
- Reset OMP (similar to BGP hard reset, be careful!)
  - **`clear sdwan omp all`**
- Check logs and look for any errors:
  - **`show logging process ompd internal start marker MY_DEBUG`**

If no failures observed, config likely committed to configuration database (CDB) by ConfD and transferred to FPM successfully.

Policy pogramming verification on WAN Edge (1)

Control Plane (RP)

Check FMAN-FP policy binding to the target VRF, find class-group (policy) id and verify AOM status in one go:

```
cE1_BR1#show platform software sdwan fp active policy bind summary
Target-id  Target-Type  Dir   AF          CG-Type          Group-id  AOM-id  AOM-Status  CG-Name
-------------------------------------------------------------------------------------------------------
3          VRF          IN    V4          DATA             1         116194  Done        VPN_1_NAT-VPN_1
```

Keep in mind that VRF ID does not always match to its name (which just happens to be a number also):

```
cE1_BR1#show ip vrf detail 1  | include Id
VRF 1 (VRF Id = 3); default RD 1:1; default VPNID <not set>
```

Policy pogramming verification on WAN Edge (2)

Control Plane (RP)

Next, you can find all class identifiers (sequences) for the corresponding group-id (policy) which was determined in the previous command:

```
cE1_BR1#show platform software sdwan fp active policy class summary
Group-id  Class-id  AOM-id  AOM-status  Class-Name
-------------------------------------------------------------------
1         1         116195  Done        VPN_1_NAT-VPN_1-seq-1
1         65535     116192  Done        VPN_1_NAT-VPN_1-def-class
```

*Calss-id 65535 is a default-action of the policy

Real life example of a failure. Sequence 1 is absent (e.g. because it has match based on an app and custom-apps definition download failed from vManage SD-AVC):

```
cE1_BR1#show platform software sdwan fp active policy class summary
Group-id  Class-id  AOM-id  AOM-status  Class-Name
-------------------------------------------------------------------
1         65535     116192  Done        VPN_1_NAT-VPN_1-def-class
```

Policy pogramming verification on WAN Edge (3)

Control Plane (RP)

You can verify class-group (policy) details and programming status in FMAN-FP, it is human readable and should be similar to FPMD (**`show sdwan policy from-vsmart`**) view (unless there is a problem):

```
cE1_BR1#show platform software sdwan f0 policy cg 1 detail
Policy: VPN_1_NAT-VPN_1, type: DATA, aom_id: 116191, aom_status: Done
  sequence 1
    name: VPN_1_NAT-VPN_1-seq-1, aom_id: 116195, aom_status: Done
    filters:
      match SRC OG IPV4
        value 57345
    actions: fo_aom_id: 116198, aom_status: Done
      action accept
      action nat_dia
  sequence 65535
    name: VPN_1_NAT-VPN_1-def-class, aom_id: 116192, aom_status: Done
    filters:
      match WILDCARD
    actions: fo_aom_id: 116193, aom_status: Done
      action drop
      action count
  target id: 1, dir: IN, af: V4, type: VRF, aom_id: 116194, aom_status: Done
```

Note filter object ID (match conditions)

Policy pogramming verification on WAN Edge (4)

Control Plane (RP)

Based on filter object IDs, we can verify the object programming in FMAN-FP:

```
cE1_BR1#show platform software common-classification f0 object-group ipv4 57345
  OG ID: 57345
  OG TYPE: IPV4
  OG Name: LAN_vs
  Pending Entry List Size: 0
  Num LKUPs in hash: 1
  Num LKUPs in Update: 0
  AOM EPOCH: 0
  State: PD Created
```

Or you can verify all objects in one go if there are not too many of them:

```
cE1_BR1#show platform software common-classification f0 object-group all
Total Number of OGs: 1

og id        og name               og type      lkup in upd  state
------------------------------------------------------------------------------
57345        LAN_vs                IPV4         0            PD Created
```

Policy pogramming verification on WAN Edge (5)

Data Plane (QFP)

Likewise, we should verify policy in QFP (data plane).

First, ensure that feature was enabled in Features Invocation Array (FIA) for interface:

```
cE1_BR1#show platform hardware qfp active interface if-name GigabitEthernet 4 | include SDWAN
  SDWAN_POLICY_FIA
```

*If localized data policy was enabled (ACL), you would see also “SDWAN_ACL_IN/OUT” in the list

If we need to verify ACL (local policy), we will also need QFP interface ID “handle”:

```
cE1_BR1#show platform hardware qfp active interface if-name GigabitEthernet4 | include QFP interface handle
  QFP interface handle: 9
```

And for data or AAR policy, which is applied on per-VRF basis, you need to know VRF ID (not the name “1”):

```
cE1_BR1#show ip vrf detail 1  | include Id
VRF 1 (VRF Id = 3); default RD 1:1; default VPNID <not set>
```

Policy pogramming verification on WAN Edge (6)

Data Plane (QFP)

Then find QFP class-group (policy) ID:

```
cE1_BR1#show platform hardware qfp active classification class-group-manager class-group client sdwan all
QFP classification class client all group

  class-group [SDWAN:1] VPN_1_NAT-VPN_1
```

Policy pogramming verification on WAN Edge (7)

Using QFP class ID, dump details of a class-group (policy) match conditions:

Data Plane (QFP)

```
cE1_BR1#show platform hardware qfp active classification class-group-manager class-group client sdwan 1
class-group [sdwan-cg:1] VPN_1_NAT-VPN_1 (classes: 2)
 clients:
 fields: ipv4_og_src:1 any:1  (100000:0:0:200:0:00000000)
  (1) class: logical-expression [1.1] VPN_1_NAT-VPN_1-seq-1 (filters: 1)
         lexp: LOG-EXP: [1]
    (1) filter: generic [1.1.1] (rules: 1)
      (1) rule: generic  [1.1.1.1] (permit)
         match ipv4_og_src 57345
  (65535) class: logical-expression [1.65535] VPN_1_NAT-VPN_1-def-class (filters: 1)
         lexp: LOG-EXP: [1]
    (1) filter: generic [1.65535.1] (rules: 1)
      (1) rule: generic  [1.65535.1.1] (permit)
         match any
```

To decode individual object like prefix-lists from the policy, use ID of the object and find its name:

```
cE1_BR1#show platform hardware qfp active classification class-group-manager object-group all | include 57345
  LAN_vs:57345  Type: IPV4  No. of Entries: 1

cE1_BR1#show platform hardware qfp active classification class-group-manager object-group name LAN_vs
Object-group LAN_vs:57345  Type: IPV4  No. of Entries: 1 AOM Id: 116190
    id: 0xe0010001  10.10.10.0/255.255.255.0
```

Policy pogramming verification on WAN Edge (8)

Then check action statements in the class-group (policy) which are stored separately:

Data Plane (QFP)

```
cE1_BR1#show platform hardware qfp active feature sdwan client policy class-group 1 detail
Policy: 1 type: NONE og_lkup: ipv4_src 4 ipv4_dst 0 ipv6_src 0 ipv6_dst 0 app_id 3
  sequence 1
  actions
    accept
    nat_dia
  sequence 65535
  actions
    drop
    count
  target id: 1, dir: IN, af: V4, type: VRF
```

... or per class (sequence):

```
cE1_BR1#show platform hardware qfp active feature sdwan client policy class-group 1 class 1
QFP sdwan client policy GroupId information


Group id            : 1
Class id            : 1

actions
  accept
  nat_dia
```

Policy pogramming verification on WAN Edge (9)

Data Plane (QFP)

Based on QFP class-group (policy) ID and QFP interface handle (for ACL) or VRF ID (for AAR/Data policy), we can check TCAM* programming:

```
cE1_BR1#show platform hardware qfp active classification feature-manager class-group tcam sdwan 1 ?
  acl             sdwan acl feature
  app-route       app route feature
  data-policy     data policy feature
  utd-tls-policy  UTD TLS decryption feature
```

TCAM – Ternary Content Addressable Memory used for very fast classification of traffic and can be implemented in hardware for even faster processing. The "ternary" aspect refers to its ability to store and process three states per bit: 0, 1, and "don't care." This capability allows TCAM to perform more flexible and efficient pattern matching than binary content addressable memory (CAM), which only handles two states.

Policy pogramming verification on WAN Edge (10)

Data Plane (QFP)

```
cE1_BR1#show platform hardware qfp active classification feature-manager class-group tcam sdwan 1 data-policy 3
proto-v4 input detail
QFP classification class group CACE

CACE classification Info::
Total entries: 2  Available entries: 65534  Total RAM used:612 bytes
IPV4 Traffic Classifier: total_entries=2 default_entry_idx=1 num_attr_clusters=2
IPV6 Traffic Classifier: total_entries=1 default_entry_idx=1 num_attr_clusters=2
MPLS Traffic Classifier: total_entries=1 default_entry_idx=1 num_attr_clusters=2
L2   Traffic Classifier: total_entries=1 default_entry_idx=1 num_attr_clusters=2

class-group [sdwan-cg:2] (classes: 2, total number of vmrs: 2)
key name: 320_Viptela_og_02  value size: 0 result size: 16 tcam id: SOFTWARE TCAM
object-group: (ipv4) lkup handle id (source: 4 dest: 0)
              (ipv6) lkup handle id (source: 0 dest: 0)

              (user) lkup handle id (appid: 0)
              (fqdn) is_valid: No version: 0
     internal (ipv6) lkup handle id (source: 0 dest: 0)
              (ext_data1) is_installed: No type: None

   Value: : 01000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00800000 00000000
    Mask: : 01000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00c00000 00000000
  Result: : 10000000 01000000 01000000 00000000

   Value: : 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
    Mask: : 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
  Result: : 02000100 00000000 ffff0000 00000000
```

Sequence 1

Default sequence 65535

Policy programming verification on WAN Edge

What if you finnaly happened to (not) find some (any) problems at such a low level?

KEEP CALM
AND
CALL
CISCO TAC

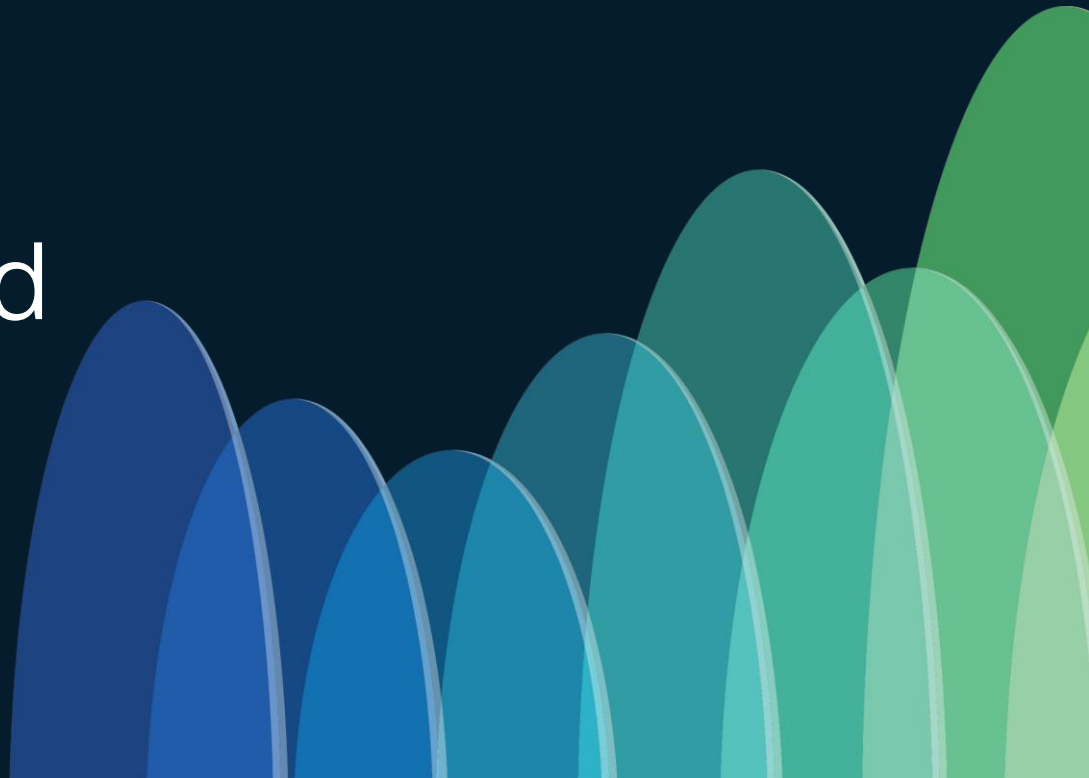
Localized policies

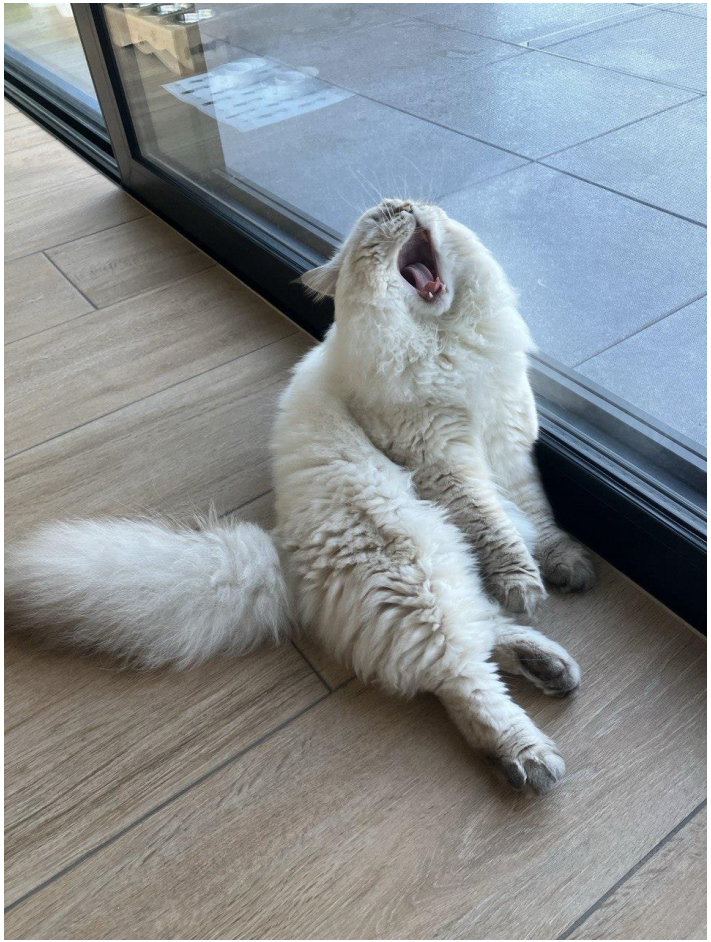
Localized policies and templates

show netconf-yang sessions

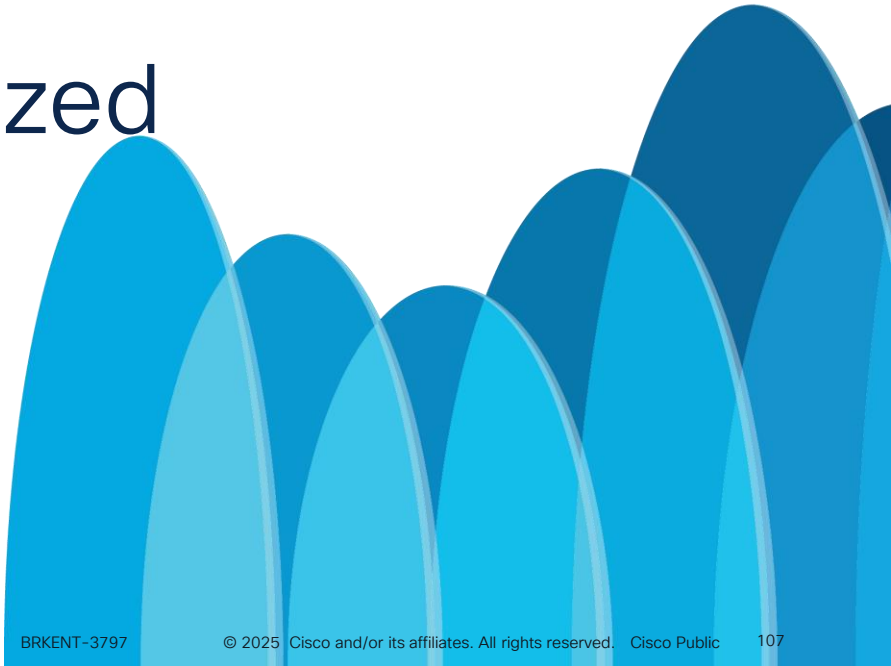
show sdwan config-pull history

Part 2. Issues seen in the field

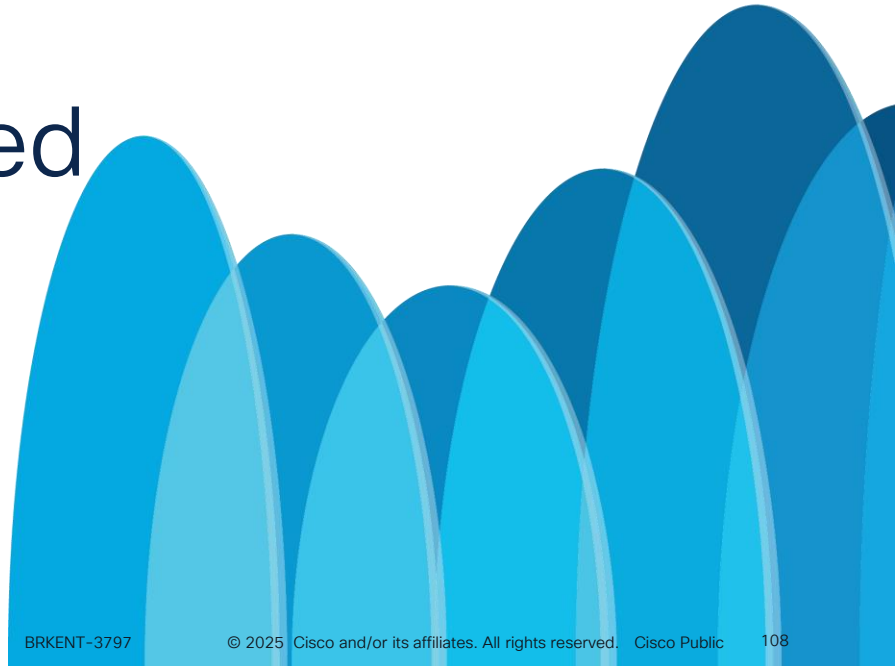




(Not so) Well Known Failures with Centralized Control Policies

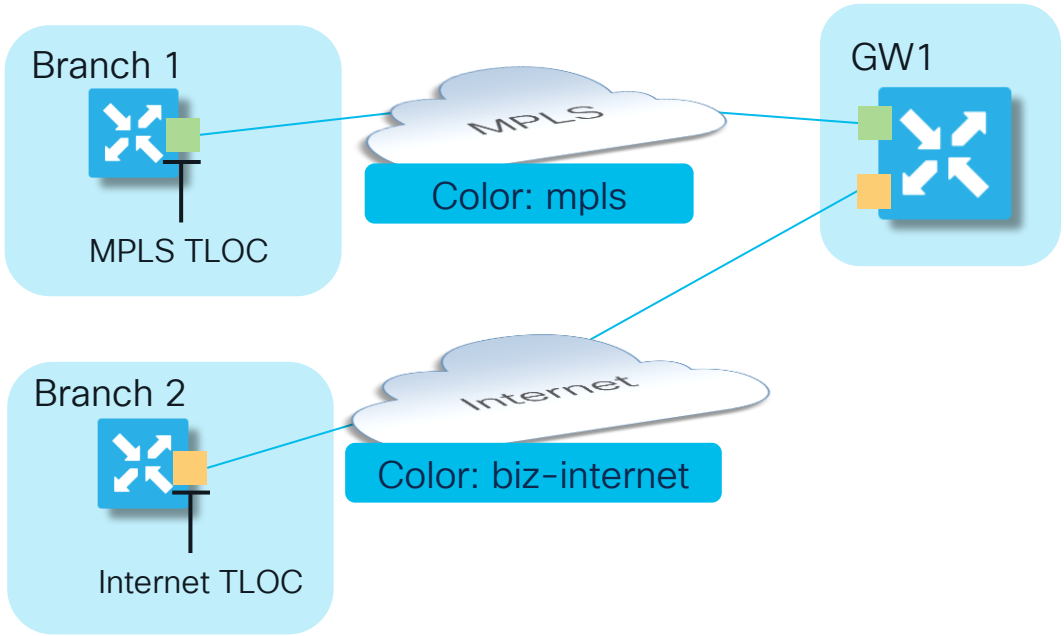


Centralized Control Policies: Failures in overlays with disjoined underlays

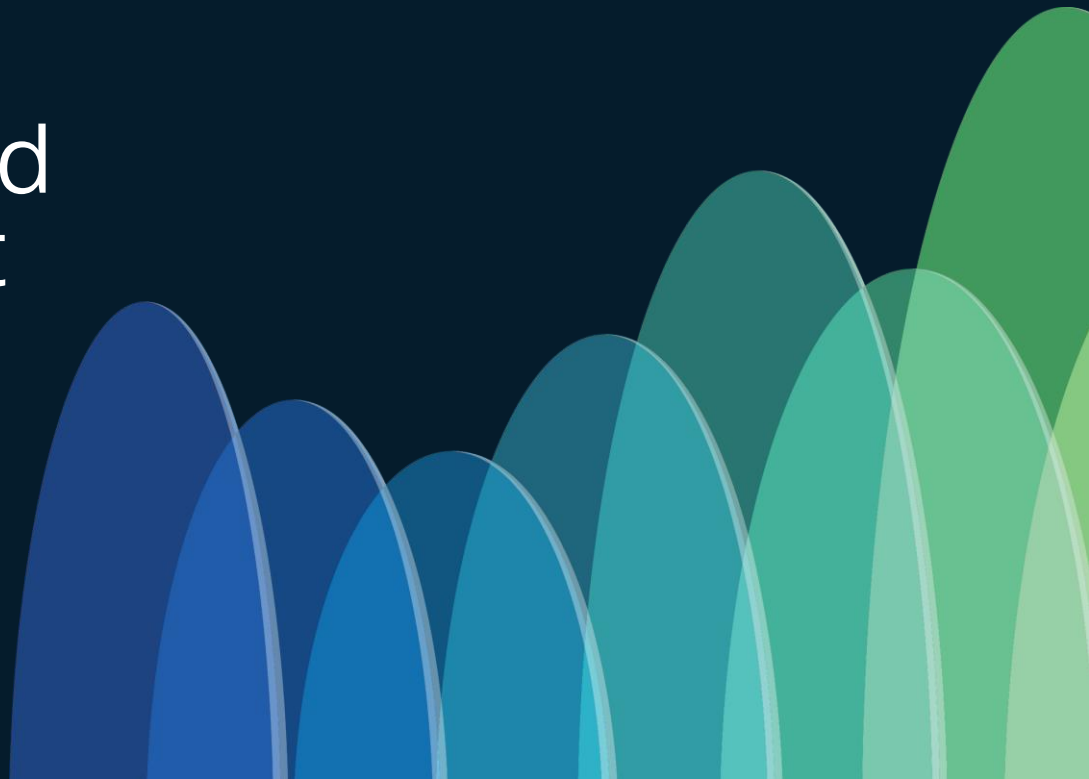


Recap: What is an overlay with disjoined underlay?

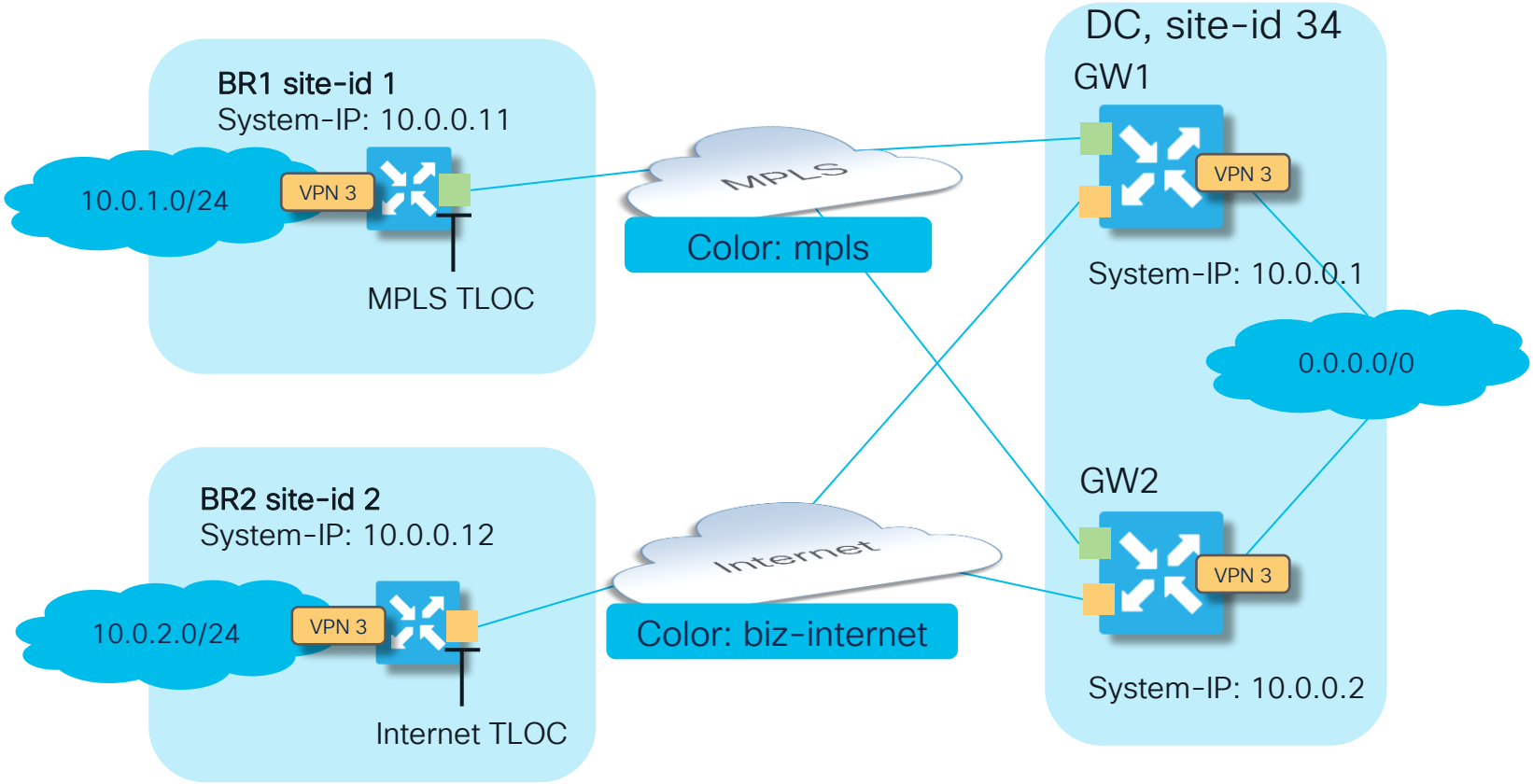
This overlay network connects different sites through various transport types, but the transports are not directly connected to each other



Case 0: Disjoined underlay without control policy (as a warmup)



Case 0. Disjoined underlay



Case 0. Disjoined underlay

A more specific route from BR2 won't be installed into RIB on BR1 because there is no direct data plane tunnel between BR1 and BR2. As a result, the TLOC is unresolved, which leads to the OMP route also being unresolved:

```
BR1#show sdwan omp routes vpn 3 10.0.2.0/24 | b PATH
                  PATH                        ATTRIBUTE
FROM PEER         ID     LABEL    STATUS      TYPE        TLOC IP          COLOR            ENCAP  PREFERENCE
-----------------------------------------------------------------------------------------------------------
10.0.0.101        531    1011     Inv,U       installed   10.0.0.12        biz-internet     ipsec  -
```

...and hence traffic will follow default route to GWs:

```
BR1#show sdwan omp routes vpn 3 0.0.0.0/0 | b PATH
                  PATH                        PSEUDO
FROM PEER         ID     LABEL    STATUS      KEY     TLOC IP          COLOR            ENCAP  PREFERENCE
-----------------------------------------------------------------------------------------------------------
10.0.0.101        1011   1004     Inv,U       1       10.0.0.2         mpls             ipsec  -
10.0.0.101        1012   1004     C,I,R       1       10.0.0.2         biz-internet     ipsec  -
10.0.0.101        1071   1008     Inv,U       1       10.0.0.1         mpls             ipsec  -
10.0.0.101        1072   1008     C,I,R       1       10.0.0.1         biz-internet     ipsec  -
10.0.0.102        1355   1004     Inv,U       1       10.0.0.2         mpls             ipsec  -
10.0.0.102        1356   1004     C,R         1       10.0.0.2         biz-internet     ipsec  -
10.0.0.102        1375   1008     Inv,U       1       10.0.0.1         mpls             ipsec  -
10.0.0.102        1376   1008     C,R         1       10.0.0.1         biz-internet     ipsec  -
```

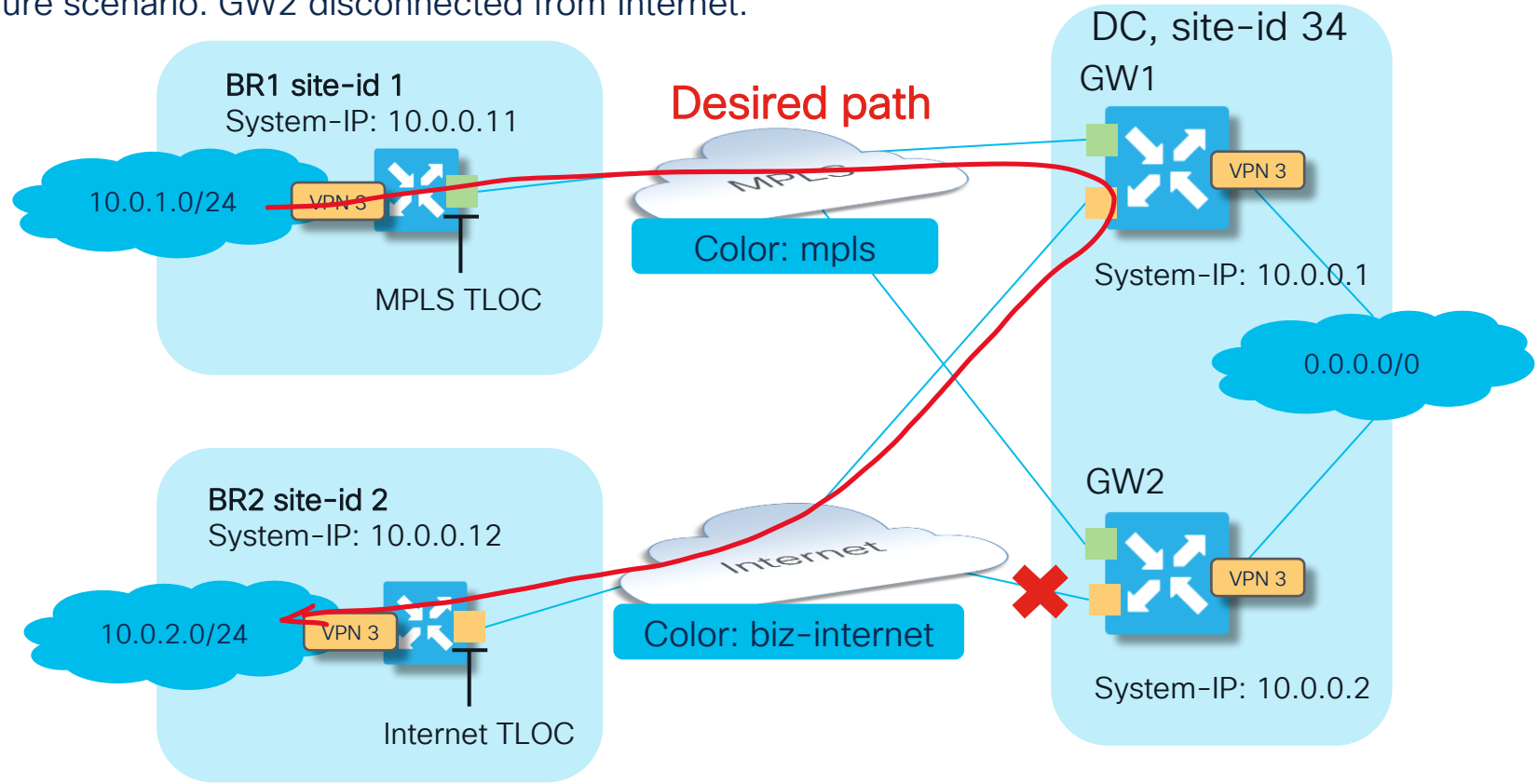
Case 0. Disjoined underlay

Depending on EMCP hash results, traffic follows default route via GW1 or GW2:

```
BR1#sh ip route vrf 3 0.0.0.0

Routing Table: 3
Routing entry for 0.0.0.0/0, supernet
  Known via "omp", distance 251, metric 0, candidate default path, type omp
  Last update from 10.0.0.12 on Sdwan-system-intf, 00:03:57 ago
  Routing Descriptor Blocks:
    10.0.0.2 (default), from 10.0.0.2, 00:03:57 ago, via Sdwan-system-intf
      Route metric is 0, traffic share count is 1
  * 10.0.0.1 (default), from 10.0.0.1, 00:03:57 ago, via Sdwan-system-intf
      Route metric is 0, traffic share count is 1

BR1#traceroute vrf 3 10.0.2.2 source 10.0.1.2 numeric
Type escape sequence to abort.
Tracing the route to 10.0.2.2
VRF info: (vrf in name/id, vrf out name/id)
  1 192.168.10.11 100 msec 1 msec 1 msec
  2 10.0.2.2 2 msec *  1 msec
BR1#traceroute vrf 3 10.0.2.2 source 10.0.1.1 numeric
Type escape sequence to abort.
Tracing the route to 10.0.2.2
VRF info: (vrf in name/id, vrf out name/id)
  1 192.168.10.12 1 msec 1 msec 0 msec
  2 10.0.2.2 1 msec *  2 msec
```

Case 0. Disjoined underlay

Failure scenario. GW2 disconnected from Internet.



Case 0. Disjoined underlay

Problem: during GW2 internet failure, ~50% traffic wil be blackholed now due to ECMP:

```
BR1#traceroute vrf 3 10.0.2.2 source 10.0.1.1 numeric
Type escape sequence to abort.
Tracing the route to 10.0.2.2
VRF info: (vrf in name/id, vrf out name/id)
  1 192.168.9.12 1 msec 1 msec 0 msec
  2 192.168.9.12 !H  *  !H
BR1#traceroute vrf 3 10.0.2.2 source 10.0.1.2 numeric
Type escape sequence to abort.
Tracing the route to 10.0.2.2
VRF info: (vrf in name/id, vrf out name/id)
  1 192.168.9.11 1 msec 0 msec 1 msec
  2 10.0.2.2 2 msec *  2 msec
BR1#ping vrf 3 10.0.2.2 source 10.0.1.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.0.2.2, timeout is 2 seconds:
Packet sent with a source address of 10.0.1.1
U.U.U
Success rate is 0 percent (0/5)
BR1#ping vrf 3 10.0.2.2 source 10.0.1.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.0.2.2, timeout is 2 seconds:
Packet sent with a source address of 10.0.1.2
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
```

Case 0. Disjoined underlay.

Typical solution. Configure control policy to change TLOCs “Next-Hops” (aka hub-n-spoke)

```
policy
 control-policy CHANGE_TLOC_NH
  sequence 10
   match route
    site-list BR1
    vpn       3
   !
   action accept
    set
     tloc-list INET_TLOCS
    !
   !
  !
  sequence 20
   match route
    site-list BR2
    vpn       3
   !
   action accept
    set
     tloc-list MPLS_TLOCS
    !
   !
  !
  default-action accept
 !
!
```

```
policy
 lists
  site-list ALL_BRANCHES
   site-id 1
   site-id 2
  !
  site-list BR1
   site-id 1
  !
  site-list BR2
   site-id 2
  !
  tloc-list INET_TLOCS
   tloc 10.0.0.1 color biz-internet encap ipsec
   tloc 10.0.0.2 color biz-internet encap ipsec
  !
  tloc-list MPLS_TLOCS
   tloc 10.0.0.1 color mpls encap ipsec
   tloc 10.0.0.2 color mpls encap ipsec
  !
 !
!
apply-policy
 site-list ALL_BRANCHES
  control-policy CHANGE_TLOC_NH out
 !
!
```

Case 0. Disjoined underlay

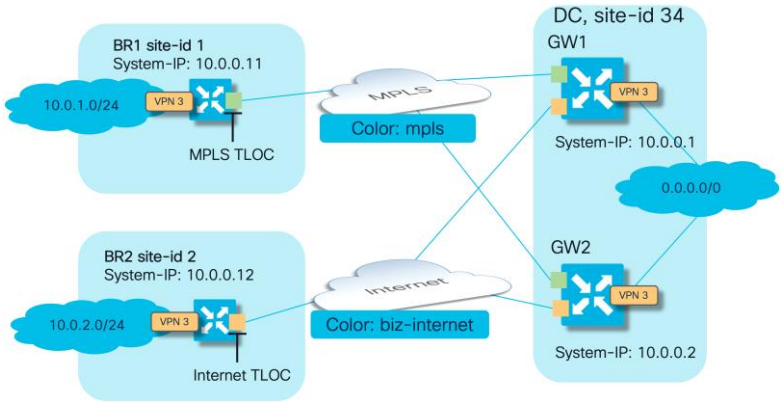
Typical solution - testing

Once policy applied, TLOC rewrite happens to GW's TLOCs:

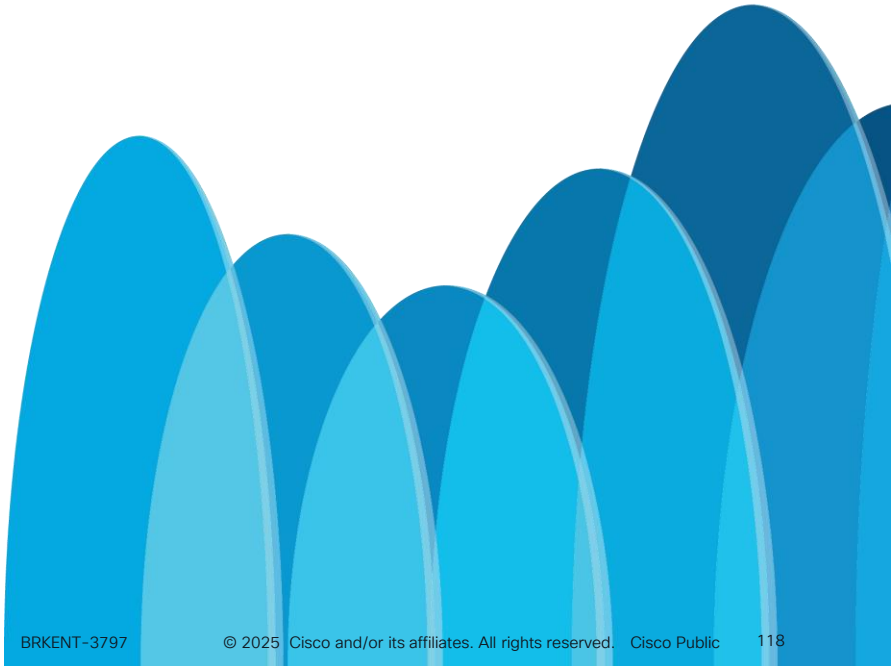
```
BR1#show sdwan omp routes vpn 3 10.0.2.0/24 | b PATH
                  PATH                        ATTRIBUTE
FROM PEER         ID     LABEL    STATUS      TYPE       TLOC IP          COLOR            ENCAP  PREFERENCE
-----------------------------------------------------------------------------------------------------------
10.0.0.101        2022   1008     C,I,R       installed  10.0.0.1         mpls             ipsec  -
10.0.0.101        2023   1004     C,I,R       installed  10.0.0.2         mpls             ipsec  -
```

And traffic follows specific path to BR2 subnet:

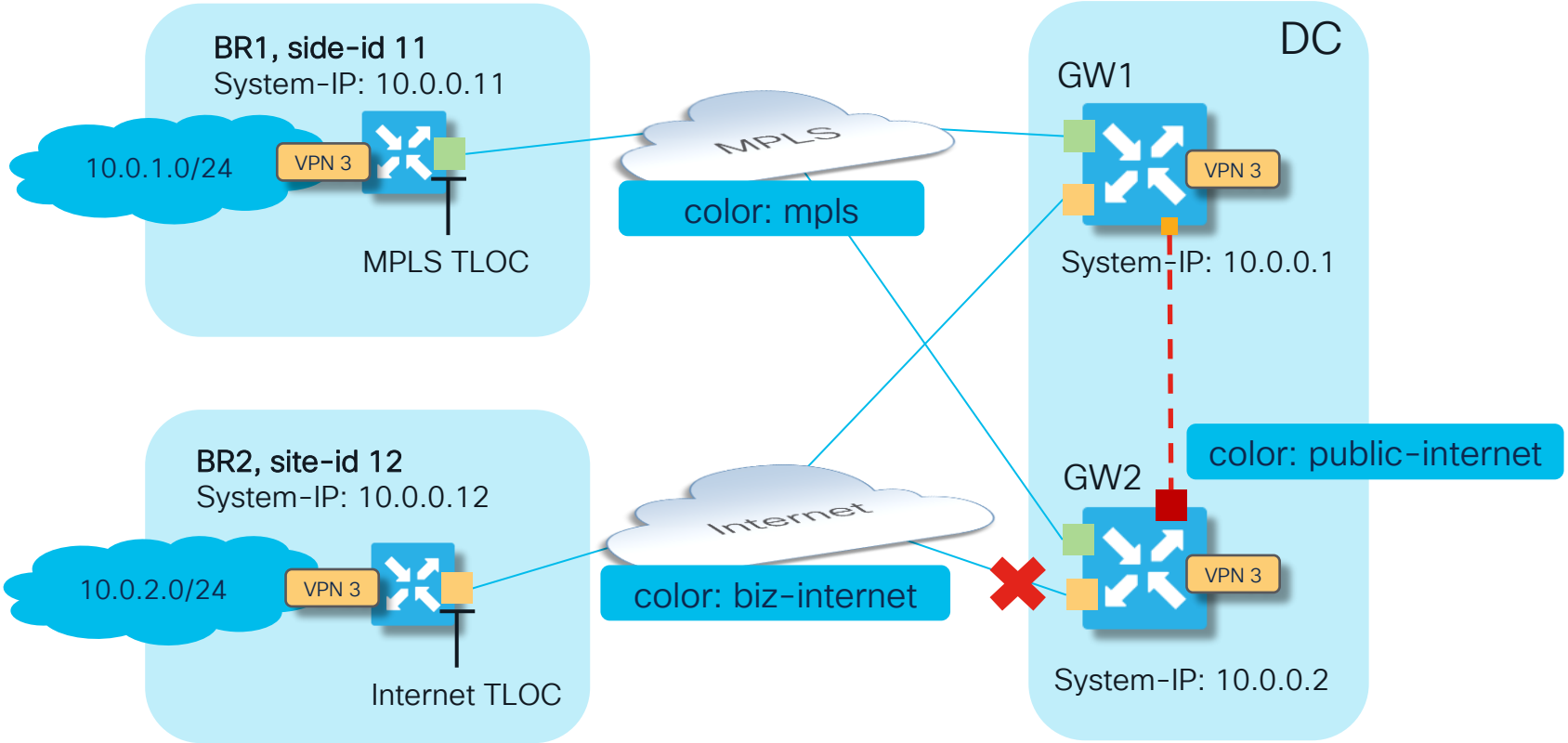
```
BR1#sh ip route vrf 3 10.0.2.0

Routing Table: 3
Routing entry for 10.0.2.0/24
  Known via "omp", distance 251, metric 0, type omp
  Last update from 10.0.0.2 on Sdwan-system-intf, 00:03:00 ago
  Routing Descriptor Blocks:
    10.0.0.2 (default), from 10.0.0.2, 00:03:00 ago, via Sdwan-system-intf
      Route metric is 0, traffic share count is 1
  * 10.0.0.1 (default), from 10.0.0.1, 00:03:00 ago, via Sdwan-system-intf
      Route metric is 0, traffic share count is 1
```



Case 0. Other possible solutions for sake of having complete picture

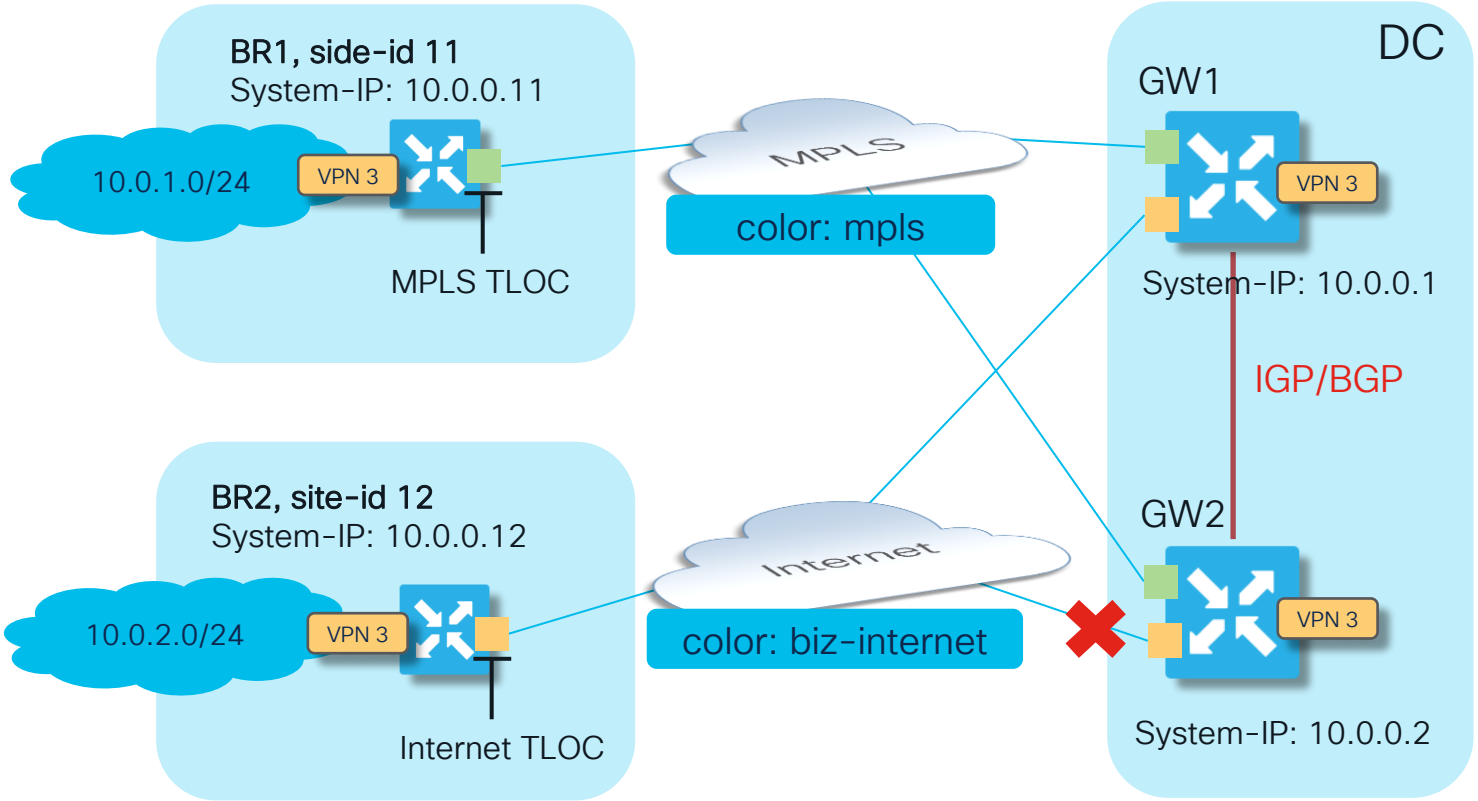


Case 0. Disjoined underlay
Solution 2. Configure tloc-extension



Case 0. Disjoined underlay

Solution 3. Configure IGP/BGP peering between GW1&GW2 and bidirectional OMP redistribution



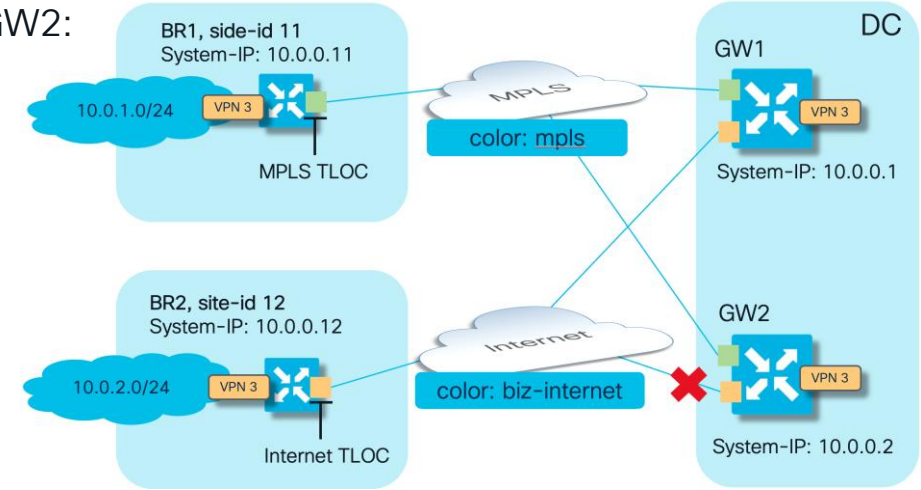
BGP SoO, OSPF DN-bit or EIGRP external protocol tag will be used for loop prevention

Case 0. Disjoined underlay
Solution 4. Multi-Regional Fabric “Light”

Transit Gateways (TGW) enabled on GW1 & GW2:

```
GW1#config-t
admin connected from 127.0.0.1 using console on GW1
GW1(config)# system
GW1(config-system)# transport-gateway enable
GW1(config-system)# commit
```

```
GW2#config-t
admin connected from 127.0.0.1 using console on GW2
GW2(config)# system
GW2(config-system)# transport-gateway enable
GW2(config-system)# commit
```

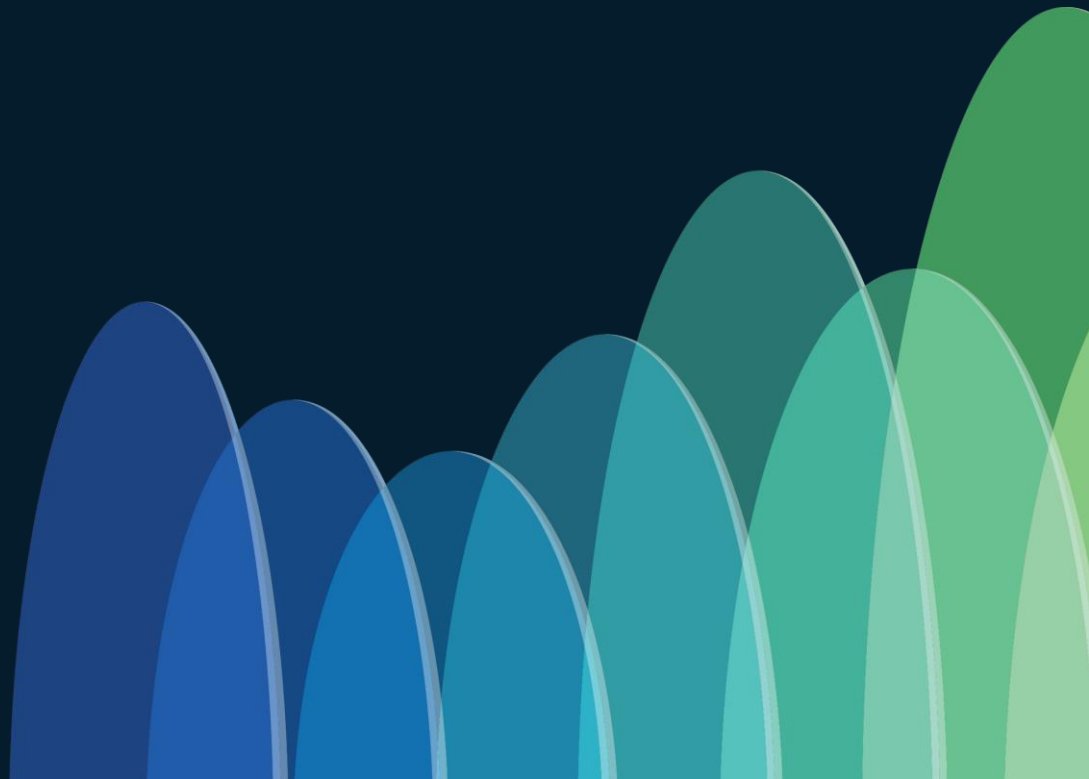


BR1 traffic to BR2 follows path via GW1 mpls:

```
BR1#show sdwan omp routes vpn 3 10.0.2.0/24 | b PATH
                  PATH                        ATTRIBUTE
FROM PEER         ID      LABEL    STATUS     TYPE        TLOC IP          COLOR            ENCAP
PREFERENCE
-----------------------------------------------------------------------------------------------------
--------
169.254.206.4     44      1006     C,I,R      installed   10.0.0.1         mpls             ipsec  -
169.254.206.4     45      1006     Inv,U      installed   10.0.0.1         biz-internet     ipsec  -
169.254.206.4     46      1011     Inv,U      installed   10.0.0.12        biz-internet     ipsec  -
169.254.206.5     61      1006     C,R        installed   10.0.0.1         mpls             ipsec  -
169.254.206.5     62      1006     Inv,U      installed   10.0.0.1         biz-internet     ipsec  -
169.254.206.5     63      1011     Inv,U      installed   10.0.0.12        biz-internet     ipsec  -
```

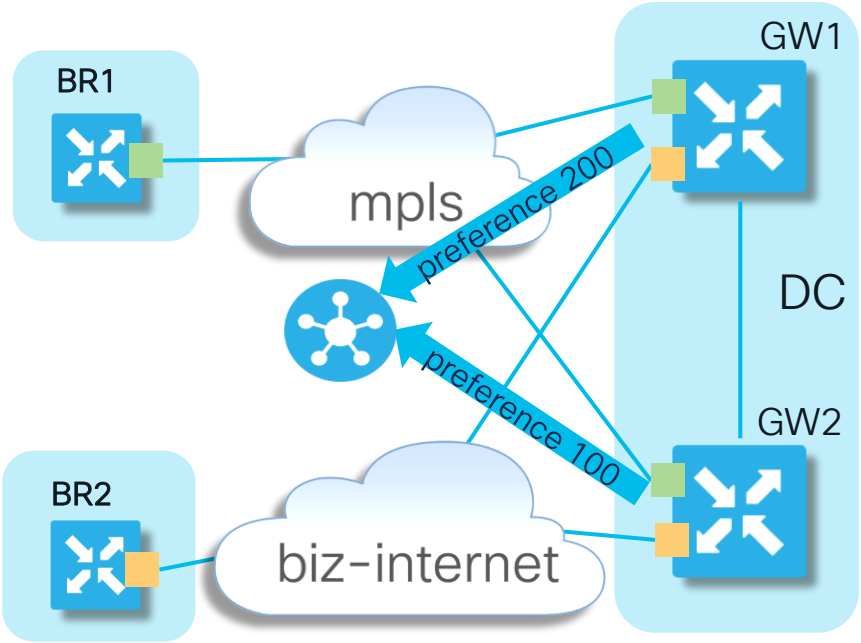
IOS-XE 17.9

Case 1: Active-standby redundancy failure with disjoined underlay



Case 1. Active-standby redundancy failure with disjoint underlay.

Failure to influence path with OMP route preference.



- Main objective here is to ensure preferred path to DC subnets is via GW1 (site-id 1)
- The OMP route preference for routes advertised by GW1 is set to 200 using a vSmart inbound policy. Alternatively, service side routing protocol metric can be adjusted to influence path selection

Case 1. Active-standby redundancy failure with disjoint underlay.

Original centralized control policy on vSmart:

```
policy
 lists
  site-list GW1
   site-id 1
  !
 !
!
 control-policy PREFER_GW1
  sequence 10
   match route
    site-id 1
   !
   action accept
    set
     preference 200
    !
   !
  !
  default-action accept
 !
!
apply-policy
 site-list GW1
  control-policy PREFER_GW1 in
 !
!
```

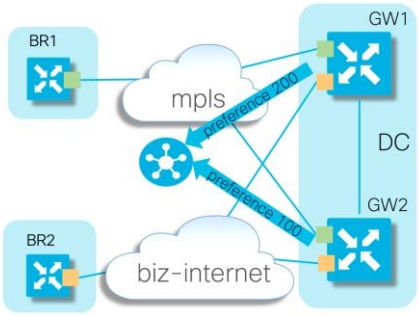
Can you see potential problem here?

Case 1. Active-standby redundancy failure with disjoint underlay.

The problem.

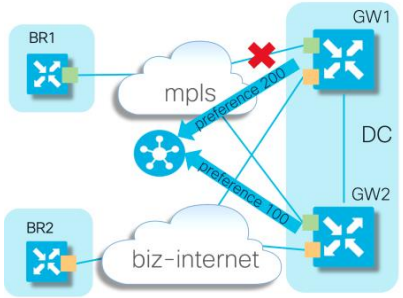
BR1 prefers GW1 because of the control policy:

```
BR1# show sdwan omp routes vpn 3 0.0.0.0/0 | begin PATH
                  PATH                        PSEUDO
FROM PEER         ID     LABEL    STATUS      KEY     TLOC IP          COLOR            ENCAP  PREFERENCE
-----------------------------------------------------------------------------------------------------------
10.0.0.101        1066   1008     C,I,R       1       10.0.0.1         mpls             ipsec  200
10.0.0.101        1067   1008     Inv,U       1       10.0.0.1         biz-internet     ipsec  200
10.0.0.102        2142   1008     C,R         1       10.0.0.1         mpls             ipsec  200
10.0.0.102        2143   1008     Inv,U       1       10.0.0.1         biz-internet     ipsec  200
```



But in case of mpls link failure on GW1, there are no more valid paths available on BR1:

```
BR1# show sdwan omp routes vpn 3 0.0.0.0/0 | begin PATH
                  PATH                        PSEUDO
FROM PEER         ID     LABEL    STATUS      KEY     TLOC IP          COLOR            ENCAP  PREFERENCE
-----------------------------------------------------------------------------------------------------------
10.0.0.101        1067   1008     Inv,U       1       10.0.0.1         biz-internet     ipsec  200
10.0.0.102        2143   1008     Inv,U       1       10.0.0.1         biz-internet     ipsec  200
```



If we follow routing troubleshooting workflow, we will find that BR1 can't establish data plane with biz-internet TLOC of GW1 and BFD is down, obviously

Case 1. Active-standby redundancy failure with disjoint underlay.

Why does problem arise here?

And BR1 can’t resolve path via internet because it is connected to mpls color ony

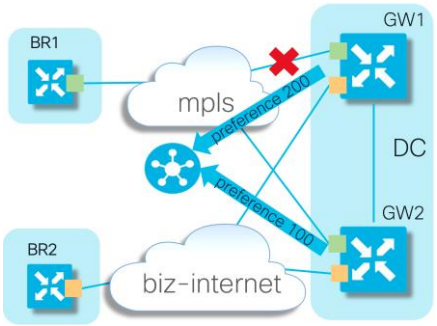
Case 1. Typical Solution.

Solution is to influence preference only **after** the best path selection (i.e. outbound control policy to set preference)

```
policy
 lists
  site-list ALL_BRANCHES
   site-id 1
   site-id 2
  !
 !
!
control-policy PREFER_BR1
  sequence 10
   match route
    site-id 1
   !
   action accept
    set
     preference 200
    !
   !
  !
  default-action accept
 !
!
apply-policy
 site-list ALL_BRANCHES
  control-policy PREFER_BR1 out
 !
```

Case 1. Typical Solution (cont.)

Testing the solution when the outbound control policy configured on vSmart which makes branches prefer GW1



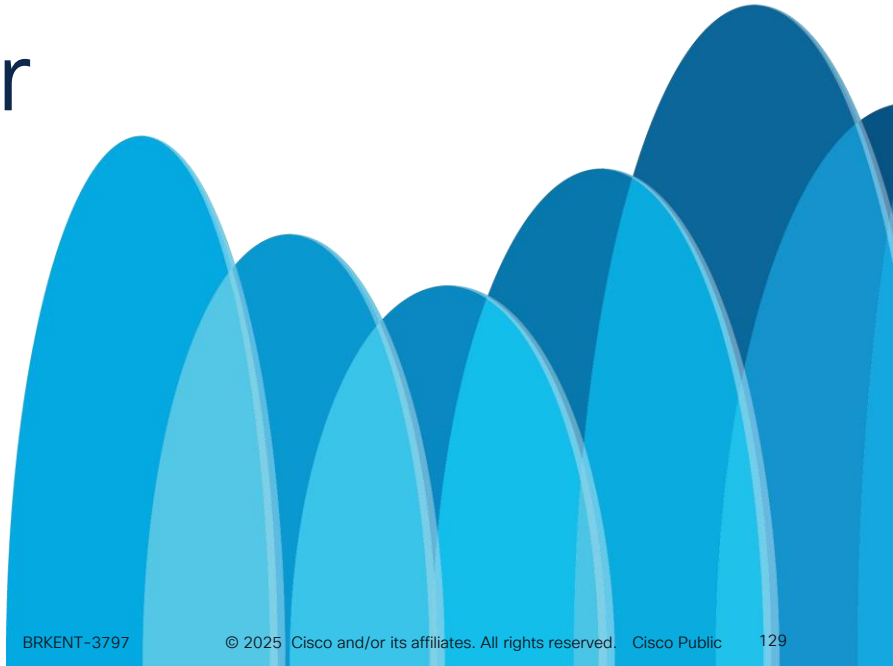
Normal pre-failover state:

```
BR1# show sdwan omp routes vpn 3 0.0.0.0/0 | b PATH
                  PATH                        PSEUDO
FROM PEER         ID     LABEL    STATUS      KEY     TLOC IP          COLOR            ENCAP  PREFERENCE
-----------------------------------------------------------------------------------------------------------
10.0.0.101        21     1008     C,I,R       1       10.0.0.1         mpls             ipsec  200
10.0.0.101        22     1008     Inv,U       1       10.0.0.1         biz-internet     ipsec  200
10.0.0.101        65     1004     R           1       10.0.0.2         mpls             ipsec  -
10.0.0.101        66     1004     Inv,U       1       10.0.0.2         biz-internet     ipsec  -
```

Failover scenario testing. GW1 has lost MPLS link, but BR1 successfully installs backup path via GW2:

```
cE1_BR1# show sdwan omp routes vpn 3 0.0.0.0/0 | b PATH
                  PATH                        PSEUDO
FROM PEER         ID     LABEL    STATUS      KEY     TLOC IP          COLOR            ENCAP  PREFERENCE
-----------------------------------------------------------------------------------------------------------
10.0.0.101        22     1008     Inv,U       1       10.0.0.1         biz-internet     ipsec  200
10.0.0.101        65     1004     C,I,R       1       10.0.0.2         mpls             ipsec  -
10.0.0.101        66     1004     Inv,U       1       10.0.0.2         biz-internet     ipsec  -
```

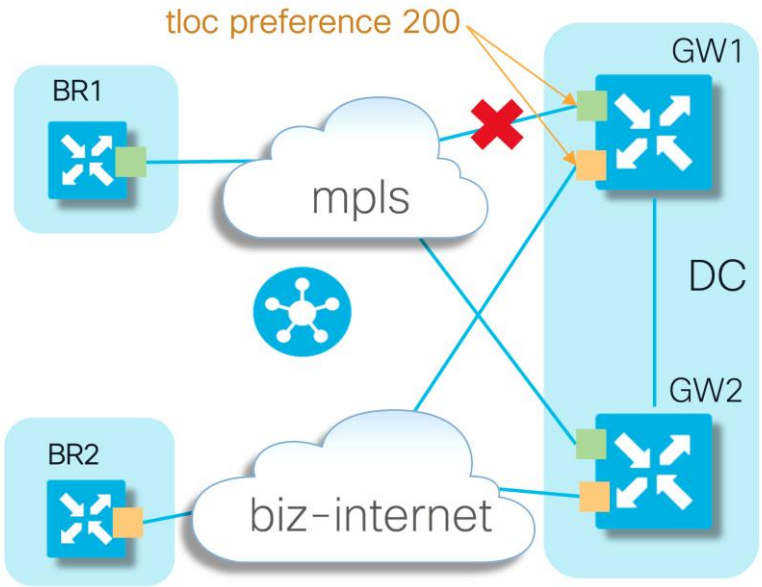
Case 2. Other possible solutions for sake of having complete picture



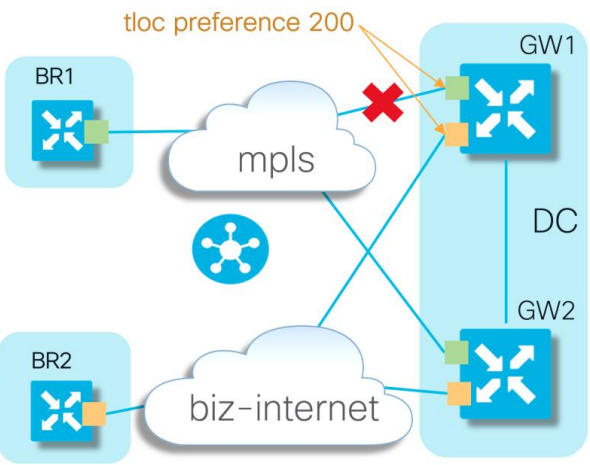
Case 2. Solution 2.

TLOC preference configured on GW1 TLOC instead of using control policy to set preference (i.e. directly on interface)

```
sdwan
 interface GigabitEthernet2
  tunnel-interface
   encapsulation ipsec preference 200
  exit
 exit
 interface GigabitEthernet3
  tunnel-interface
   encapsulation ipsec preference 200
  exit
 exit
!
```



Case 2. Solution 2. (cont.)



Normal conditions:

```
cE1_BR1# show sdwan omp routes vpn 3 0.0.0.0/0 | b PATH
                  PATH                        PSEUDO
FROM PEER         ID     LABEL    STATUS      KEY     TLOC IP          COLOR            ENCAP  PREFERENCE
-----------------------------------------------------------------------------------------------------------
10.0.0.101        22     1008     Inv,U       1       10.0.0.1         biz-internet     ipsec  -
10.0.0.101        65     1004     R           1       10.0.0.2         mpls             ipsec  -
10.0.0.101        66     1004     Inv,U       1       10.0.0.2         biz-internet     ipsec  -
10.0.0.101        1098   1008     C,I,R       1       10.0.0.1         mpls             ipsec  -
```

Route preference is **not** set. Path selection enforced with TLOC preference.

Case 2. Solution 2 (cont.)

BR1 receives all TLOCs but preference set to 200 for TLOCs from GW1 (system-ip 10.0.0.1)

```
BR1#show sdwan omp tlocs "ip 10.0.0.1" | exclude not set
--------------------------------------------------
tloc entries for 10.0.0.1
                 mpls
                 ipsec
--------------------------------------------------
            RECEIVED FROM:
peer            10.0.0.101
status          C,I,R
    Attributes:
     attribute-type    installed
     encap-proto       0
     encap-spi         12851
     encap-auth        sha1-hmac,ah-sha1-hmac
     encap-encrypt     aes256
     public-ip         192.168.9.13
     public-port       12426
     private-ip        192.168.9.13
     private-port      12426
     public-ip         ::
     public-port       0
     private-ip        ::
     private-port      0
     bfd-status        up
     site-id           1
     preference        200
     weight            1
     version           3
    gen-id             0x80000006
     carrier           default
     restrict          1
     on-demand          0
     groups            [ 0 ]
     bandwidth         0
     bandwidth-dmin    0
     bandwidth-down    0
     bandwidth-dmax    0
     adapt-qos-period  0
     adapt-qos-up      0
     qos-group         default-group
--------------------------------------------------
tloc entries for 10.0.0.1
                 biz-internet
                 ipsec
--------------------------------------------------
            RECEIVED FROM:
peer            10.0.0.101
status          C,I,R
    Attributes:
     attribute-type    installed
     encap-proto       0
     encap-spi         555
     encap-auth        sha1-hmac,ah-sha1-hmac
     encap-encrypt     aes256
     public-ip         192.168.10.13
     public-port       12366
     private-ip        192.168.10.13
     private-port      12366
     public-ip         ::
     public-port       0
     private-ip        ::
     private-port      0
     bfd-status        down
     site-id           1
     preference        200
     weight            1
     version           3
    gen-id             0x80000006
     carrier           default
     restrict          0
     on-demand          0
     groups            [ 0 ]
     bandwidth         0
     bandwidth-dmin    0
     bandwidth-down    0
     bandwidth-dmax    0
     adapt-qos-period  0
     adapt-qos-up      0
     qos-group         default-group
```

Case 2. Solution 2 (cont.)

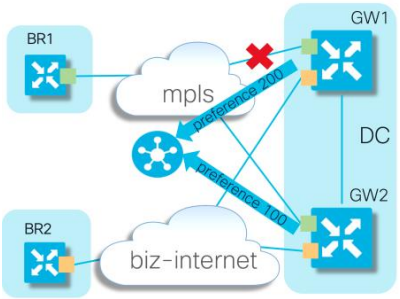
Solution 2 testing: GW1 MPLS link failure scenario.
GW2 route selected because it's the only remaining that can be resolved, all OK:

```
BR1#show sdwan omp routes vpn 3 0.0.0.0/0 | b PATH
                  PATH                        PSEUDO
FROM PEER         ID      LABEL    STATUS     KEY      TLOC IP          COLOR            ENCAP  PREFERENCE
-----------------------------------------------------------------------------------------------------------
10.0.0.101        22      1008     Inv,U      1        10.0.0.1         biz-internet     ipsec  -
10.0.0.101        65      1004     C,I,R      1        10.0.0.2         mpls             ipsec  -
10.0.0.101        66      1004     Inv,U      1        10.0.0.2         biz-internet     ipsec  -
```

Case 8. Solution 2.

Send non-best paths as well (and keep control policy inbound):

```
vsmart1# show running-config omp
omp
 no shutdown
 send-backup-paths
```



Under normal conditions:

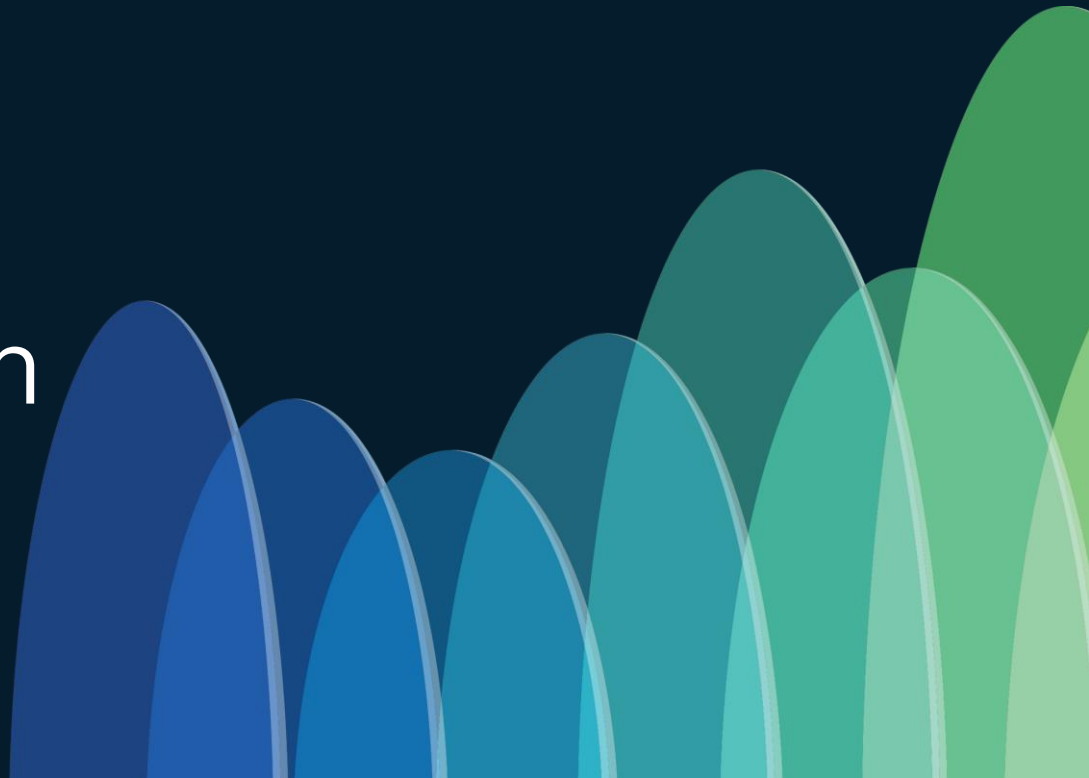
```
BR1#show sdwan omp routes vpn 3 0.0.0.0/0 | b PATH
                PATH                        PSEUDO
FROM PEER       ID     LABEL    STATUS      KEY     TLOC IP          COLOR            ENCAP  PREFERENCE
-----------------------------------------------------------------------------------------------------
10.0.0.101      22     1008     Inv,U       1       10.0.0.1         biz-internet     ipsec  200
10.0.0.101      1102   1004     R           1       10.0.0.2         mpls             ipsec  -
10.0.0.101      1103   1004     Inv,U       1       10.0.0.2         biz-internet     ipsec  -
10.0.0.101      1108   1008     C,I,R       1       10.0.0.1         mpls             ipsec  200
```

4 paths

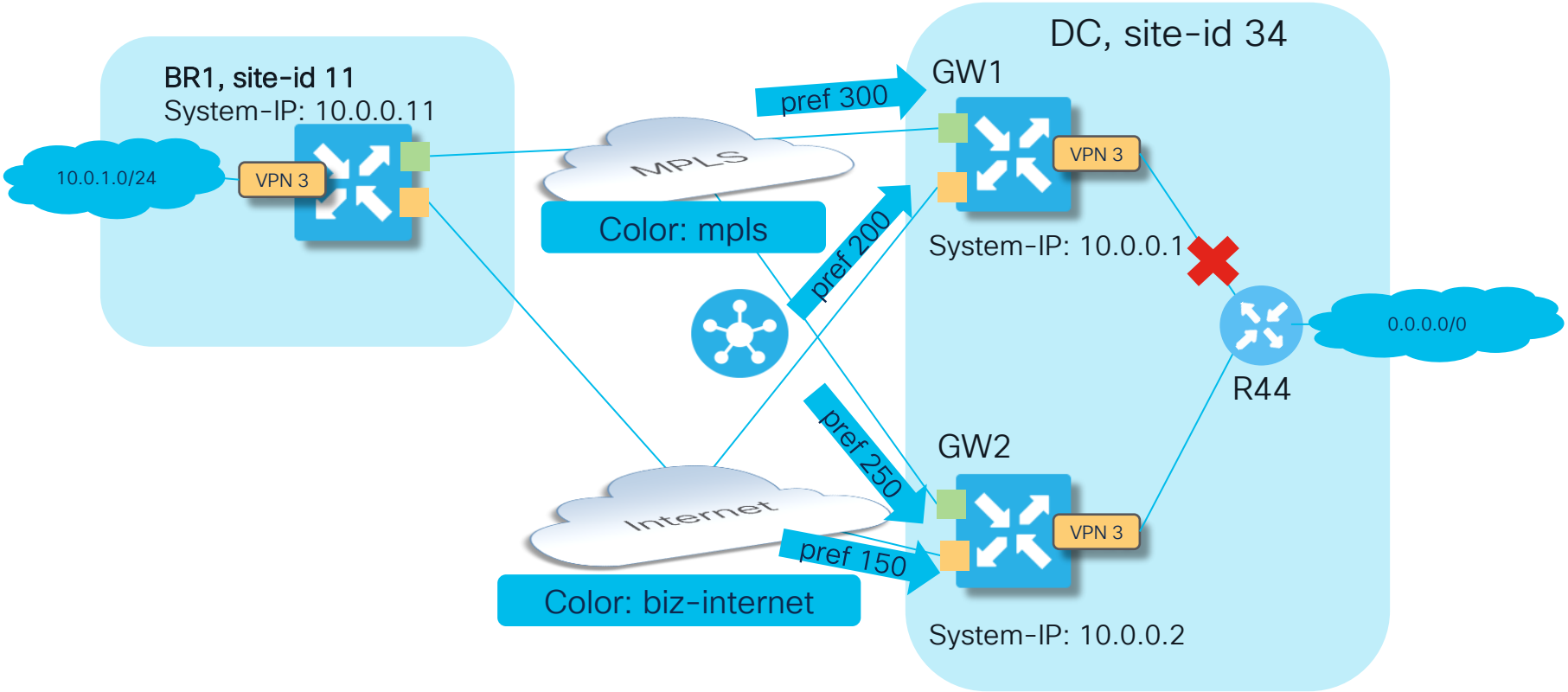
GW1 MPLS link failure scenario, backup GW2 successfully selected:

```
BR1#show sdwan omp routes vpn 3 0.0.0.0/0 | b PATH
                PATH                        PSEUDO
FROM PEER       ID     LABEL    STATUS      KEY     TLOC IP          COLOR            ENCAP  PREFERENCE
-----------------------------------------------------------------------------------------------------
10.0.0.101      22     1008     Inv,U       1       10.0.0.1         biz-internet     ipsec  200
10.0.0.101      1102   1004     C,I,R       1       10.0.0.2         mpls             ipsec  -
10.0.0.101      1103   1004     Inv,U       1       10.0.0.2         biz-internet     ipsec  -
```

Case 2: Multi-level backup preference with “set tloc-list”



Case 2: Multi-level backup preference with ”set tloc-list”



Case 2: Multi-level backup preference with ”set tloc-list” (2)

Centralized control policy on vSmart:

```
policy
 lists
  tloc-list DC_TLOCS_W_PREF
   tloc 10.0.0.1 color mpls encap ipsec preference 300
   tloc 10.0.0.1 color biz-internet encap ipsec preference 200
   tloc 10.0.0.2 color mpls encap ipsec preference 250
   tloc 10.0.0.2 color biz-internet encap ipsec preference 150
  !
 lists
  site-list DCs
   site-id 34
  !
  site-list ALL_BRANCHES
   site-id 11-12
  !
 !
!
```

```
control-policy DC_PREFERENCES
  sequence 10
   match route
    site-list DCs
   !
   action accept
    set
     tloc-list DC_TLOCS_W_PREF
    !
   !
  !
  default-action accept
 !
!
apply-policy
 site-list ALL_BRANCHES
  control-policy DC_PREFERENCES out
 !
!
```

Can you see any problems here?

Case 2: Multi-level backup preference with ”set tloc-list” (3)

Check routing and policy behavior under normal conditions:

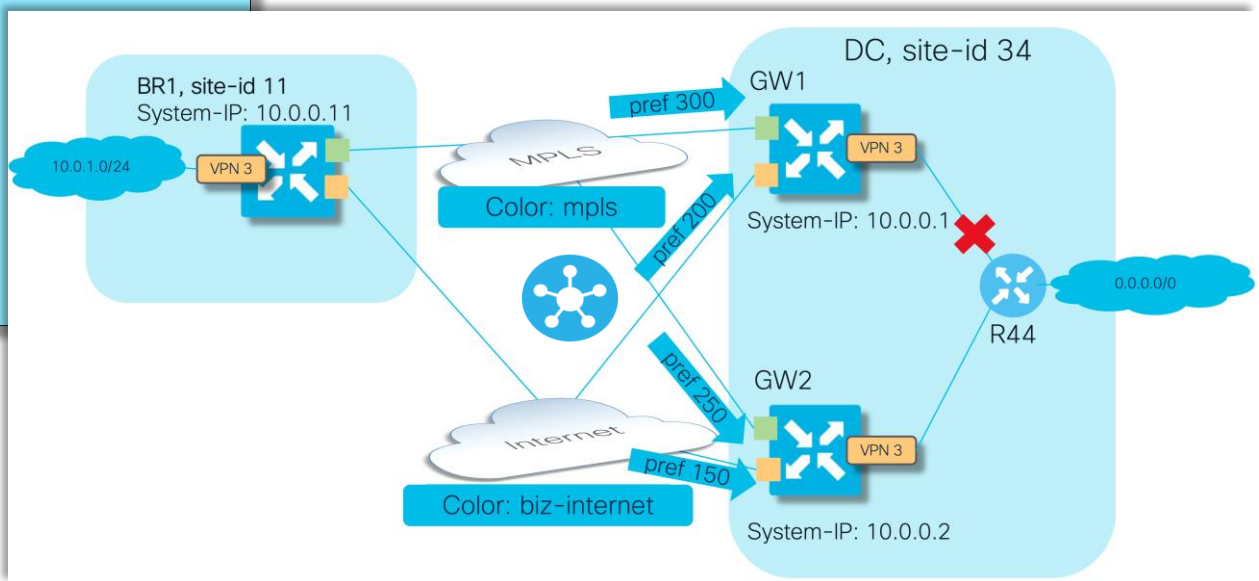
```
BR1#show sdwan omp routes vpn 3 0.0.0.0/0 | b PATH
                PATH                        ATTRIBUTE
FROM PEER       ID     LABEL    STATUS      TYPE       TLOC IP          COLOR            ENCAP  PREFERENCE
-----------------------------------------------------------------------------------------------------------
10.0.0.101      1146   1008     C,I,R       installed  10.0.0.1         mpls             ipsec  300
10.0.0.101      1147   1008     R           installed  10.0.0.1         biz-internet     ipsec  200
10.0.0.101      1148   1004     R           installed  10.0.0.2         mpls             ipsec  250
10.0.0.101      1149   1004     R           installed  10.0.0.2         biz-internet     ipsec  150

BR1#show sdwan policy service-path vpn 3 interface Loopback 3 source-ip 10.0.1.1 dest-ip 10.10.10.10 protocol 6 all
Number of possible next hops: 1
Next Hop: IPsec
  Source: 192.168.9.11 12366 Destination: 192.168.9.13 12426 Local Color: mpls Remote Color: mpls Remote System IP:
10.0.0.1

BR1#show ip route vrf 3 10.10.10.10 resolve

Routing Table: 3
Routing entry for 0.0.0.0/0
  Known via "omp", distance 251, metric 0, candidate default path, type omp
  Last update from 10.0.0.1 on Sdwan-system-intf, 00:02:39 ago
  Routing Descriptor Blocks:
  * 10.0.0.1 (default), from 10.0.0.1, 00:02:39 ago, via Sdwan-system-intf
      Route metric is 0, traffic share count is 1
```

GW1 is preferred and it is the only path to the destination

Case 2: Multi-level backup preference with ”set tloc-list” (4)

Failover testing: GW1 disconnected from the service-side (LAN) segment:

```
GW1#show ip cef vrf 3 10.10.10.10
0.0.0.0/0
  nexthop 10.0.34.44 GigabitEthernet4
GW1#config-t

admin connected from 127.0.0.1 using console on cE3_GW1
GW1(config)#
GW1(config)# interface GigabitEthernet4
GW1(config-if)# shutdown
GW1(config-if)# commit
Commit complete.
GW1(config-if)#end
GW1#show ip cef vrf 3 10.10.10.10
0.0.0.0/0
  no route
GW1#show ip route vrf 3 10.10.10.0

Routing Table: 3
% Subnet not in table
```



Case 2: Multi-level backup preference with ”set tloc-list” (5)

Failover testing (cont.)

Despite that only GW2 now advertises default route and GW1 route was withdrawn from vSmart…

```
vsmart1# show omp routes vpn 3 0.0.0.0/0 received | tab | begin PATH
                 PATH                      ATTRIBUTE
FROM PEER        ID     LABEL    STATUS    TYPE       TLOC IP          COLOR            ENCAP  PREFERENCE
-----------------------------------------------------------------------------------------------------------
10.0.0.2         66     1004     C,R       installed  10.0.0.2         mpls             ipsec  -
10.0.0.2         68     1004     C,R       installed  10.0.0.2         biz-internet     ipsec  -
```

… somehow BR1 still selects GW1 as a preferred path:

```
BR1#show sdwan policy service-path vpn 3 interface Loopback 3 source-ip 10.0.1.1 dest-ip 10.10.10.44 protocol 6 all
Number of possible next hops: 1
Next Hop: IPsec
  Source: 192.168.9.11 12366 Destination: 192.168.9.13 12426 Local Color: mpls Remote Color: mpls Remote System IP:
10.0.0.1

BR1#show ip route vrf 3 10.10.10.10 resolve

Routing Table: 3
Routing entry for 0.0.0.0/0
  Known via "omp", distance 251, metric 0, candidate default path, type omp
  Last update from 10.0.0.1 on Sdwan-system-intf, 00:11:27 ago
  Routing Descriptor Blocks:
  * 10.0.0.1 (default), from 10.0.0.1, 00:11:27 ago, via Sdwan-system-intf
      Route metric is 0, traffic share count is 1
```

Case 2: Multi-level backup preference with ”set tloc-list” (6)

Failover testing (cont.)

Note that GW1 MPLS TLOC is still preferred, but order of paths has changed:

```
BR1#show sdwan omp routes vpn 3 0.0.0.0/0 | b PATH
                  PATH                        PSEUDO
FROM PEER         ID      LABEL    STATUS     KEY     TLOC IP          COLOR            ENCAP  PREFERENCE
-----------------------------------------------------------------------------------------------------------
10.0.0.101        12      1004     R          1       10.0.0.2         mpls             ipsec  250
10.0.0.101        13      1004     R          1       10.0.0.2         biz-internet     ipsec  150
10.0.0.101        31      1008     C,I,R      1       10.0.0.1         mpls             ipsec  300
10.0.0.101        32      1008     R          1       10.0.0.1         biz-internet     ipsec  200
```

Certainly, it leads to blackholing of traffic from BR1 because GW1 has no reachability to LAN anymore:

```
BR1#ping vrf 3 10.10.10.44
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.10.10.44, timeout is 2 seconds:
U.U.U
Success rate is 0 percent (0/5)
```

Case 2: Multi-level backup preference with ”set tloc-list” (7)

Why? This is because vSmart still executes policy as instructed and sets route preference + TLOC:

```
vsmart1# show omp routes vpn 3 0.0.0.0/0 advertised detail | nomore | exclude not\ set | begin 10.0.0.11
```

```
peer    10.0.0.11
    Attributes:
     originator       10.0.0.2
     label            1004
     path-id          12
     tloc             10.0.0.2, mpls, ipsec
     site-id          34
     overlay-id        1
     preference       250
     origin-proto     static
     origin-metric    0
    Attributes:
     originator       10.0.0.2
     label            1004
     path-id          13
     tloc             10.0.0.2, biz-internet, ipsec
     site-id          34
     overlay-id        1
     preference       150
     origin-proto     static
     origin-metric    0
    Attributes:
     originator       10.0.0.2
     label            1008
     path-id          31
     tloc             10.0.0.1, mpls, ipsec
     site-id          34
     overlay-id        1
     preference       300
     origin-proto     static
     origin-metric    0
    Attributes:
     originator       10.0.0.2
     label            1008
     path-id          32
     tloc             10.0.0.1, biz-internet, ipsec
     site-id          34
     overlay-id        1
     preference       200
     origin-proto     static
     origin-metric    0
```

*Note that originator is always 10.0.0.2 (GW2)

Case 2: Multi-level backup preference with ”set tloc-list” (8)

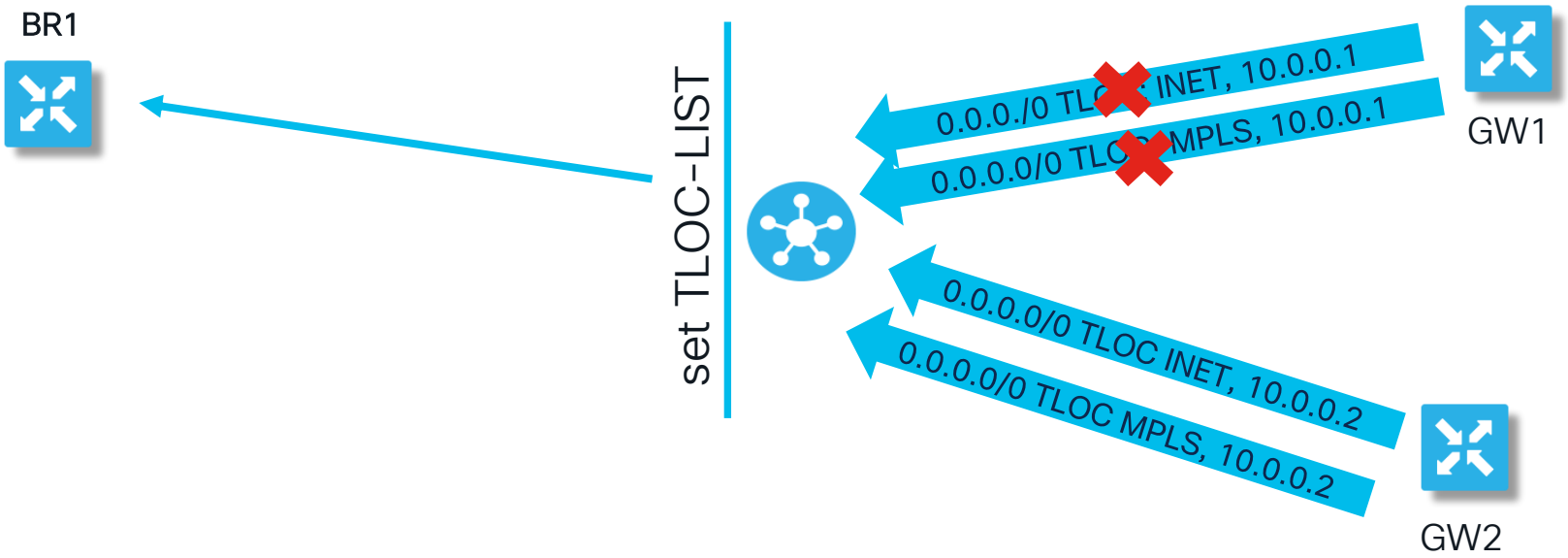
Recap the original control policy.

```
policy
 lists
  tloc-list DC_TLOCS_W_PREF
   tloc 10.0.0.1 color mpls encap ipsec preference 300
   tloc 10.0.0.1 color biz-internet encap ipsec preference
200
   tloc 10.0.0.2 color mpls encap ipsec preference 250
   tloc 10.0.0.2 color biz-internet encap ipsec preference
150
  !
 lists
  site-list DCs
   site-id 34
  !
  site-list ALL_BRANCHES
   site-id 11-12
  !
 !
!
```

```
control-policy DC_PREFERENCES
  sequence 10
   match route
    site-list DCs
   !
   action accept
    set
     tloc-list DC_TLOCS_W_PREF
    !
   !
  !
  default-action accept
 !
!
apply-policy
 site-list ALL_BRANCHES
  control-policy DC_PREFERENCES out
 !
!
```

Case 2: Multi-level backup preference with ”set tloc-list” (9)

What caused the problem?



GW1 has stopped advertising default routes, yet vSmart continues to replace the route's TLOC with a list that includes GW1 TLOCs due to the policy. This is the expected behaviour according to the configured policy logic.

Case 2: Multi-level backup preference with ”set tloc-list” (10)

Typical solution: Set TLOC preference conditionally and only if received route has corresponding TLOC :

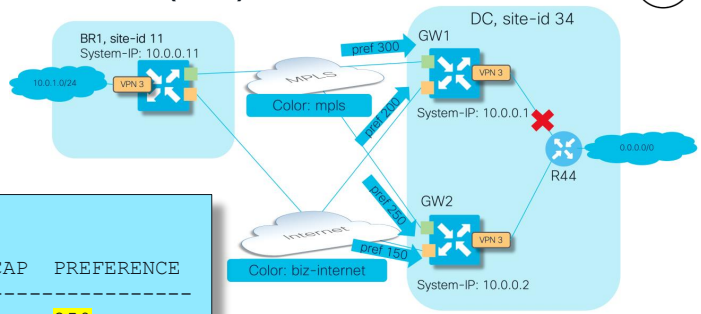
```
policy
 lists
  tloc-list GW1_TLOCS
   tloc 10.0.0.1 color mpls encap ipsec
   tloc 10.0.0.1 color biz-internet encap ipsec
  !
  tloc-list GW1_TLOCS_W_PREF
   tloc 10.0.0.1 color mpls encap ipsec preference 300
   tloc 10.0.0.1 color biz-internet encap ipsec preference 200
  !
  tloc-list GW2_TLOCS
   tloc 10.0.0.2 color mpls encap ipsec
   tloc 10.0.0.2 color biz-internet encap ipsec
  !
  tloc-list GW2_TLOCS_W_PREF
   tloc 10.0.0.2 color mpls encap ipsec preference 250
   tloc 10.0.0.2 color biz-internet encap ipsec preference 150
  !
```

```
apply-policy
 site-list ALL_BRANCHES
  control-policy DC_PREFERENCES_FIX out
 !
!
```

```
control-policy DC_PREFERENCES_FIX
 sequence 10
  match route
   site-list DCs
   tloc-list GW1_TLOCS
  !
  action accept
   set
    tloc-list GW1_TLOCS_W_PREF
   !
  !
 !
 sequence 20
  match route
   site-list DCs
   tloc-list GW2_TLOCS
  !
  action accept
   set
    tloc-list GW2_TLOCS_W_PREF
   !
  !
 !
 default-action accept
!
!
```

* Unlike some other available solutions, this is the best one because it won’t lead to suboptimal routing

Case 2: Multi-level backup preference with ”set tloc-list” (11)



Testing the solution when GW1 experiences a LAN link failure:

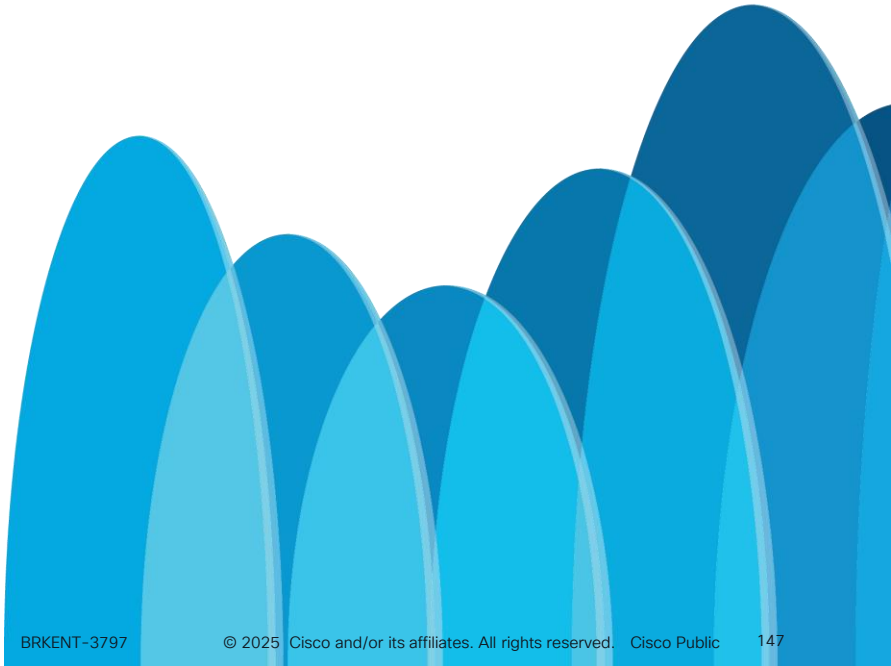
```
BR1#show sdwan omp routes vpn 3 0.0.0.0/0 | begin PATH
                  PATH                        PSEUDO
FROM PEER         ID      LABEL    STATUS     KEY     TLOC IP          COLOR            ENCAP  PREFERENCE
-----------------------------------------------------------------------------------------------------------
10.0.0.101        1166    1004     C,I,R      1       10.0.0.2         mpls             ipsec  250
10.0.0.101        1167    1004     R          1       10.0.0.2         biz-internet     ipsec  150
```

Note that there are only 2 paths remain and GW2 MPLS path is preferred:

```
BR1#show sdwan policy service-path vpn 3 interface Loopback 3 source-ip 10.0.1.1 dest-ip 10.10.10.44 protocol 6 all
Number of possible next hops: 1
Next Hop: IPsec
  Source: 192.168.9.11 12366 Destination: 192.168.9.14 12406 Local Color: mpls Remote Color: mpls Remote System IP:
10.0.0.2


BR1#ping vrf 3 10.10.10.44
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.10.10.44, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
```

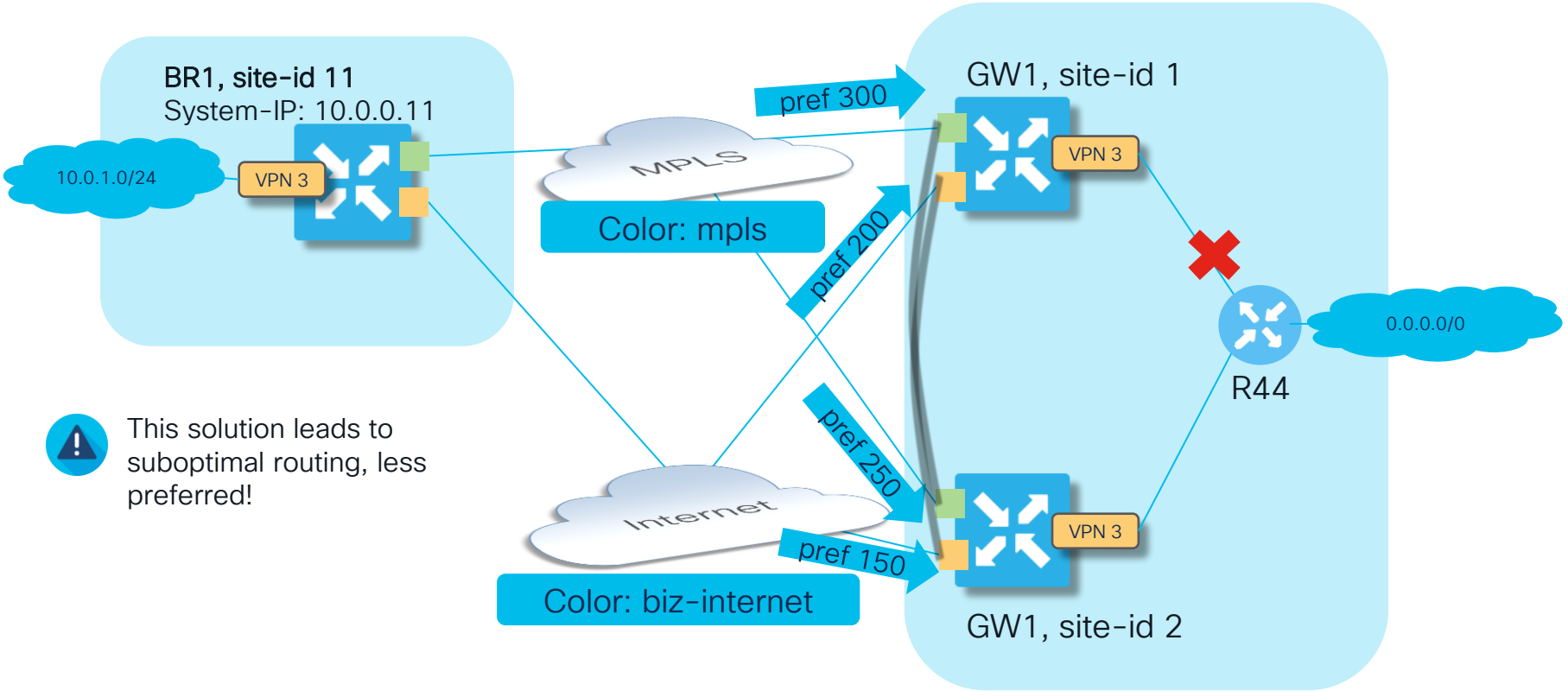
So, failover works as intended

Case 2. Other possible solutions for sake of having complete picture



Case 2: Multi-level backup preference with ”set tloc-list” (12)

Solution 2. Assign different site-ids to GW1/GW2 or configure **`allow-same-site-tunnels`**



Case 2: Multi-level backup preference with ”set tloc-list” (13)

Solution 2. Assign different site-ids to GW1/GW2 or configure **`allow-same-site-tunnels`**.

- As a result GW1/GW2 will build data plane tunnels between them
- GW1 will install GW2’s default OMP route into RIB/FIB in case of LAN link failure.

```
GW1#show sdwan omp routes vpn 3 0.0.0.0/0 | b PATH
                  PATH                        PSEUDO
FROM PEER         ID     LABEL    STATUS      KEY     TLOC IP          COLOR            ENCAP  PREFERENCE
-----------------------------------------------------------------------------------------------------------
10.0.0.101        200    1004     C,I,R       1       10.0.0.2         mpls             ipsec  -
10.0.0.101        201    1004     C,I,R       1       10.0.0.2         biz-internet     ipsec  -
10.0.0.102        194    1004     C,R         1       10.0.0.2         mpls             ipsec  -
10.0.0.102        195    1004     C,R         1       10.0.0.2         biz-internet     ipsec  -

GW1#sh ip route vrf 3 0.0.0.0

Routing Table: 3
Routing entry for 0.0.0.0/0, supernet
  Known via "omp", distance 251, metric 0, candidate default path, type omp
  Last update from 10.0.0.2 on Sdwan-system-intf, 00:09:50 ago
  Routing Descriptor Blocks:
  * 10.0.0.2 (default), from 10.0.0.2, 00:09:50 ago, via Sdwan-system-intf
      Route metric is 0, traffic share count is 1
```

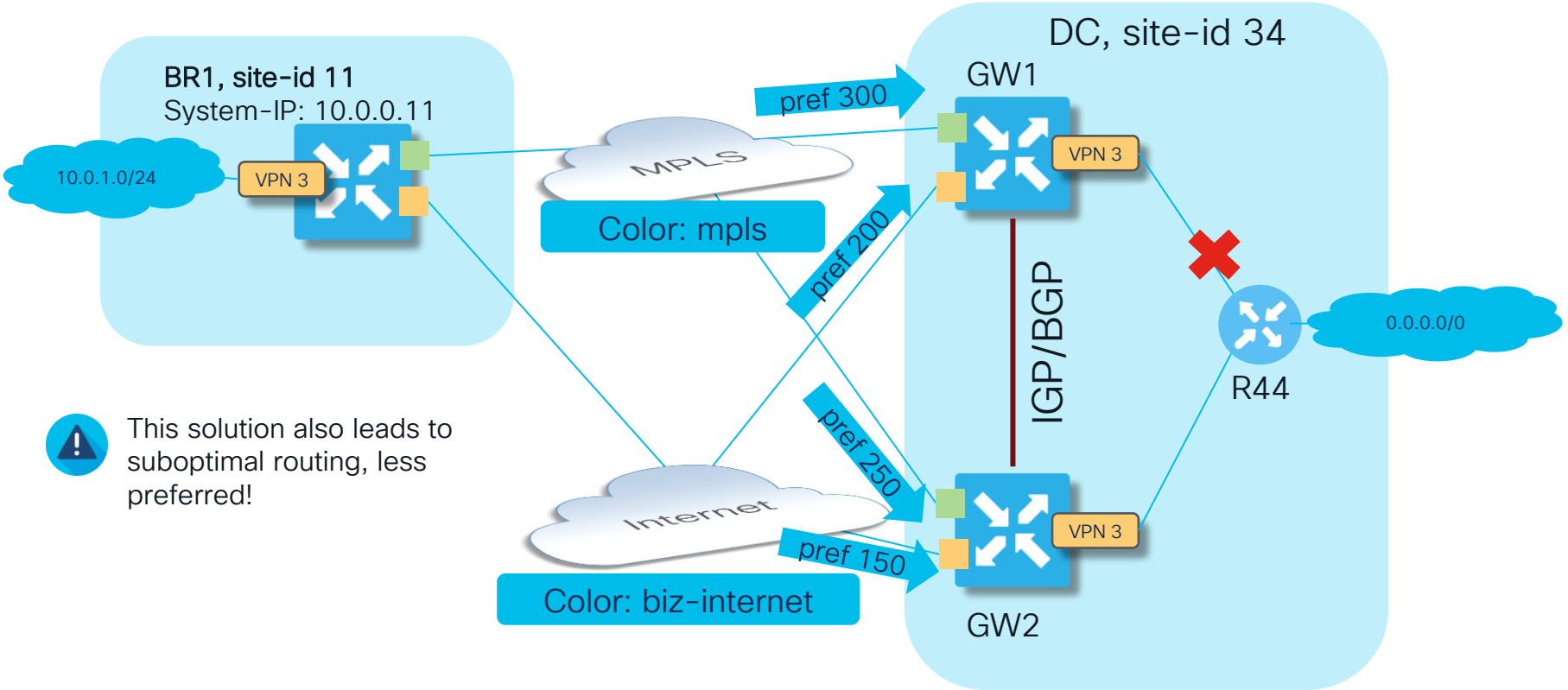
But traffic path from BR1 will be suboptimal because GW1 still preferred as per policy:

```
BR1#traceroute vrf 3 10.10.10.44
Type escape sequence to abort.
Tracing the route to 10.10.10.44
VRF info: (vrf in name/id, vrf out name/id)
  1 192.168.9.13 1 msec 0 msec 1 msec
  2 192.168.10.14 1 msec 0 msec 1 msec
  3 10.0.34.44 2 msec *  2 msec
```

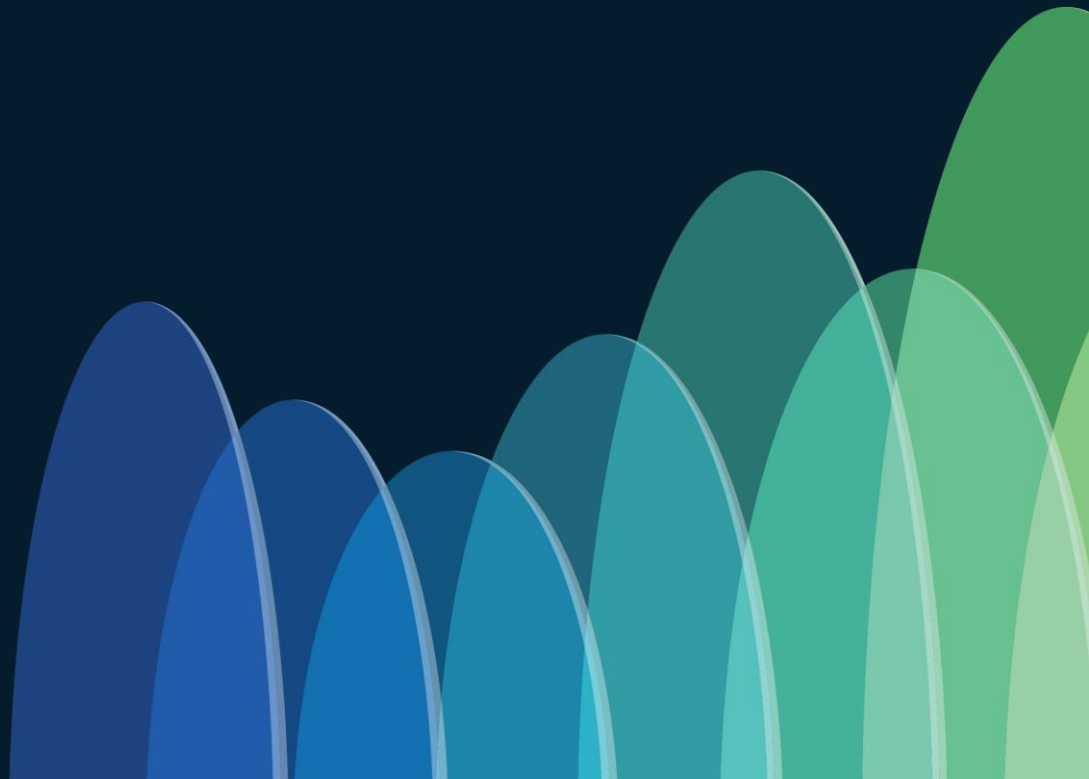
GW1

Case 2: Multi-level backup preference with ”set tloc-list” (14)

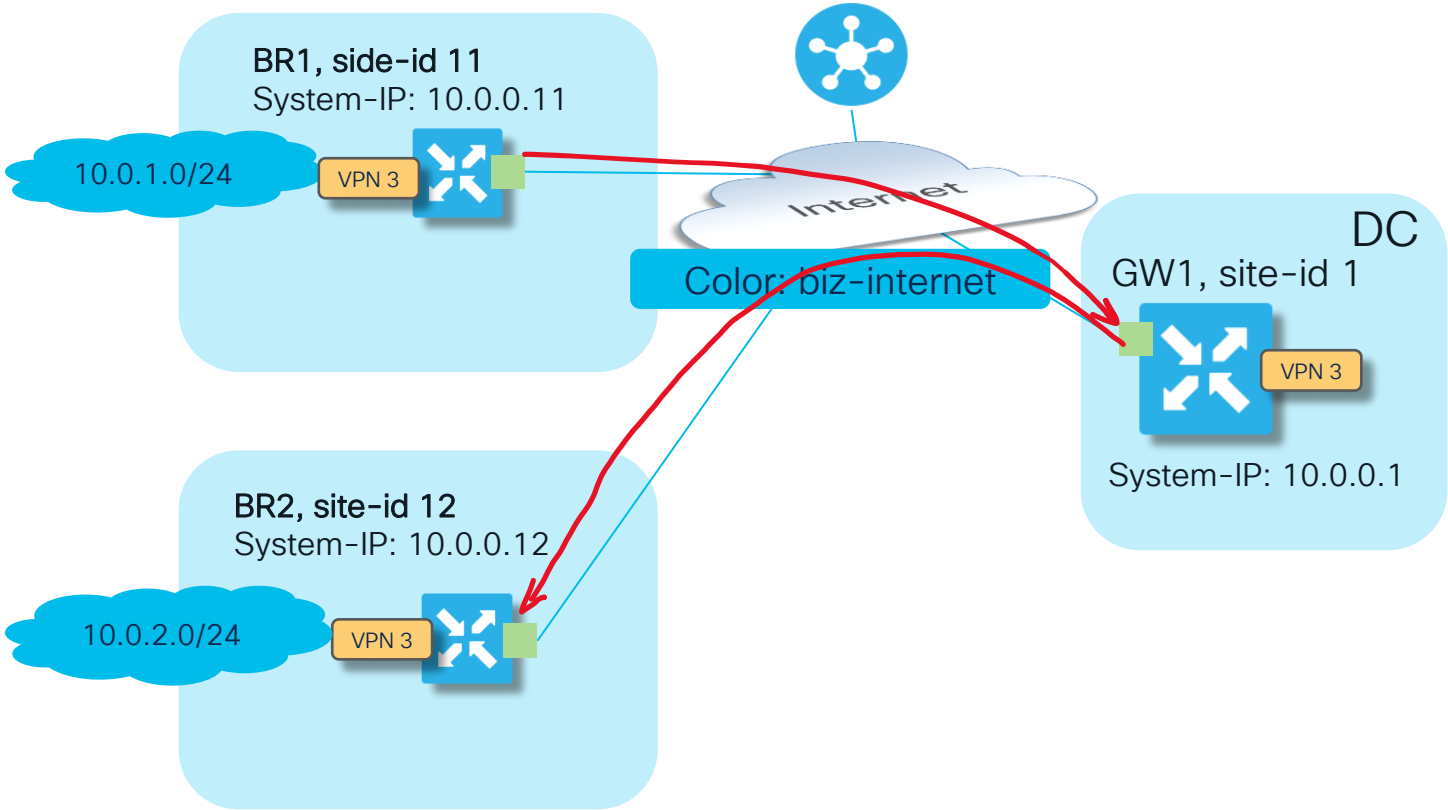
Solution 3. Similar to previous, but introduce additional link and IGP/BGP between GW1/GW2.



Case 3. Traffic engineering with “set tloc-action"



Case 3. Traffic engineering with “set tloc-action"

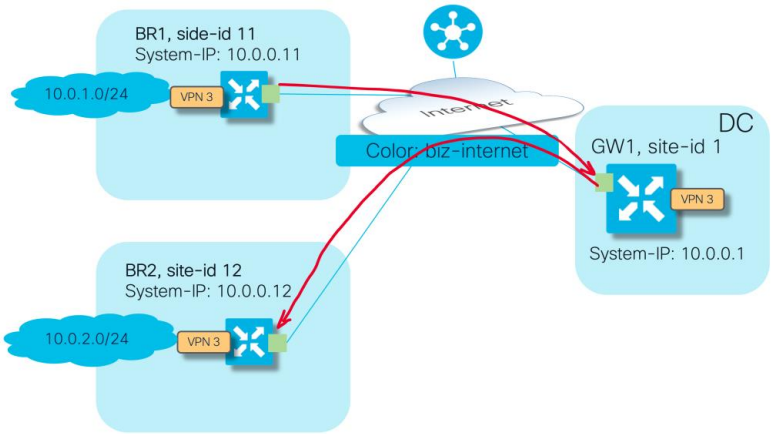


Aim here is to steer traffic from BR1 to BR2 via GW1

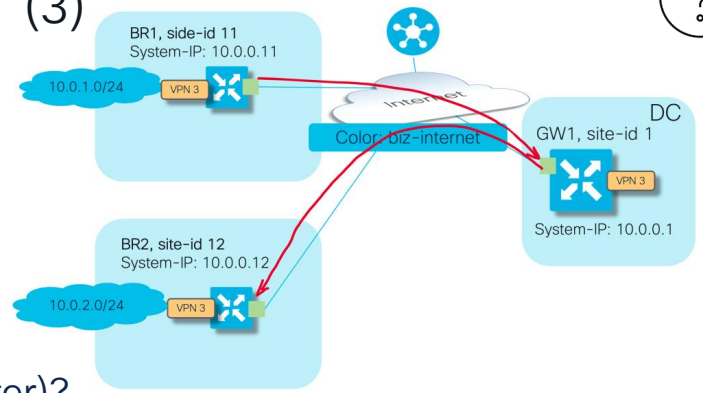
Case 3. Traffic engineering with “set tloc-action” (2)

vSmart policy to enforce traffic path via GW1:

```
policy
 lists
  site-list ALL_BRANCHES
   site-id 11-12
 !
 control-policy REDIRECT_VIA_GW1
  sequence 10
   match route
    site-list ALL_BRANCHES
   !
   action accept
    set
     tloc-action primary
     tloc 10.0.0.1 color biz-internet encap ipsec
    !
   !
  !
  default-action accept
 !
!
apply-policy
 site-list ALL_BRANCHES
  control-policy REDIRECT_VIA_GW1 out
 !
!
```



Case 3. Traffic engineering with “set tloc-action” (3)



Testing and the problem.

Why does the traffic take a direct path (skips intermediate router)?

```
BR1#traceroute vrf 3 10.0.2.2
Type escape sequence to abort.
Tracing the route to 10.0.2.2
VRF info: (vrf in name/id, vrf out name/id)
  1 10.0.2.2 1 msec *  2 msec
```

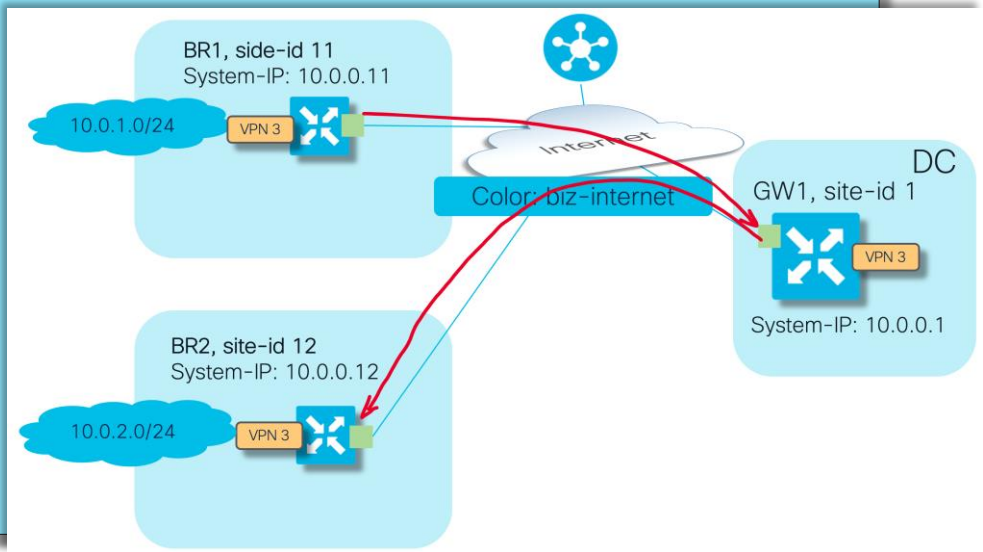
Why is there only one path available which points directly to BR2?

```
BR1#show sdwan policy service-path vpn 3 interface Loopback 3 source-ip 10.0.1.1 dest-ip 10.0.2.2 protocol 6 all
Number of possible next hops: 1
Next Hop: IPsec
  Source: 192.168.10.11 12366 Destination: 192.168.10.12 12366 Local Color: biz-internet Remote Color: biz-internet Remote
System IP: 10.0.0.12
```

Case 3. Traffic engineering with “set tloc-action” (4)

Let’s check OMP routes on BR1 (unimportant attributes excluded)

```
cE1_BR1#show sdwan omp routes 10.0.2.0/24 detail | exclude not\ set|metric|overlay|label|path-id|origin
----------------------------------------------------
omp route entries for vpn 3 route 10.0.2.0/24
----------------------------------------------------
            RECEIVED FROM:
peer            10.0.0.101
status          C,I,R
    Attributes:
     originator       10.0.0.12
     type             installed
     tloc             10.0.0.12, biz-internet, ipsec
     site-id          12
            RECEIVED FROM:
peer            10.0.0.101
status          Inv,U
loss-reason     invalid
lost-to-peer    10.0.0.101
    Attributes:
     originator       10.0.0.12
     type             installed
     tloc             10.0.0.1, biz-internet, ipsec
     ultimate-tloc    10.0.0.12, biz-internet, ipsec -- primary
     site-id          12
```



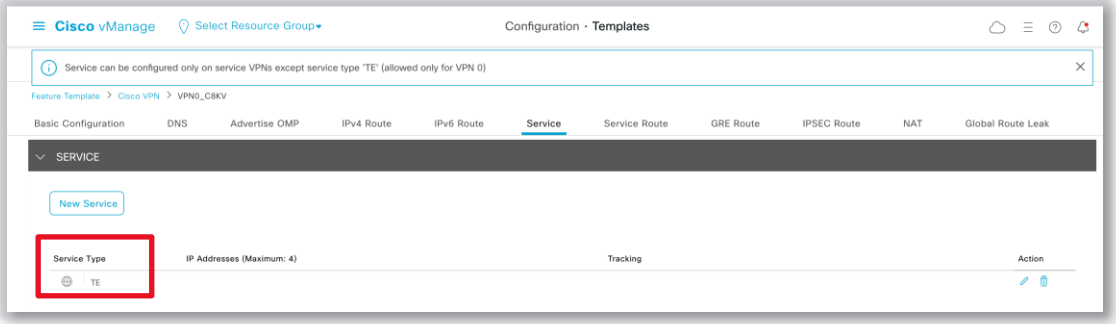
- Note that second “traffic-engineering” path via GW1 is invalid and unresolved. It also has unusual attribute called “**ultimate-tloc**”
- An “**ultimate-tloc**” is the target TLOC that an intermediate hop, such as GW1, uses to establish a data plane tunnel to reach the final destination, BR2 in this scenario

Case 3. Traffic engineering with “set tloc-action” (5)

This is an example of a solution you must be familiar with and know beforehand. The issue caused by misconfiguration (specifically, missing mandatory config):

- Rule: if the action is **“set tloc-action”**, you must configure **“service TE”** in the global VRF on the intermediate router

```
GW1#sh sdwan running-config "sdwan service"
sdwan
 service TE vrf global
  !
!

GW1#show sdwan omp services | include TE
GW1#
```



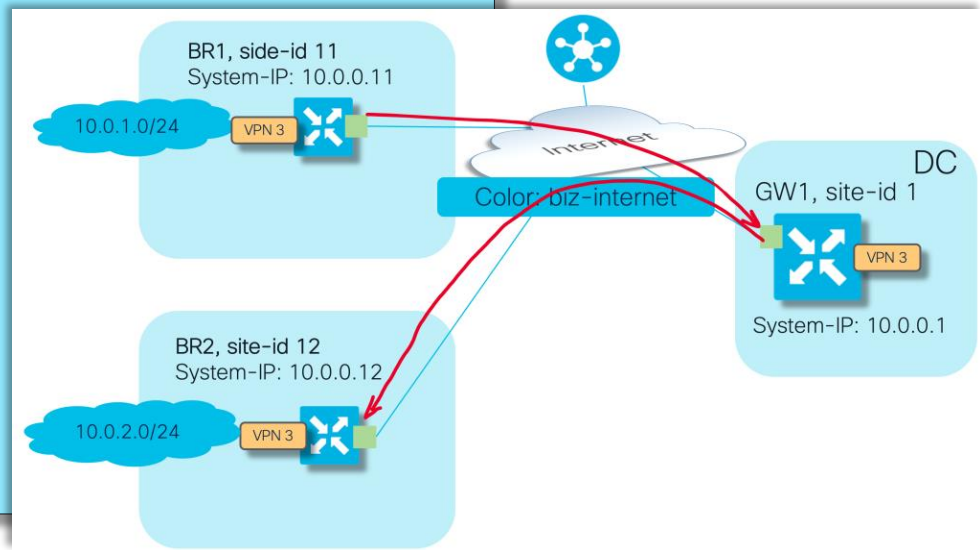
- To add complexity: It cannot be seen with **show sdwan omp services** command
- By the way: same config is also a pre-requisite for dynamic on-demand tunnels (ODT) to function correctly

Case 3. Traffic engineering with “set tloc-action” (6)

Testing the solution. Route with ultimate-tloc now selected and traffic goes via GW1 as desired:

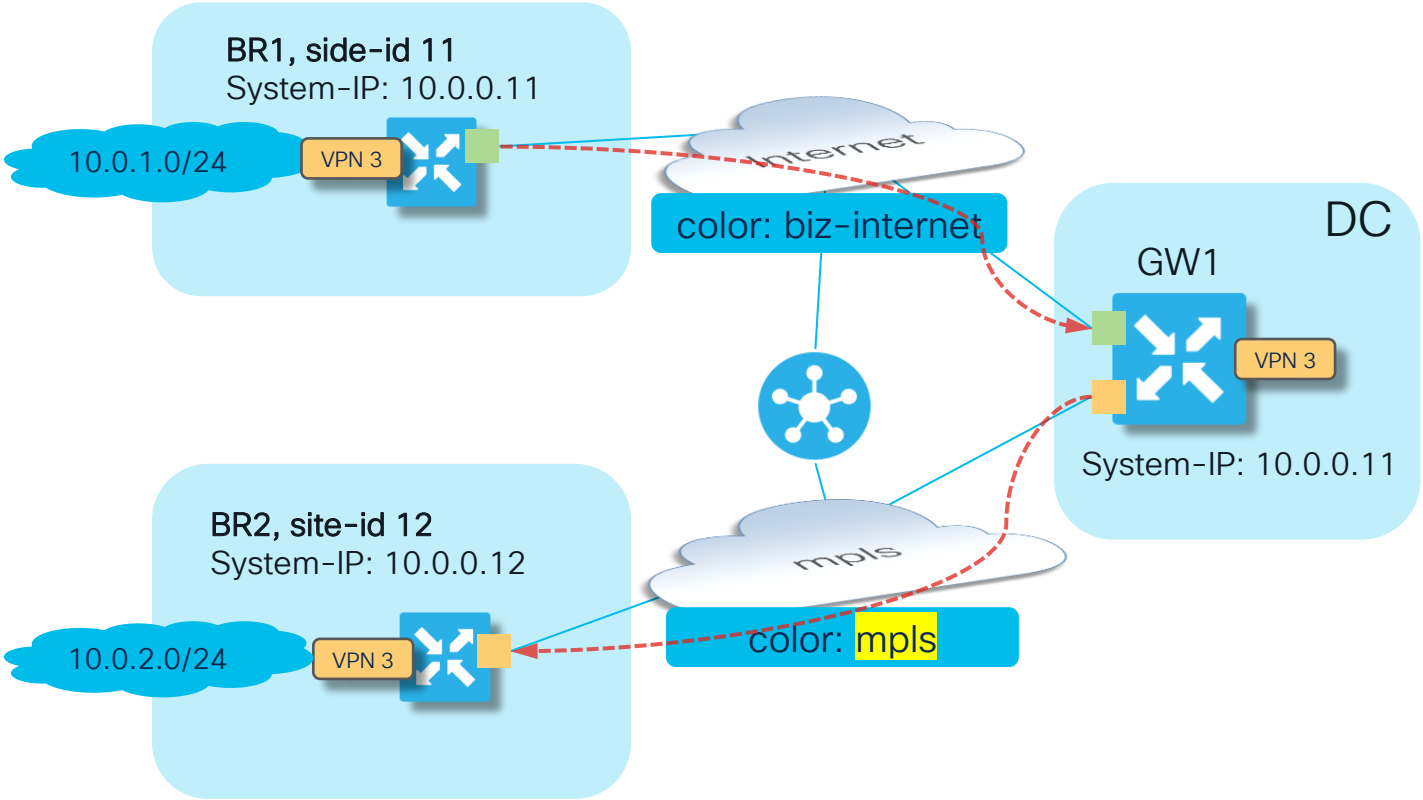
```
BR1#show sdwan omp routes 10.0.2.0/24 detail | exclude not\ set|metric|overlay|label|path-id|origin-
----------------------------------------------------
omp route entries for vpn 3 route 10.0.2.0/24
----------------------------------------------------
            RECEIVED FROM:
peer            10.0.0.101
status          R
loss-reason     tloc-action
lost-to-peer    10.0.0.101
    Attributes:
     originator       10.0.0.12
     type             installed
     tloc             10.0.0.12, biz-internet, ipsec
     site-id          12
            RECEIVED FROM:
peer            10.0.0.101
status          C,I,R
    Attributes:
     originator       10.0.0.12
     type             installed
     tloc             10.0.0.1, biz-internet, ipsec
     ultimate-tloc    10.0.0.12, biz-internet, ipsec -- primary
     site-id          12

BR1#traceroute vrf 3 10.0.2.2
Type escape sequence to abort.
Tracing the route to 10.0.2.2
VRF info: (vrf in name/id, vrf out name/id)
  1 192.168.10.13 1 msec 0 msec 1 msec
  2 10.0.2.2 1 msec *  2 msec
```

GW1



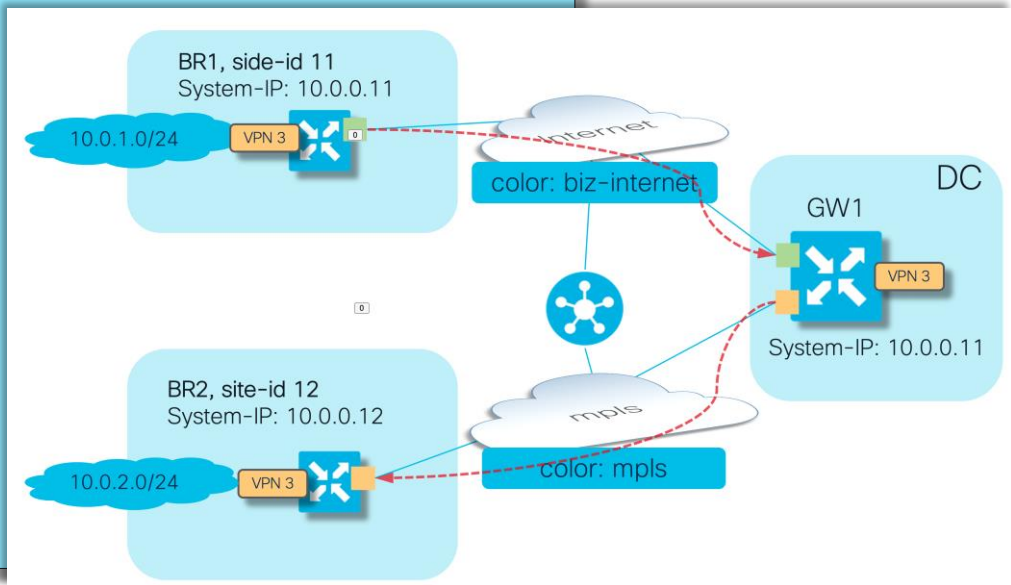
Case 3b. Traffic engineering with “set tloc-action". Disjoined underlay.

Same control policy, but BR2 connected to a different transport.



Case 3b. Traffic engineering with “set tloc-action". Disjoined underlay (2)

```
BR1#show sdwan omp routes 10.0.2.0/24 detail | exclude not\ set|metric|overlay|label|origin-|site|type
----------------------------------------------------
omp route entries for vpn 3 route 10.0.2.0/24
----------------------------------------------------
            RECEIVED FROM:
peer            10.0.0.101
path-id         1
status          Inv,U
loss-reason     tloc-action
lost-to-peer    10.0.0.101
lost-to-path-id 2
    Attributes:
     originator       10.0.0.12
     tloc             10.0.0.12, mpls, ipsec
            RECEIVED FROM:
peer            10.0.0.101
path-id         2
status          Inv,U
    Attributes:
     originator       10.0.0.12
     tloc             10.0.0.1, biz-internet, ipsec
     ultimate-tloc    10.0.0.12, mpls, ipsec -- primary
```



- Path 1 is unresolved because underlay is disjoined (no data plane tunnels with BR2, expected)
- But why is path 2 unresolved and invalid?

Case 3b. Traffic engineering with “set tloc-action". Disjoined underlay (3)

Path 2 is unresolved and invalid because different colors can not be joined with tloc-action

Rule: tloc-action is only supported end-to-end if the transport color is the same from a site to the intermediate hop and from the intermediate hop to the final (ultimate) destination.

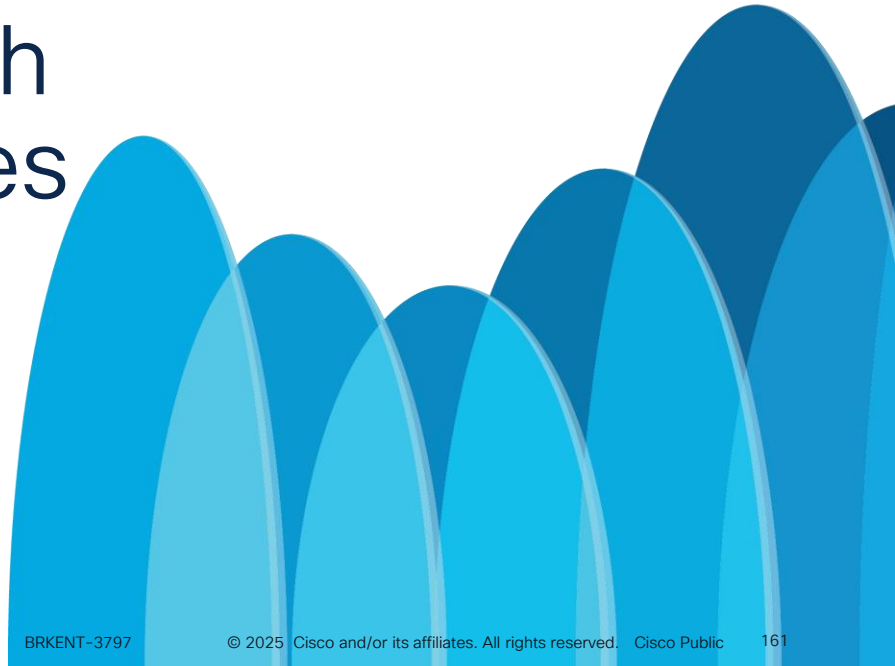
If the transport used to get to the intermediate hop from a site is a different color than the transport used from the intermediate hop to get to the final (ultimate) destination, then this will cause a policy failure with tloc-action.

Reference:

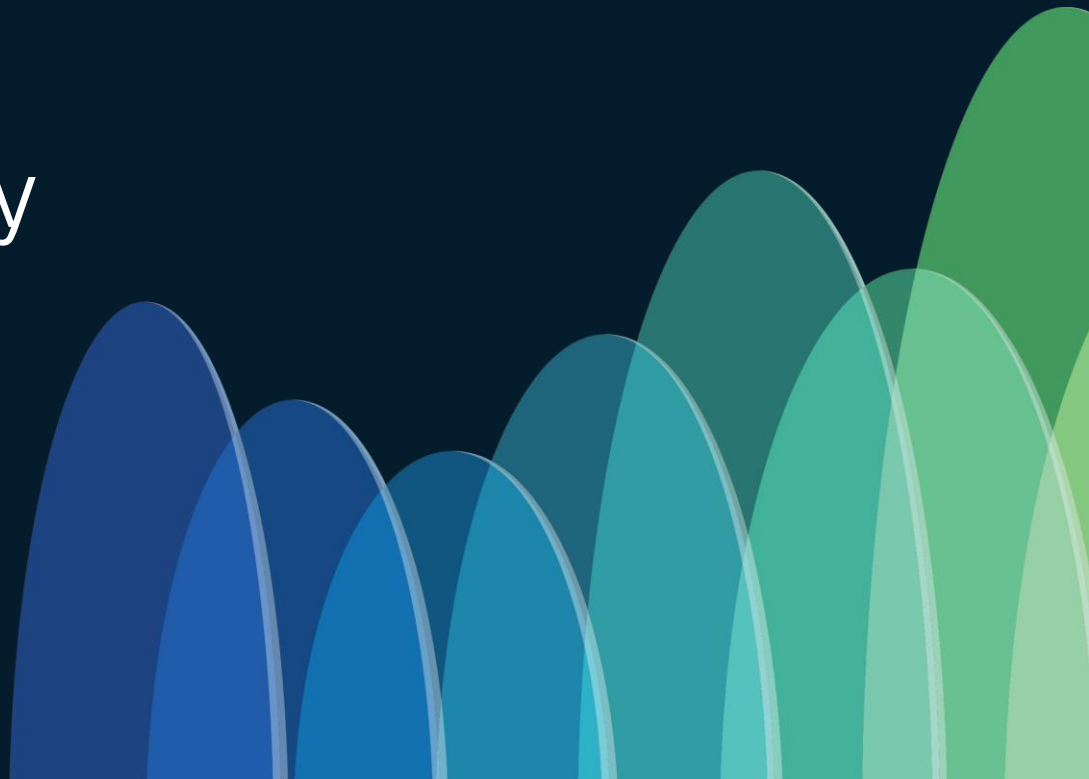
https://www.cisco.com/c/en/us/td/docs/routers/sdwan/command/sdwan-cr-book/config-cmd.html#r_action_1267.xml

Enhancement request: CSCvr80957

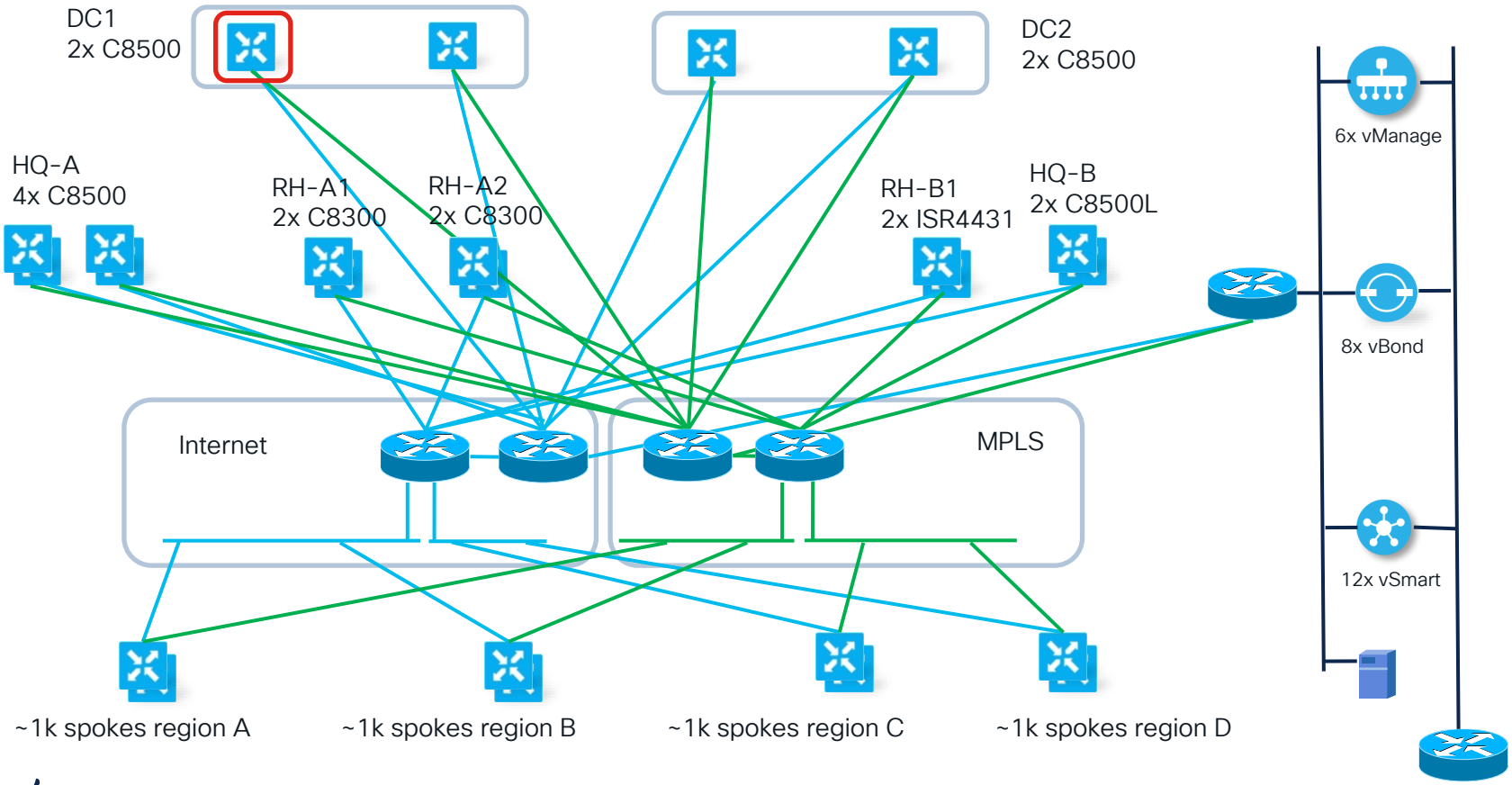
Interesting cases with Data and AAR policies



Case 4. Device can’t install policy after reload



Case 4. Device can’t install policy after reload (1)



Case 4. Device can’t install policy after reload (2)

Symptoms:

- Hub router was reloaded to perfrom software upgrade
- After the upgrade, device does not install any policy anymore
- Hub router was sucessfully downgraded to exclude possibility of a software defect with the same result – no policy installed on the device

Case 4. Device can’t install policy after reload (3)

Toubleshooting from vSmart side
OMP peering established and stable (hence underlying control connection as well):

```
vsmart1# show omp peers 10.0.0.101
R -> routes received
I -> routes installed
S -> routes sent

                        DOMAIN    OVERLAY   SITE
PEER            TYPE    ID        ID        ID        STATE    UPTIME           R/I/S
-------------------------------------------------------------------------------------
10.0.0.101      vedge   1         1         12        up       0:00:18:13       10/0/2
```

And policy assigned properly on vSmart:

```
vSmart1# show support omp peer peer-ip 1.1.1.101 | include -pol
        site-pol: STL_DC,STL_DC_1 route-pol-in: None route-pol-out: CPL_DC_1 data-pol-in: _VPN_LIST10_QOS_MARKING
           data-pol-out: None pfr-pol: _VPN_LIST10_APP_Route1 mem-pol: None cflowd:None
<data-policy>
<direction>from-service</direction></data-policy><app-route-policy>
</app-route-policy>
```

Case 4. Device can’t install policy after reload (4)

And translated to XML properly:

```
vsmart1# show support omp peer peer-ip 10.0.0.101 | begin "Policy received" | until "Statistics"
 Policy received: Complete
       Forwarding policy len: 4981
<data-policy>
  <name>_VPN_LIST10_QOS_MARKING</name>
  <vpn-list>
    <name>VPN_LIST10</name>
    <sequence>
      <seq-value>1</seq-value>
      <match>
        <destination-port>5000</destination-port>
      </match>
      <action>
        <action-value>accept</action-value>
        <set>
          <dscp>46</dscp>
          <forwarding-class>Queue0</forwarding-class>
        </set>
      </action>
<app-route-policy>
  <name>_VPN_LIST10_APP_Route1</name>
  <vpn-list>
    <name>VPN_LIST10</name>
    <sequence>
      <seq-value>1</seq-value>
      <match>
        <source-ip>0.0.0.0/0</source-ip>
        <destination-port>5000</destination-port>
      </match>
      <action>
        <sla-class>
          <sla-class-name>SLA_CLASS1</sla-class-name>
          <preferred-color>mpls</preferred-color>
        </sla-class>
      </action>
    </sequence>
<…>
 Statistics:
```

Case 4. Device can’t install policy after reload (5)

Troubleshooting from device side:

1. No policy chages, just hub router was reloaded so we are not checking commit changes
2. Control connnetions are up and mostly stable, but not all of them:

```
DC1_101#show sdwan control connections
                                                  PEER            PEER                                    CONTROLLER
PEER    PEER PEER      SITE  DOMAIN PEER          PRIV  PEER      PUB                                     GROUP
TYPE    PROT SYSTEM IP ID    ID     PRIVATE IP    PORT  PUBLIC IP PORT  ORGANIZATION      LOCAL COLOR   PROXY STATE UPTIME      ID
-------------------------------------------------------------------------------------------------------------------------------
vsmart  dtls 1.1.1.20  1     1      10.50.1.20    12346 10.50.1.20 12346 OrgName 1 - 31337  biz-internet  No    up      0:00:19:03 1
vsmart  dtls 1.1.1.21  1     1      10.50.1.21    12346 10.50.1.21 12346 OrgName 1 - 31337  biz-internet  No    up      0:00:03:35 2
vsmart  dtls 1.1.1.21  1     1      10.50.1.21    12346 10.50.1.21 12346 OrgName 1 - 31337  mpls          No    up      0:00:08:38 2
vsmart  dtls 1.1.1.27  1     1      10.50.1.27    12346 10.50.1.27 12346 OrgName 1 - 31337  mpls          No    up      0:00:08:35 8
vbond   dtls 0.0.0.0   0     0      10.50.1.10    12346 10.50.1.10 12346 OrgName 1 - 31337  biz-internet  -     up      0:00:26:19 0
vbond   dtls 0.0.0.0   0     0      10.50.1.13    12346 10.50.1.13 12346 OrgName 1 - 31337  mpls          -     up      0:00:23:20 0
vmanage dtls 1.1.1.4   1     0      10.50.1.4     12346 10.50.1.4  12346 OrgName 1 - 31337  biz-internet  No    up      0:00:19:06 0
```

3. OMP peering is stable (not really necessary to check because it is stable from vSmart perspective):

```
DC1_101#show sdwan omp peers
R -> routes received
I -> routes installed
S -> routes sent

TENANT                          DOMAIN    OVERLAY   SITE      REGION
ID        PEER              TYPE    ID        ID        ID        ID        STATE    UPTIME           R/I/S
-------------------------------------------------------------------------------------------------------------
0         1.1.1.20          vsmart  1         1         1         None      up       0:00:18:17       0/0/10
0         1.1.1.26          vsmart  1         1         1         None      up       0:00:14:26       59742/22939/10
```

Case 4. Device can’t install policy after reload (6)

But the mystery is that device still does not have any policy:

```
DC1_101#show sdwan omp summary | include policy
policy-sent             0
policy-received         0
```

And certainly other commands confirm the same:

```
DC1_101#show sdwan from-vsmart commit-history
summary

DC1_101#show sdwan policy from-vsmart
% No entries found.
```

From the logs it says no policy assigned and seems other vSmarts are less stable (*hint!*):

```
Mar 16 12:17:21.268: %Cisco-SDWAN-DC1_101-OMPD-3-ERRO-400002: vSmart peer 1.1.1.21 state changed to Init
Mar 16 12:17:21.268: %Cisco-SDWAN-DC1_101-OMPD-6-INFO-400005: Number of vSmarts connected : 2
Mar 16 12:17:23.268: %Cisco-SDWAN-DC1_101-OMPD-6-INFO-400007: Using empty policy from peer 1.1.1.20
```

Case 4. Device can’t install policy after reload (7)

What if something strange happens on a device?
Then always check QFP drop counters first of all:

```
DC1_101#show platform hardware qfp active statistics drop clear
Last clearing of QFP drops statistics : Thu Mar 16 13:20:11 2023


---------------------------------------------------------------------
Global Drop Stats                          Packets                Octets
---------------------------------------------------------------------
Disabled                                         2                   506
Ipv6NoRoute                                      1                    56
Nat64v6tov4                                      6                   480
PuntPerCausePolicerDrops                   8504352            1625710362
SdwanImplicitAclDrop                          2844                451300


DC1_101#show platform hardware qfp active statistics drop detail
Last clearing of QFP drops statistics : Thu Mar 16 13:20:11 2023
 (13s ago)

-----------------------------------------------------------------------------
   ID  Global Drop Stats                          Packets                Octets
-----------------------------------------------------------------------------
   206 PuntPerCausePolicerDrops                   49419                9442474
```

Case 4. Device can’t install policy after reload (8)

Then you can use packet-trace to see dropped packets detals:

```
DC1_101#debug platform condition both
DC1_101#debug platform packet-trace drop code 206
DC1_101#debug platform packet-trace packet 1024
 Please remember to turn on 'debug platform condition start' for packet-trace to work
DC1_101#debug platform condition start


DC1_101#show platform packet-trace summary
Pkt   Input                     Output                    State  Reason
1     Te0/0/0                   internal0/0/rp:0          DROP   206 (PuntPerCausePolicerDrops)
2     Te0/0/1                   internal0/0/rp:0          DROP   206 (PuntPerCausePolicerDrops)
3     Te0/0/1                   internal0/0/rp:0          DROP   206 (PuntPerCausePolicerDrops)
4     Te0/0/0                   internal0/0/rp:0          DROP   206 (PuntPerCausePolicerDrops)
5     Te0/0/1                   internal0/0/rp:0          DROP   206 (PuntPerCausePolicerDrops)
6     Te0/0/1                   internal0/0/rp:0          DROP   206 (PuntPerCausePolicerDrops)
7     Te0/0/0                   internal0/0/rp:0          DROP   206 (PuntPerCausePolicerDrops)
8     Te0/0/1                   internal0/0/rp:0          DROP   206 (PuntPerCausePolicerDrops)
9     Te0/0/1                   internal0/0/rp:0          DROP   206 (PuntPerCausePolicerDrops)
10    Te0/0/0                   internal0/0/rp:0          DROP   206 (PuntPerCausePolicerDrops)
11    Te0/0/1                   internal0/0/rp:0          DROP   206 (PuntPerCausePolicerDrops)
12    Te0/0/1                   internal0/0/rp:0          DROP   206 (PuntPerCausePolicerDrops)
…
```

Case 4. Device can’t install policy after reload (9)

While checking packets, noticed that some of them are originated from controllers:

```
DC1_101#show platform packet-trace packet 3
Packet: 3           CBUG ID: 3
Summary
  Input     : TenGigabitEthernet0/0/0
  Output    : internal0/0/rp:0
  State     : DROP 206 (PuntPerCausePolicerDrops)
  Timestamp
    Start   : 2699999601354 ns (03/16/2023 13:22:19.296832 UTC)
    Stop    : 2700000237397 ns (03/16/2023 13:22:19.297468 UTC)
Path Trace
  Feature: IPV4(Input)
    Input       : TenGigabitEthernet0/0/0
    Output      : <unknown>
    Source      : 10.50.1.26
    Destination : 10.60.1.6
    Protocol    : 17 (UDP)
      SrcPort   : 12346
      DstPort   : 12346
  Feature: SDWAN Implicit ACL
    Action : ALLOW
    Reason : SDWAN_TUN_CTRL
```

vSmart2

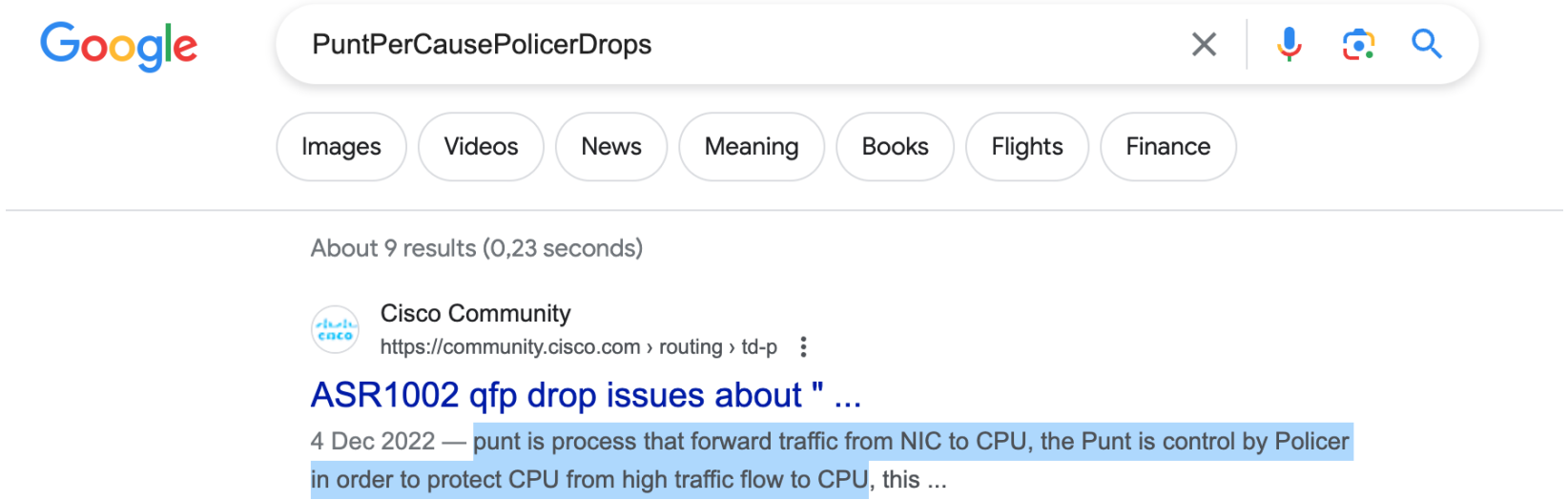
Case 4. Device can’t install policy after reload (10)

Facts and summary so far:

- No problems observed from vSmart perspective
- The hub router does not have policies installed but it should as per vSmart PoV
- The hub router dropping packets extensively with the drop code “PuntPerCausePolicerDrops”, some of the packets are from controllers
- We can guess by the name of the drop reason that there is some policer
- Reload of the device is a trigger

Case 4. Device can’t install policy after reload (11)

If you search for “PuntPerCausePolicerDrops” on the Internet, the very first result will help to explain the reason and find corresponding commands to check drop level settings



Conclusion: there is a rate limiter for punted (sent to CPU) control plane packets which is exceeded

Case 4. Device can’t install policy after reload (12)

… and you will find a command to confirm that there are a lot of packets dropped by this policer:

```
DC1_101#show platform software punt-policer drop-only

Per Punt-Cause Policer Configuration and Packet Counters


Punt                Config Rate(pps)     Conform Packets           Dropped Packets           Config Burst(pkts)  Config Alert
Cause  Description    Normal   High     Normal          High      Normal          High        Normal   High     Normal   High
-------------------------------------------------------------------------------------------------------------------------------
 11    For-us data    40000    5000     230             19482005  0               14789128    40000    5000     Off      Off

DC1_101#show platform software punt-policer drop-only

Per Punt-Cause Policer Configuration and Packet Counters


Punt                Config Rate(pps)     Conform Packets           Dropped Packets           Config Burst(pkts)  Config Alert
Cause  Description    Normal   High     Normal          High      Normal          High        Normal   High     Normal   High
-------------------------------------------------------------------------------------------------------------------------------
 11    For-us data    40000    5000     232             19607381  0               14883968    40000    5000     Off      Off
```

Why? Keep in mind there are ~4000 routers trying to establish tunnels at the same time after hub being reloaded and the hub has default settings for control plane policing

Case 4. Device can’t install policy after reload (13)

Solution is to increase the punt policer:

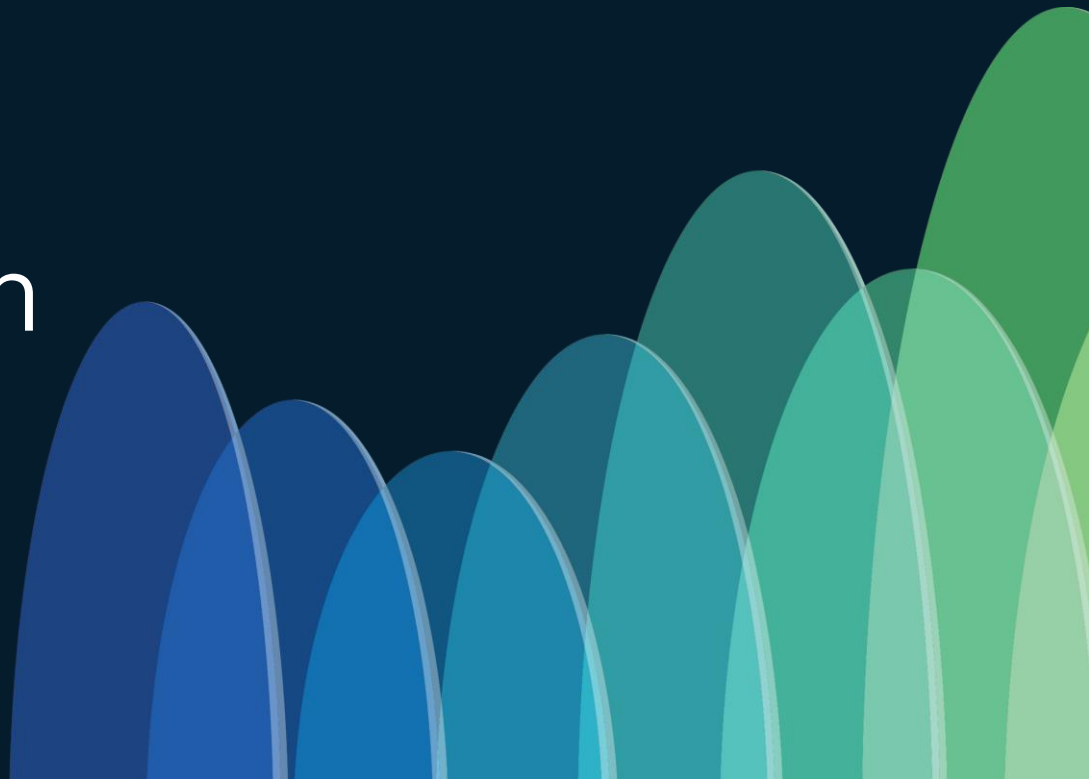
```
DC1_101#config-t

admin connected from 127.0.0.1 using console on Router
Router(config)# platform punt-policer 11 10000 high
Router(config)# commit
Commit complete
```

Test with a new reload confirms that policy installed sucessully after reconfiguration:

```
Mar 16 14:44:16.089: %Cisco-SDWAN-DC1_101-OMPD-6-INFO-400002: vSmart peer 1.1.1.26 state changed to Handshake
Mar 16 14:44:16.098: %Cisco-SDWAN-DC1_101-OMPD-5-NTCE-400002: vSmart peer 1.1.1.26 state changed to Up
Mar 16 14:44:16.098: %Cisco-SDWAN-DC1_101-OMPD-6-INFO-400005: Number of vSmarts connected : 2
Mar 16 14:44:20.260: %Cisco-SDWAN-DC1_101-OMPD-6-INFO-400007: Using policy from peer 1.1.1.20
```

```
DC1_101#sh sdwan omp summary | include policy

policy-sent             0
policy-received         4

DC1_101#sh sdwan bfd summary
sessions-total          8076
sessions-up             0
sessions-max            8076
sessions-flap           0
poll-interval           600000
```

Case 5. Traffic blackholing with DIA policy



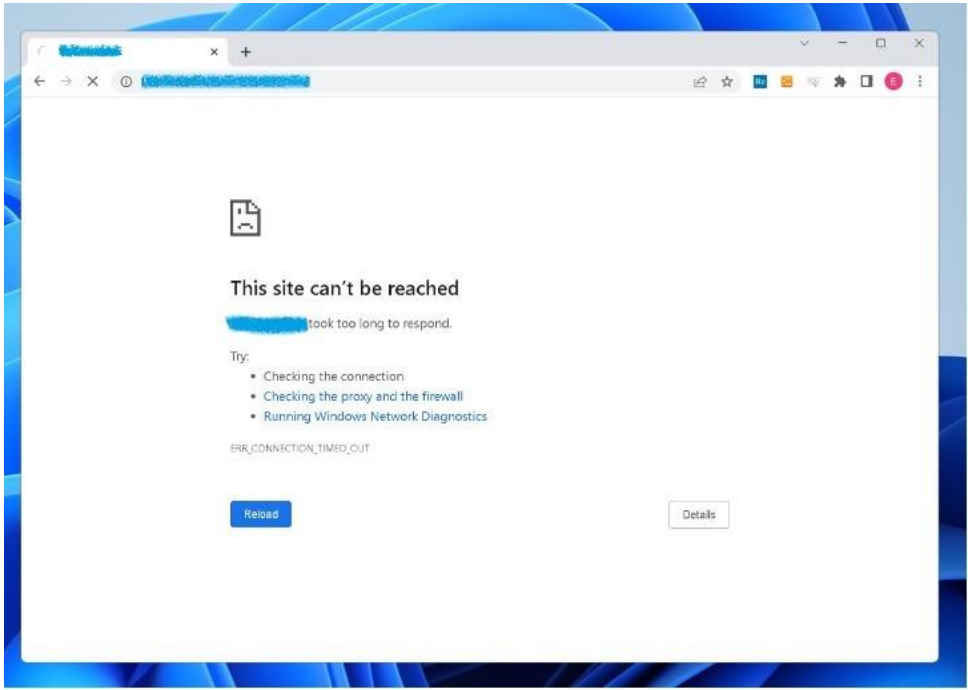
Case 5. Traffic blackholing with DIA policy (1)

Typical symptoms:

- User traffic affected, no connections to some enterprise internal servers
- Only limited set or a single application affected, but destination server is reachable with ”ping”, ports are opened
- Trigger is an implementation of Direct Internet Access (DIA) data policy or Cloud on Ramp (CoR) for SaaS (AAR policy).

Case 5. Traffic blackholing with DIA policy (2)

A common symptom reported by users: They cannot connect to internal servers (the connection times out):



Case 5. Traffic blackholing with DIA policy (3)

The data policy is very simple. Its purpose is to to implement DIA for Office 365, for example:

```
cE2_BR2#show sdwan policy from-vsmart
from-vsmart data-policy VPN_1_NAT
 direction from-service
 vpn-list VPN_1
  sequence 1
   match
    app-list O365
   action accept
    nat use-vpn 0
    no nat fallback
  default-action accept
from-vsmart lists vpn-list VPN_1
 vpn 1
from-vsmart lists app-list O365
 app ms-office-365
```

Case 5. Traffic blackholing with DIA policy (4)

As per the troubleshooting workflow, we need ot ensure correct next-hop/interface selection for the affected application traffic (https):

```
cE1#show sdwan policy service-path vpn 1 interface GigabitEthernet 4 source-ip 192.168.4.100 dest-ip 10.0.1.12 protocol 6 dest-port 443
Next Hop: Blackhole
```

Usually, this issue is due to a missing route to the destination, but that's not the case here.

```
cE1#show ip route vrf 1 10.0.1.12

Routing Table: 1
Routing entry for 10.0.0.0/16
  Known via "omp", distance 251, metric 0, type omp
  Last update from 169.254.206.35 on Sdwan-system-intf, 00:02:24 ago
  Routing Descriptor Blocks:
  * 169.254.206.35 (default), from 169.254.206.35, 00:02:24 ago, via Sdwan-system-intf
      opaque_ptr 0x7FB0E6FB62A0
      Route metric is 0, traffic share count is 1
```

And ”ping” works just fine, also confirmed by “**show sdwan policy service-path**” for ping app:

```
cE1#show sdwan policy service-path vpn 1 interface GigabitEthernet 4 source-ip 192.168.4.100 dest-ip 10.0.1.12 protocol 1 app ping
Next Hop: IPsec
  Source: 192.168.10.11 12366 Destination: 192.168.9.35 12346 Local Color: biz-internet Remote Color: biz-internet Remote System IP:
169.254.206.35
```

Case 5. Traffic blackholing with DIA policy (5)

If traffic blackholed, there is a reasonable assumption that it must be dropped on the device, right? Let's check QFP drop counters hence:

```
cE1#show platform hardware qfp active statistics drop clear
Last clearing of QFP drops statistics : Mon May  8 17:45:08 2023


-------------------------------------------------------------------------
Global Drop Stats                          Packets                Octets
-------------------------------------------------------------------------
BFDoffload                                     345                 29670
Disabled                                       247                 15414
Ipv4EgressIntfEnforce                            8                  1544
Ipv4NoAdj                                        6                   413
Ipv6NoRoute                                      5                   280
Nat64v6tov4                                      6                   480
SdwanDataPolicyDrop                            114                 15504
SdwanImplicitAclDrop                         11544               1984076
UnconfiguredIpv6Fia                            502                 54287

cE1#show platform hardware qfp active statistics drop
Last clearing of QFP drops statistics : Mon May  8 17:45:08 2023
 (58s ago)

-------------------------------------------------------------------------
Global Drop Stats                          Packets                Octets
-------------------------------------------------------------------------
<empty>
```

But there are only legitimate drops or no drops at all...

Case 5. Traffic blackholing with DIA policy (6)

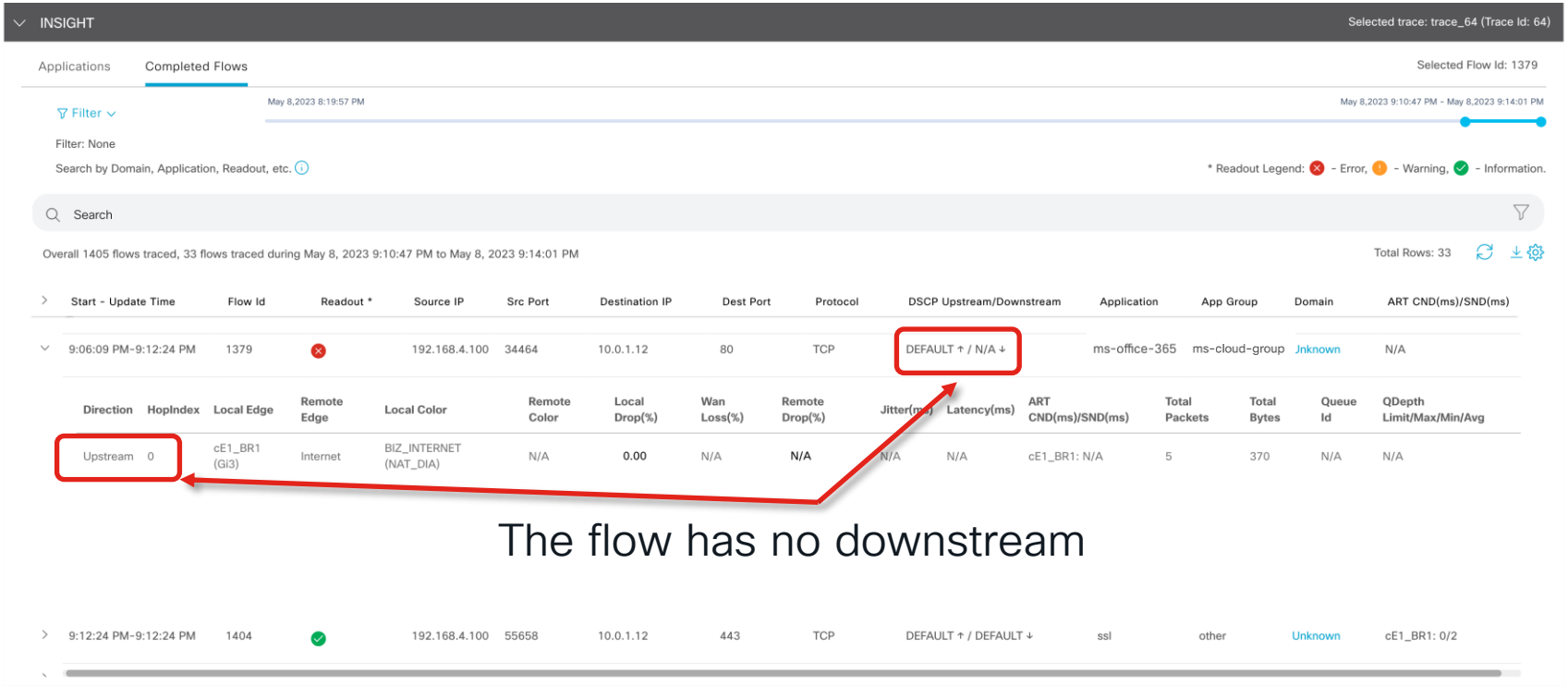
Packet trace to rescue again?

```
cE1#debug platform condition ipv4 10.0.1.12/32 both
cE1#debug platform packet-trace packet 1024 fia-trace
 Please remember to turn on 'debug platform condition start' for packet-trace to
work
cE1#debug platform condition start
cE1#show platform packet-trace summary
…
680   Tu2                       Gi4                       FWD
681   Gi4                       Gi2                       FWD
682   Gi4                       Gi2                       FWD
683   Tu2                       Gi4                       FWD
684   Tu2                       Gi4                       FWD
685   Gi4                       Gi2                       FWD
686   Gi4                       Gi2                       FWD
687   Tu2                       Gi4                       FWD
688   Tu2                       Gi4                       FWD
689   Gi4                       Gi2                       FWD
690   Gi4                       Gi2                       FWD
691   Tu2                       Gi4                       FWD
692   Gi4                       Gi2                       FWD
693   Gi4                       Gi2                       FWD
694   Tu2                       Gi4                       FWD
695   Tu2                       Gi4                       FWD
696   Gi4                       Gi2                       FWD
697   Gi4                       Gi3                       FWD
698   Gi4                       Gi3                       FWD
699   Gi4                       Gi3                       FWD
700   Gi4                       Gi3                       FWD
701   Gi4                       Gi3                       FWD
702   Gi4                       Gi3                       FWD
```

Unlikely, unless you know the exact source and destination because multiple various parallel flows are established usually.

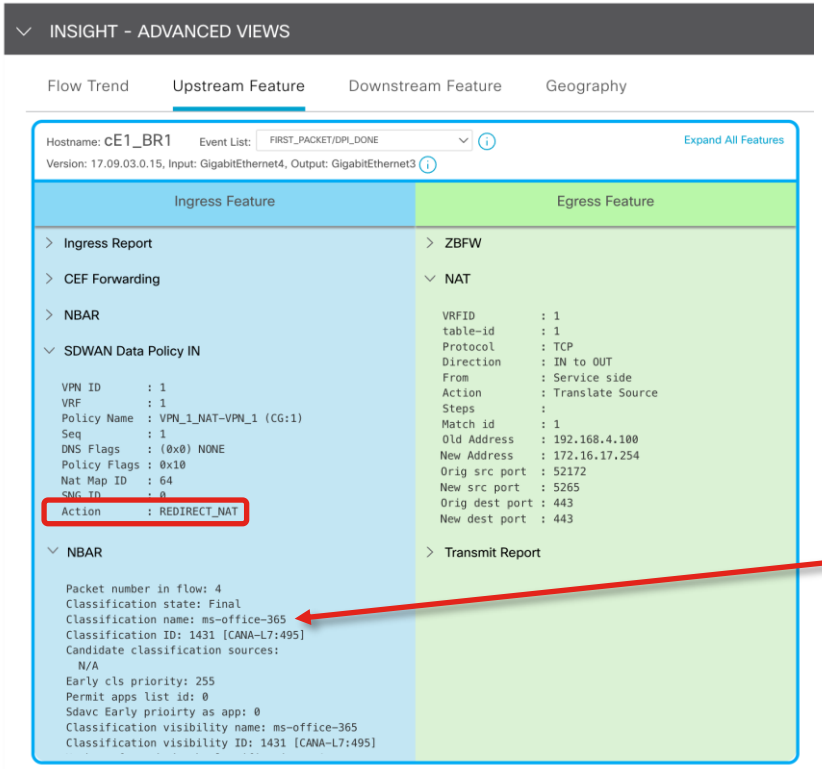
Case 5. Traffic blackholing with DIA policy (7)

In a live network packet-trace may cause a lot of confusion if you can't define exact flow filter, so NWPI is preferred because it will trace all flows end-to-end and show all-in-one insignt:



Case 5. Traffic blackholing with DIA policy (8)

From ”insight - advanced view”, you can find that DIA data policy was applied:



In the “Insights – Advanced view” you can see DPI (NBAR) misclassified internal application as “ms-office-365” and traffic was sent to DIA circuit instead of overlay tunnel, hence, causing traffic blackholing

Case 5. Traffic blackholing with DIA policy (9)

Why does misclassification happen?

Great question and the answer is that it depends… Sometimes apps are just hard to recognize and differentiate (on-prem services vs SaaS) or it may be a bug or outdated NBAR DPI package.

The good news is that most of the time solution for the case with DIA is very simple:

- Ensure RFC1918 prefixes excluded from DPI evaluation
- Inherited benefit: reduced load on a router because less complex match condition for RFC1918 addresses VS app based match

How?

- Insert data policy sequence above the sequence for NAT and match RFC1918 addresses, then just accept matching packets (accept is a final action)

```
policy
 data-policy VPN_1_NAT
  vpn-list VPN_1
   !
   sequence 1
    match
     destination-data-prefix-list RFC1918
    !
    action accept
    !
   !
<the rest of the policy sequences goes there>
```

Case 5. Traffic blackholing with DIA policy (10)

Keep in mind, same problem may be experienced with with CoR for SaaS:

```
Router#show sdwan policy from-vsmart
from-vsmart app-route-policy _CC1_AAR_POLICY
vpn-list CC1
sequence 41
   match
    source-ip           0.0.0.0/0
    cloud-saas-app-list office365_apps
   action
    count office365_apps_-856788698
    cloud-saas
  sequence 51
   match
    source-ip           0.0.0.0/0
    cloud-saas-app-list salesforce_apps
   action
    count salesforce_apps_-856788698
    cloud-saas
  default-action sla-class DEFAULT
```

Solution is similar, configure sequence to exclude RFC1918 in AAR CoR for SaaS policy or use data-policy which sets remote TLOC (**`set tloc`** or **`set tloc-list`**) to override AAR policy.

Why so complicated? I thought the same. That's why enhancement CSCvv68740 was implemented in 20.13/17.13 to exclude RFC1918 by default

Side note: applications classification issues

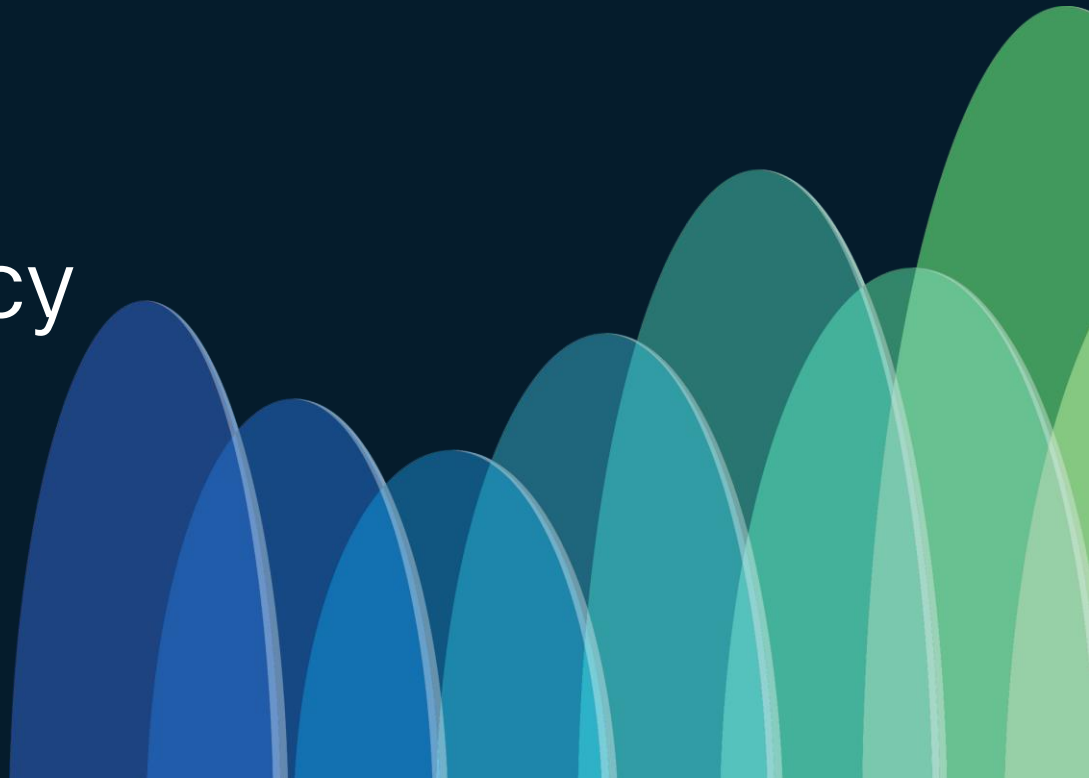
Why can not DPI (NBAR) recognize application properly or fails to recognize it based on a first packet (First Packet Match or FPM)?

Overwhelming majority of apps recognized based on preceding DNS request/reply, hence the top reason is:

- request and/or reply is not seen on the router performing match based on NBAR classification because:

a. DNS encrypted, e.g. DNSoverHTTPS (DoH), DNSoverTLS (DoT) and so on
b. Asymmetric routing in dual-homed sites
c. DNS traffic is not passing via the router (e.g. follows another path outside SD-WAN overlay)
d. DNS traffic forwarded within VRF which is different from VRF where application data is sent
e. So called DNS-pipelining (multiple DNS request sent over the same UDP stream within short period of time, typically on a heavily loaded system). IOS-XE 17.12+ can handle up to 32 consecutive requests in the same stream, but older versions recognize first request only

Case 6. DSCP marking not applied with policy



Case 6. Incorrect DSCP marking symptoms

- **`show sdwan app-fwd cflowd flows`** shows DSCP mark as “0”

```
cE1_BR1#show sdwan app-fwd cflowd flows vpn 4
Generating output, this might take time, please wait ...
app-fwd cflowd flows vpn 4 src-ip 192.168.5.197 dest-ip
192.168.4.196 src-port 22 dest-port 37748 dscp 4 ip-proto 6
 tcp-cntrl-bits           24
 icmp-opcode              0
 total-pkts               6
 total-bytes              2064
 start-time               "Fri Dec 22 15:35:11 2023"
 egress-intf-name         GigabitEthernet4
 ingress-intf-name        GigabitEthernet3
 application              ssh
 family                   terminal
 drop-cause               "No Drop"
 drop-octets              0
 drop-packets             0
 sla-not-met              0
 color-not-met            0
 queue-id                 2
 initiator                2
 tos                      0
 dscp-output              0
 sampler-id               0
 fec-d-pkts               0
 fec-r-pkts               0
 pkt-dup-d-pkts-orig      0
 pkt-dup-d-pkts-dup       0
 pkt-dup-r-pkts           0
 pkt-cxp-d-pkts           0
 category                 0
 service-area             0
 cxp-path-type            0
 region-id                0
 ssl-read-bytes           0
 ssl-written-bytes        0
 ssl-en-read-bytes        0
 ssl-en-written-bytes     0
 ssl-de-read-bytes        0
 ssl-de-written-bytes     0
 ssl-service-type         0
 ssl-traffic-type         0
 ssl-policy-action        0
 appqoe-action            0
 appqoe-sn-ip             0.0.0.0
 appqoe-pass-reason       0
 appqoe-dre-input-bytes   0
 appqoe-dre-input-packets 0
 appqoe-flags             0
```

Case 6. Incorrect DSCP marking symptoms (2)

- But data policy configured to mark it as “10” (AF11). Policy received on the device:

```
cE1_BR1#show sdwan policy from-vsmart data-policy
from-vsmart data-policy SET_DSCP
 direction from-service
 vpn-list VPN_4
  sequence 1
   match
    destination-port 22
    protocol         6
   action accept
    cflowd
    set
     dscp 10
  sequence 2
   match
    source-port 22
    protocol    6
   action accept
    cflowd
    set
     dscp 10
  default-action accept
```

Case 6. Incorrect DSCP marking symptoms (3)

Here is cflowd template for reference:

```
vsmart1# show running-config policy cflowd-template
policy
 cflowd-template test-cflowd-template
  template-refresh 90
  collector vpn 0 address 192.168.10.240 port 9555
transport transport_udp
  !
 !
!
```

Case 6. Incorrect DSCP marking symptoms (4)

- Packet trace performed to confirm if DSCP 10 was set, and it was:

```
cE1_BR1#show platform packet-trace packet 1
Packet: 1           CBUG ID: 138
Summary
  Input     : GigabitEthernet4
  Output    : GigabitEthernet3
  State     : FWD
  Timestamp
    Start   : 15111231553111 ns (12/22/2023 15:25:22.147214 UTC)
    Stop    : 15111231650980 ns (12/22/2023 15:25:22.147311 UTC)
Path Trace
  Feature: IPV4(Input)
    Input       : GigabitEthernet4
    Output      : <unknown>
    Source      : 192.168.4.196
    Destination : 192.168.5.197
    Protocol    : 6 (TCP)
      SrcPort   : 22
      DstPort   : 44408

  <skipped>
```

```
Feature: IPV4_INPUT_FNF_FIRST
    Entry       : Input - 0x814db670
    Input       : GigabitEthernet4
    Output      : <unknown>
    Lapsed time : 1682 ns
  Feature: DEBUG_COND_APPLICATION_IN_CLR_TXT
    Entry       : Input - 0x814ca90c
    Input       : GigabitEthernet4
    Output      : <unknown>
    Lapsed time : 32 ns
  Feature: SDWAN Data Policy IN
    VPN ID      : 4
    VRF         : 2
    Policy Name : SET_DSCP-VPN_4 (CG:4)
    Seq         : 2
    DNS Flags   : (0x0) NONE
    Policy Flags : 0x408
    Policy Flags2: 0x0
    Action      : FNF
    Action      : SET_DSCP af11(10)
  Feature: SDWAN_POLICY_FIA

<rest is skipped>
```

Case 6. Incorrect DSCP marking symptoms (5)

- Which output should we trust?
- Packet capture (tcpdump) on the remote host ultimately confirmed that DSCP set properly by the router:

```
root@user:/home/user# tcpdump -v "host 192.168.4.196" -i ens192
tcpdump: listening on ens192, link-type EN10MB (Ethernet), capture size 262144 bytes
17:01:45.554798 IP (tos 0x28, ttl 62, id 55357, offset 0, flags [DF], proto TCP (6), length 88)
    192.168.4.196.37748 > 192.168.5.197.ssh: Flags [P.], cksum 0x8f47 (correct), seq 290589212:290589248, ack 2393058474, win 501,
options [nop,nop,TS val 3705085476 ecr 4291921734], length 36
17:01:45.555261 IP (tos 0x10, ttl 64, id 25646, offset 0, flags [DF], proto TCP (6), length 152)
    192.168.5.197.ssh > 192.168.4.196.37748: Flags [P.], cksum 0x8c64 (incorrect -> 0x3cdd), seq 1:101, ack 36, win 501, options
[nop,nop,TS val 4291946734 ecr 3705085476], length 100
17:01:45.555967 IP (tos 0x28, ttl 62, id 55358, offset 0, flags [DF], proto TCP (6), length 52)
    192.168.4.196.37748 > 192.168.5.197.ssh: Flags [.], cksum 0x9a62 (correct), ack 101, win 501, options [nop,nop,TS val
3705085477 ecr 4291946734], length 0
17:01:45.710499 IP (tos 0x28, ttl 62, id 55359, offset 0, flags [DF], proto TCP (6), length 88)
    192.168.4.196.37748 > 192.168.5.197.ssh: Flags [P.], cksum 0xe590 (correct), seq 36:72, ack 101, win 501, options [nop,nop,TS
val 3705085632 ecr 4291946734], length 36
<rest is skipped>
```

* 0x28 ToS HEX = 10 DSCP decimal = AF11

Case 6. Incorrect DSCP marking symptoms (6)

- Let's check packet-trace again and analyse it:

```
cE1_BR1#show platform packet-trace packet 1
Packet: 1           CBUG ID: 138
Summary
  Input     : GigabitEthernet4
  Output    : GigabitEthernet3
  State     : FWD
  Timestamp
    Start   : 15111231553111 ns (12/22/2023 15:25:22.147214 UTC)
    Stop    : 15111231650980 ns (12/22/2023 15:25:22.147311 UTC)
Path Trace
  Feature: IPV4(Input)
    Input       : GigabitEthernet4
    Output      : <unknown>
    Source      : 192.168.4.196
    Destination : 192.168.5.197
    Protocol    : 6 (TCP)
      SrcPort   : 22
      DstPort   : 44408

  <skipped>
```

```
Feature: IPV4_INPUT_FNF_FIRST
  Entry       : Input - 0x814db670
  Input       : GigabitEthernet4
  Output      : <unknown>
  Lapsed time : 1682 ns
Feature: DEBUG_COND_APPLICATION_IN_CLR_TXT
  Entry       : Input - 0x814ca90c
  Input       : GigabitEthernet4
  Output      : <unknown>
  Lapsed time : 32 ns
Feature: SDWAN Data Policy IN
  VPN ID      : 4
  VRF         : 2
  Policy Name : SET_DSCP-VPN_4 (CG:4)
  Seq         : 2
  DNS Flags   : (0x0) NONE
  Policy Flags : 0x408
  Policy Flags2: 0x0
  Action      : FNF
  Action      : SET_DSCP af11(10)
Feature: SDWAN_POLICY_FIA

<rest is skipped>
```

Note that data policy action "cflowd" called here FNF (Flexible Net Flow), but FNF feature involed before the data policy feature in FIA

Case 6. Incorrect DSCP marking - solution

- Is it an order of operations issue?
- Yes, kind of, but there is an option available to ensure DSCP/ToS marking recorded into NetFlow data anyway.
- I did not show cflowd template view as per the router perspective intentionally because then problem and solution would be obvious (if you attentive enough):

```
cE1_BR1#show sdwan policy from-vsmart cflowd-template
from-vsmart cflowd-template test-cflowd-template
 flow-active-timeout    600
 flow-inactive-timeout  60
 template-refresh       90
 flow-sampling-interval 1
 protocol               ipv4
 no collect-tloc-loopback
 customized-ipv4-record-fields
  no collect-tos
  no collect-dscp-output
 collector vpn 0 address 192.168.10.240 port 9555 transport transport_udp
```

Case 6. Incorrect DSCP marking – solution (2)

- Let's fix it and reconfigure vSmart cflowd template (this feature was introduced in 20.6+ to address such requirement)

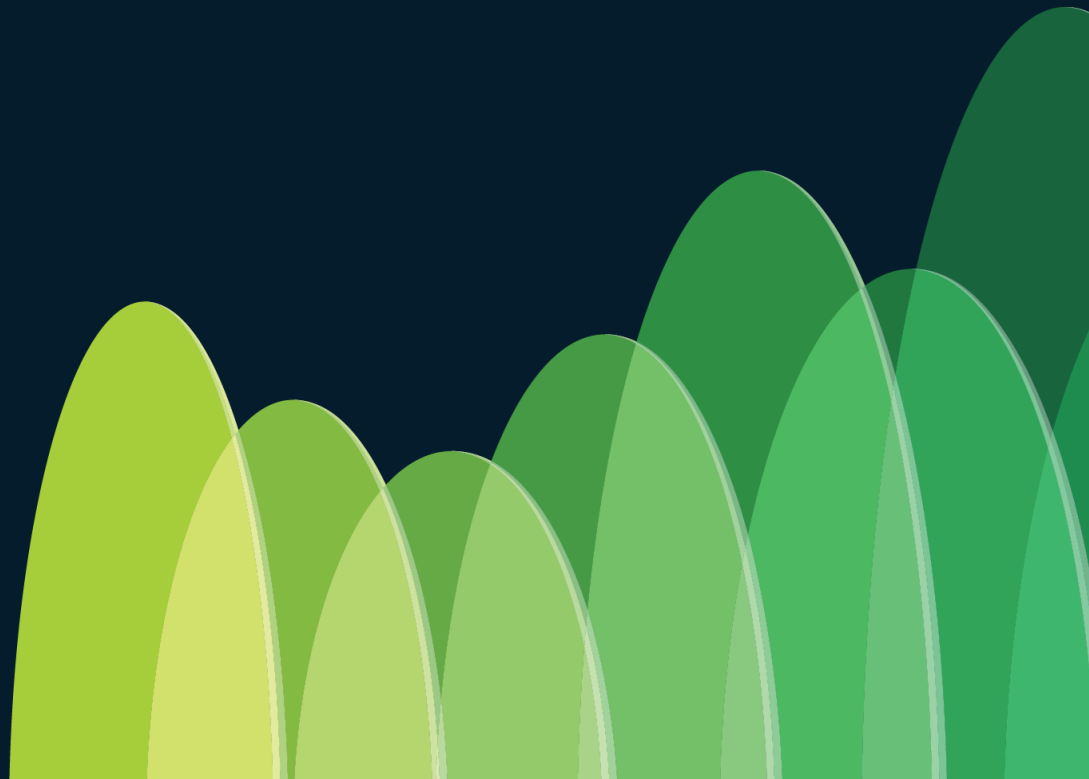
```
vsmart1(config)# show configuration
policy
 cflowd-template test-cflowd-template
  customized-ipv4-record-fields
   collect-tos
   collect-dscp-output
  !
 !
!
vsmart1(config)# commit
```

Case 6. Incorrect DSCP marking – solution (3)

… and then let's check the same output again:

```
cE1_BR1#show sdwan app-fwd cflowd flows vpn 4
Generating output, this might take time, please wait ...
app-fwd cflowd flows vpn 4 src-ip 192.168.4.196 dest-ip
192.168.5.197 src-port 33418 dest-port 22 dscp 4 ip-proto 6
 tcp-cntrl-bits           24
 icmp-opcode              0
 total-pkts               26
 total-bytes              1568
 start-time               "Fri Dec 22 16:28:57 2023"
 egress-intf-name         GigabitEthernet3
 ingress-intf-name        GigabitEthernet4
 application              ssh
 family                   terminal
 drop-cause               "No Drop"
 drop-octets              0
 drop-packets             0
 sla-not-met              0
 color-not-met            0
 queue-id                 2
 initiator                1
 tos                      16
 dscp-output              10
 sampler-id               0
 fec-d-pkts               0
 fec-r-pkts               0
 pkt-dup-d-pkts-orig      0
 pkt-dup-d-pkts-dup       0
 pkt-dup-r-pkts           0
 pkt-cxp-d-pkts           0
 category                 0
 service-area             0
 cxp-path-type            0
 region-id                0
 ssl-read-bytes           0
 ssl-written-bytes        0
 ssl-en-read-bytes        0
 ssl-en-written-bytes     0
 ssl-de-read-bytes        0
 ssl-de-written-bytes     0
 ssl-service-type         0
 ssl-traffic-type         0
 ssl-policy-action        0
 appqoe-action            0
 appqoe-sn-ip             0.0.0.0
 appqoe-pass-reason       0
 appqoe-dre-input-bytes   0
 appqoe-dre-input-packets 0
 appqoe-flags             0
```

As a conclusion: Typical policy faults



Typical policy issues (generic)

- `**default-action reject**` or `**default-action accept**`
- wrong direction of policy application (in vs out, from-tunnel vs from-service)
- policy application scope is too narrow or too wide (i.e. site-id not specified in a sequence match statement and action is applied to the whole set of site-list defined under **apply-policy** section)
- simple misconfigurations and typos (e.g. a prefix missing from a prefix-list, wrong mask, wrong site-id and so on).
- Always keep in mind policy processing logic: once matched by a policy sequence, the subject to policy application is final, further sequences are not processed for the same subject

Typical Control Policy specific issues

- Control policy applied on inbound direction before OMP best-path selection resulting in backup paths missing
- Unconditional TLOC rewrites (e.g. “**`set tloc-list`**” used. vSmart does not care of TLOCs state)
- Attempt to use “**`set tloc-action`**” while “**`service TE`**” is not enabled on WAN Edge
- Attempt to glue/stick together different colors with “**`set tloc-action`**”

Typical AAR and Data policies specific issues

- Common AAR issues:
  - return traffic is asymmetric. It does not mean that AAR malfunctions (because feature is unidirectional, traffic may return on a different color if a remote device has no policy to ensure symmetry)
  - equal cost paths (ECMP) over multiple colours are not available, hence AAR policy has no choice.
- Common AAR misunderstanding:
  - by default, it may take up to 1 hour for AAR policy to change a path (**`app-route poll-interval`** 600s x **`multiplier`** 6 = 1h)
  - **`bfd poll-interval`** impacts frequency of **`app-route poll-interval`** updates (accuracy), but not AAR reaction time (convergence) as such
- Common issues AAR+Data policy in use: in short, DP overrides AAR, but considers AAR SLA class match (20.6+ behaviour)
- Common traffic classification issue: DPI (NBAR) can't match an application because DNS packets are not seen
- Fallback issues: DIA **`nat fallback`** or SIG **`sig-action fallback-to-routing`** not configured by default.
- Policy bypass happens because first packet match fails (Policy-Bypass-FPM-Fail): may need **`policy flow-stickiness-disable`** (17.6+ feature), but it may cause TCP connection resets
- Fragmented packets match to a "wrong" sequence (i.e. UDP fragments considered matching to a sequence with port match condition despite that there is no UDP port info available in IP fragment)

AAR

For Your Reference

Below doc provides detailed explanation about AAR Policy, its config and how SLA is measured with the help of BFD interval and Multiplier https://www.cisco.com/c/en/us/td/docs/routers/sdwan/configuration/policies/ios-xe-17/policies-book-xe/application-aware-routing.html

* For each sliding window time period, application-aware routing sees 600 BFD Hello packets (polling interval / BFD Hello interval: 600 seconds / 1 = 600 hello packets per interval). These packets provide measurements of packet loss and latency on the data plane tunnels.
* Application-aware routing retains the statistics for 1 hour (polling interval x multiplier: 10 minutes x 6 = 60 minutes).
* The statistics are placed in six separate buckets, indexed with the numbers 0 through 5. Bucket 0 contains the latest statistics, and bucket 5 the oldest. Every 10 minutes, the newest statistics are placed in bucket 0, the statistics in bucket 5 are discarded, and the remaining statistics move into the next bucket.
* Every 60 minutes (every hour), application-aware routing acts on the loss and latency statistics. It calculates the mean of the loss and latency of all the buckets in all the sliding windows and compares this value to the specified SLAs for the tunnel. If the calculated value satisfies the SLA, application-aware routing does nothing. If the value does not satisfy the SLA, application-aware routing calculates a new tunnel.
* Application-aware routing uses the values in all six buckets to calculate the mean loss and latency for a data tunnel. This is because the multiplier is 6. While application-aware always retains six buckets of data, the multiplier determines how many it actually uses to calculate the loss and latency. For example, if the multiplier is 3, buckets 0, 1, and 2 are used.

References and recommended resources

- Cisco Troubleshooting Tech Notes: https://www.cisco.com/c/en/us/support/routers/sd-wan/products-tech-notes-list.html
- BRKENT-3793/BRKTRS-3793 “Advanced SD-WAN Routing Troubleshooting”
- BRKTRS-3475 “Advanced Troubleshooting of CAT8k, ASR1k, ISR and SD-WAN Edge made easy”
- BRKRST-2791 “Building and Using Policies with Cisco SD-WAN”
- BRKENT-2477 “Cisco SD-WAN Troubleshooting”

Webex App

Questions?

Use the Webex app to chat with the speaker after the session

How

1. Find this session in the Cisco Events mobile app
2. Click “Join the Discussion”
3. Install the Webex app or go directly to the Webex space
4. Enter messages/questions in the Webex space

Webex spaces will be moderated by the speaker until February 28, 2025.

Fill Out Your Session Surveys

Participants who fill out a minimum of 4 session surveys and the overall event survey will get a unique Cisco Live t-shirt.

(from 11:30 on Thursday, while supplies last)

All surveys can be taken in the Cisco Events mobile app or by logging in to the Session Catalog and clicking the ‘Participant Dashboard’

Content Catalog

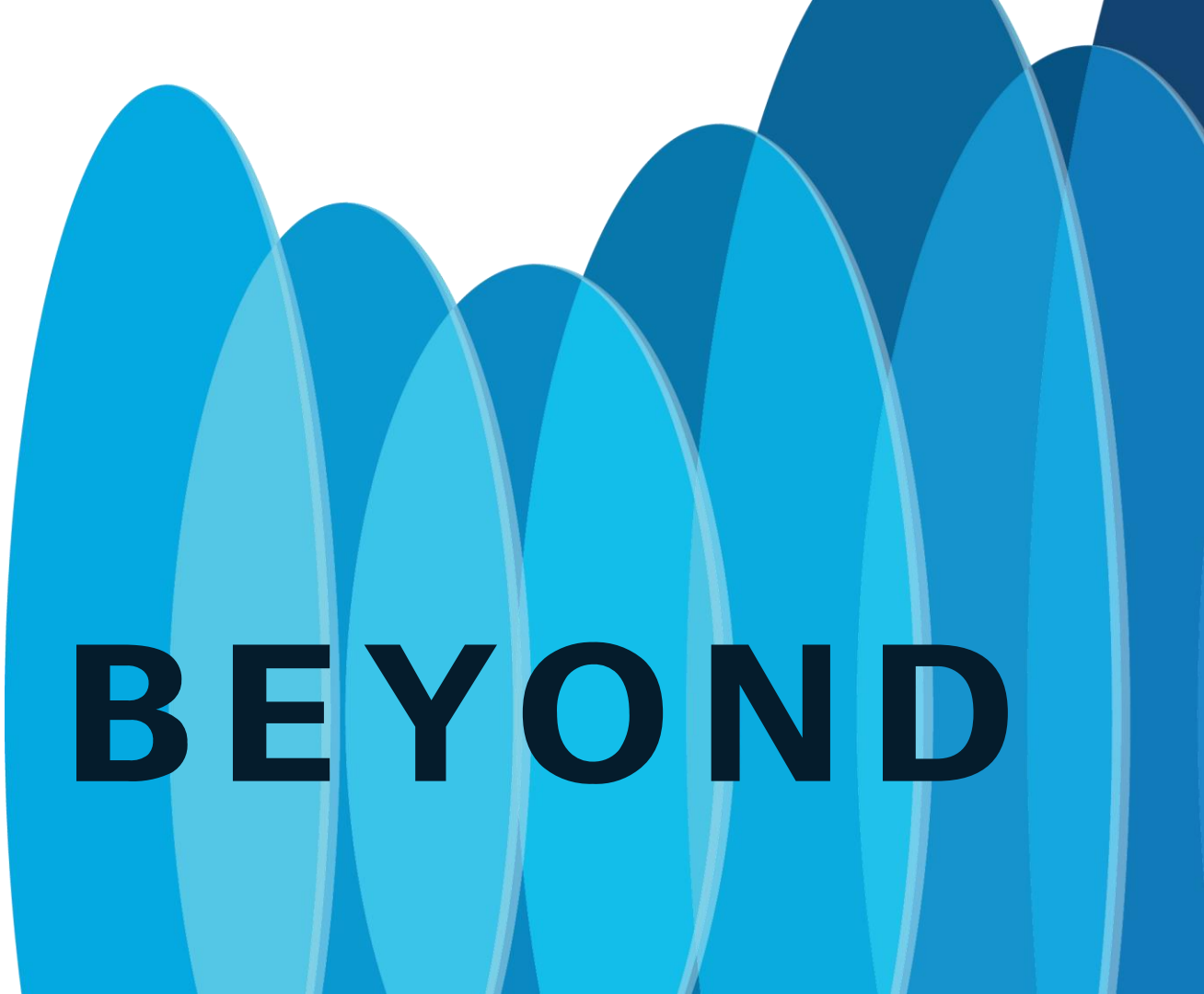
Continue your education

- Visit the On-Demand Library for more sessions at ciscolive.com/on-demand. Sessions from this event will be available from March 3.

Contact me at: enk@cisco.com


cisco
Thank you
cisco Live!

cisco Live!



GO BEYOND