Segment Routing - On Demand Next Hop

This document gives an example of how On Demand Next Hop works based on the below topology. Traffic flow will be examined from CE1 to CE2 based on policies set by the ingress PE (PE2). No custom SR Flexible-Algorithms are used.

What is On Demand Next Hop?
An SR Policy can be instantiated (meaning put into the forwarding table) in one of two ways:

> *Explicity:* using a full policy configuration or a PCE controller
> *On Demand Next-Hop*: using a template for a given **color**.

Unlike explicitly configuring an SR TE policy, ODN does not need to specify an endpoint. It will also add/remove the SR Policy dynamically based on the presence/absence of a corresponding service route (e.g. a route with the correct **color**)

How this relates to Automatic Steering
ODN will dynamically create the SR Policy. Automatic Steering (AS) steers the matching traffic down that policy (meaning it instantiates the necessary forwarding instances and imposes the necessary labels). In this way, ODN and AS work together.

When to use ODN?
The idea behind ODN is that the engineer can focus on the policy and desired SLA without worrying about the transport. The destination is not fixed. This means that the headend might not even be aware of the full topology towards the destination - including all the link and node characteristics. If this is the case, a PCE can be used to fill in those gaps. This leaves the headend free to configure the optimisations and restrictions that adhere to the desired policy without worrying about the exact hop-by-hop or even needing to signal it.

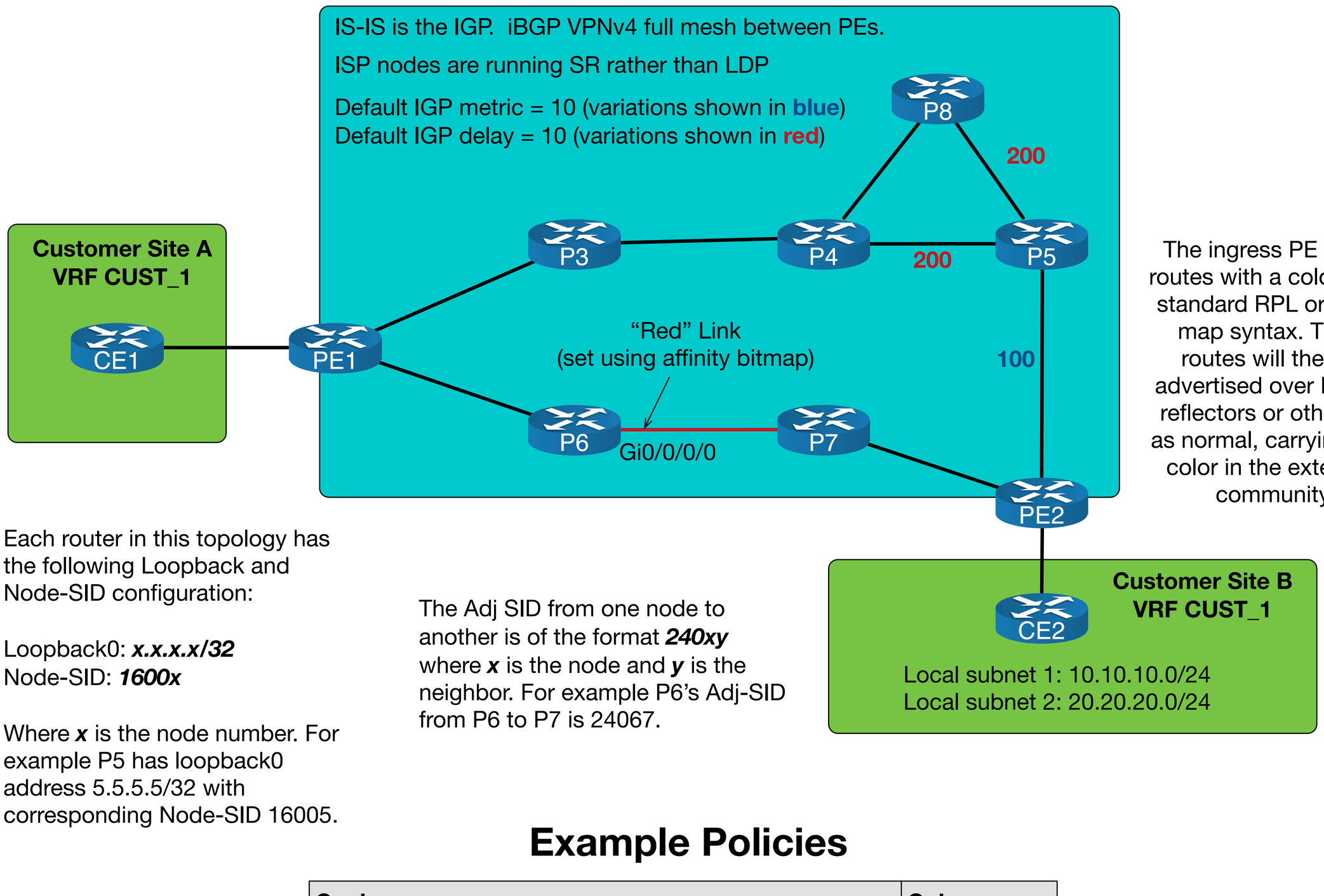
Instantiation Process

Assume ODN template for Color **C** is configured at headend router **H**.

Step	Operation
1	**H** receives a BGP prefix with Color community **C** and next-hop **E**.
2	If an SR Policy (**C**, **E**) doesn't already exist, one will be created.
3	Two candidate paths are instantiated: **>** One with preference 200 matching what the head-end node **H** computes. **>** Another with preference 100 from a PCE. If no PCE is configured this is marked as invalid.
4	The best candidate path (highest preference) is selected.
5	A corresponding SID List and BSID are put into the forwarding plane.
6	AS begins to steer traffic down this policy.

If a BGP withdrawal is received for the last remaining BGP prefix with Color **C**, the SR Policy is removed.

Network Topology



The ingress PE marks routes with a color using standard RPL or route-map syntax. These routes will then be advertised over BGP to reflectors or other PEs as normal, carrying their color in the extended community.

Each router in this topology has the following Loopback and Node-SID configuration:

Loopback0: ***x.x.x.x/32***
Node-SID: ***1600x***

Where ***x*** is the node number. For example P5 has loopback0 address 5.5.5.5/32 with corresponding Node-SID 16005.

The Adj SID from one node to another is of the format ***240xy*** where ***x*** is the node and ***y*** is the neighbor. For example P6's Adj-SID from P6 to P7 is 24067.

Example Policies

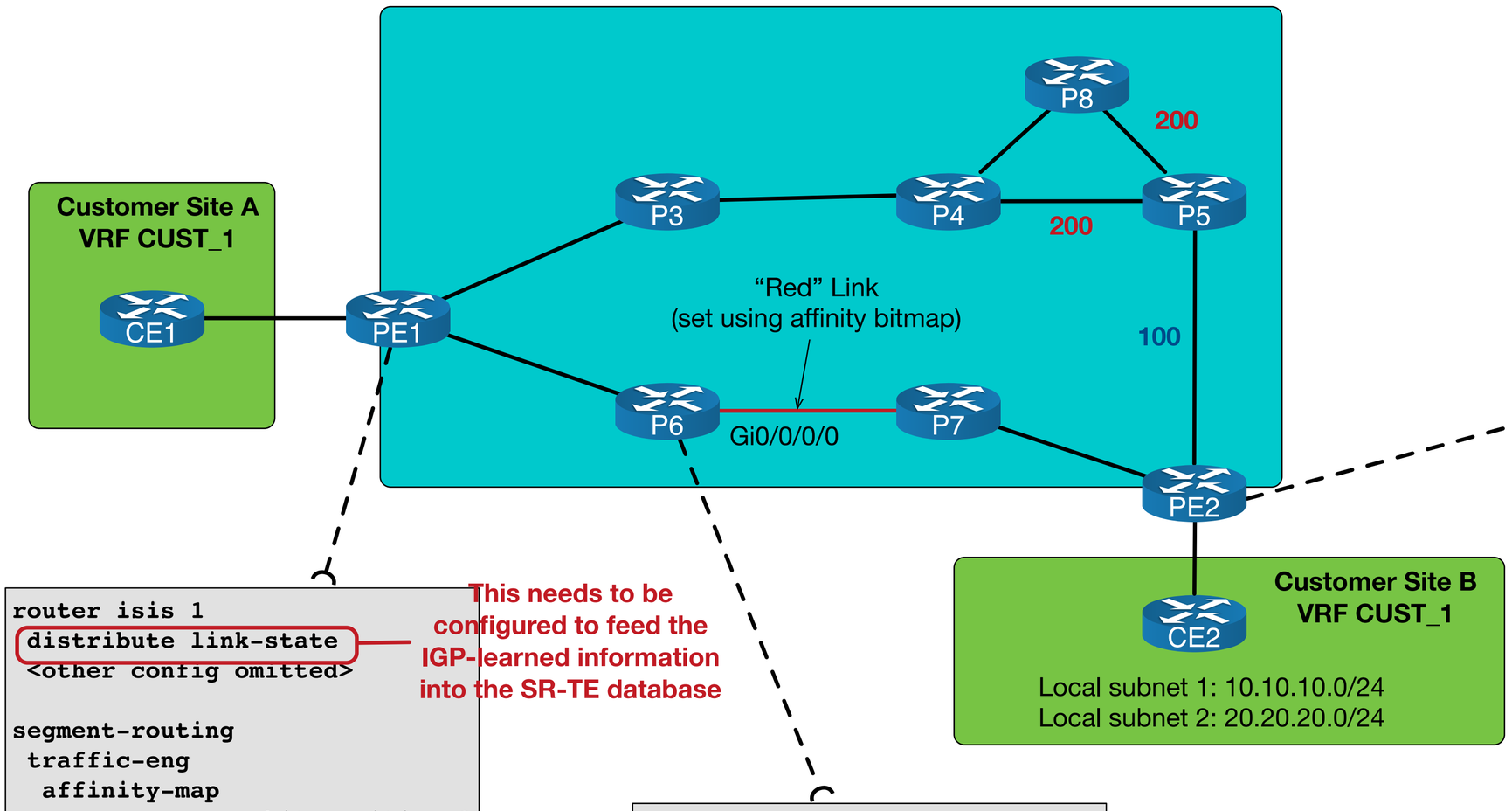
Goal	Color
Traffic from Customer Site A to 10.10.10.10 should follow path with smallest IGP metric and avoid red links	**BLUE (20)**
Traffic from Customer Site A to 20.20.20.20 should follow path with smallest IGP delay	**GREEN (30)**

by Steven Crutchley

Segment Routing - On Demand Next Hop

Configuration

This page shows IOS-XR CLI configuration to setup the policies shown on the previous page. PE1 and PE2 have a loopback to loopback VPNv4 iBGP session between each other. The PE to CE routing protocol used is eBGP.




```
router isis 1
 distribute link-state
 <other config omitted>

segment-routing
 traffic-eng
  affinity-map
   name RED_LINK bit-position 0
  on-demand color 20
   dynamic
    metric
     type igp
    affinity exclude-any
      name RED_LINK
  on-demand color 30
   dynamic
    metric
     type delay
```

This needs to be configured to feed the IGP-learned information into the SR-TE database

```
segment-routing
 traffic-eng
  affinity-map
   name RED_LINK bit-position 0
  !
  interface Gi0/0/0/0
   affinity name RED_LINK
```

Refers to the position in the affinity bitmap that RED_LINK represents

This name is locally significant

Unlike an explicit configuration, an end-point is not specified. It is provided by the BGP route that is received with the matching color community

```
extcommunity-set opaque BLUE
 20
end-set
!
extcommunity-set opaque GREEN
 30
end-set
!
route-policy COLOR-INBOUND-PREFIXES
 if destination in (10.10.10.0/24) then
  set extcommunity color BLUE
 endif
 if destination in (20.20.20.0/24) then
  set extcommunity color GREEN
 endif
  pass
end-policy
!
router bgp 1
 bgp router-id 2.2.2.2
 address-family ipv4 unicast
 address-family vpnv4 unicast
 !
 neighbor 1.1.1.1
  description to PE1
  remote-as 1
  update-source Loopback0
  address-family ipv4 unicast
  address-family vpnv4 unicast
 !
 vrf CUST_1
  rd auto
  address-family ipv4 unicast
  !
  neighbor 172.16.1.2
   remote-as 2
   description to CE2
   address-family ipv4 unicast
    route-policy COLOR-INBOUND-PREFIXES in
```

Restricting ODN Next-hops

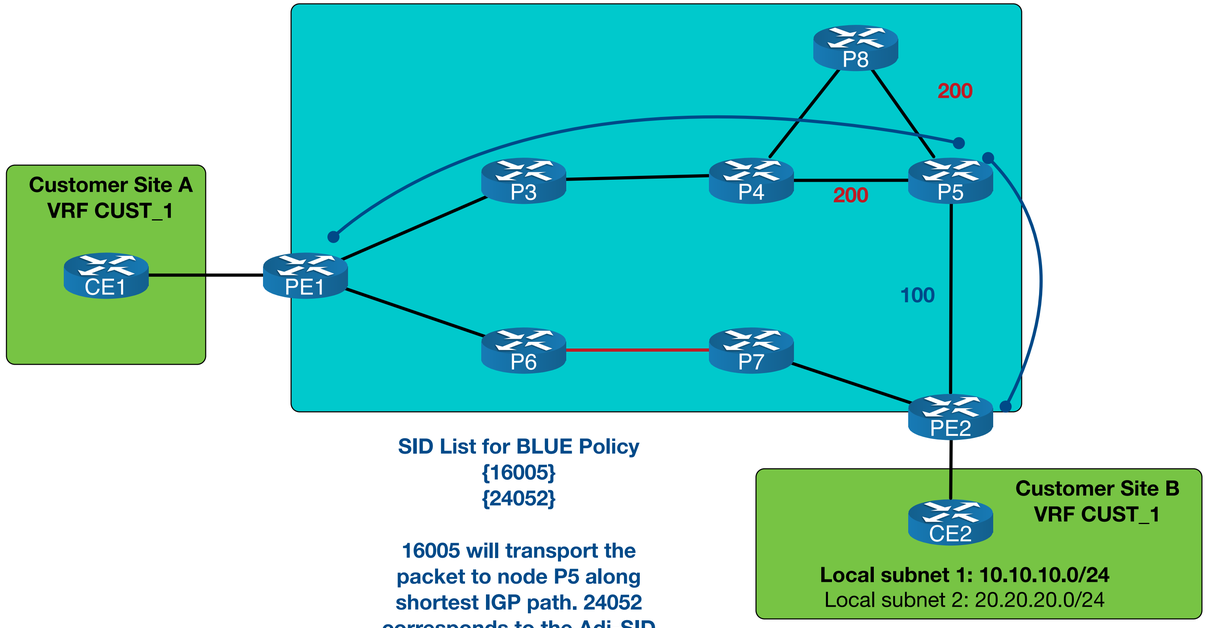
A color is authorised when an on-demand template for it is configured. If needed, a filter can be setup that only performs automatic instantiation for the next-hops that pass the filter. For example:

```
ipv4 prefix-list ODN_RESTRICT
  10 permit 2.2.2.0/24
!
segment-routing
  traffic-eng
    on-demand color 30
      restrict ODN_RESTRICT
      dynamic
        metric
          type delay
```

Segment Routing - On Demand Next Hop

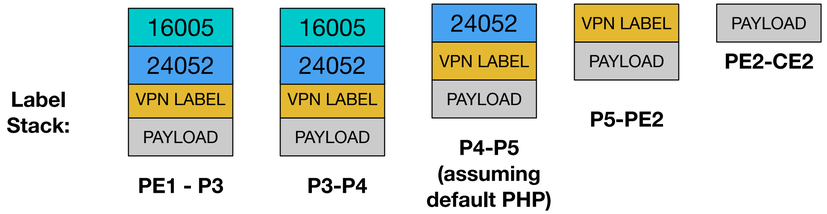
This page examines the SID lists that are created as a results of the 2 policies covered in this document. Assume PE1 has received the BGP routes for subnet1 and subnet2 with color communities for BLUE (20) and GREEN (30) respectively (see previous page for config). Each SID in the resulting SID list represents a segment used to adhere to the desired policy.

BLUE Policy Traffic Flow
(Smallest IGP Metric Avoiding red links)

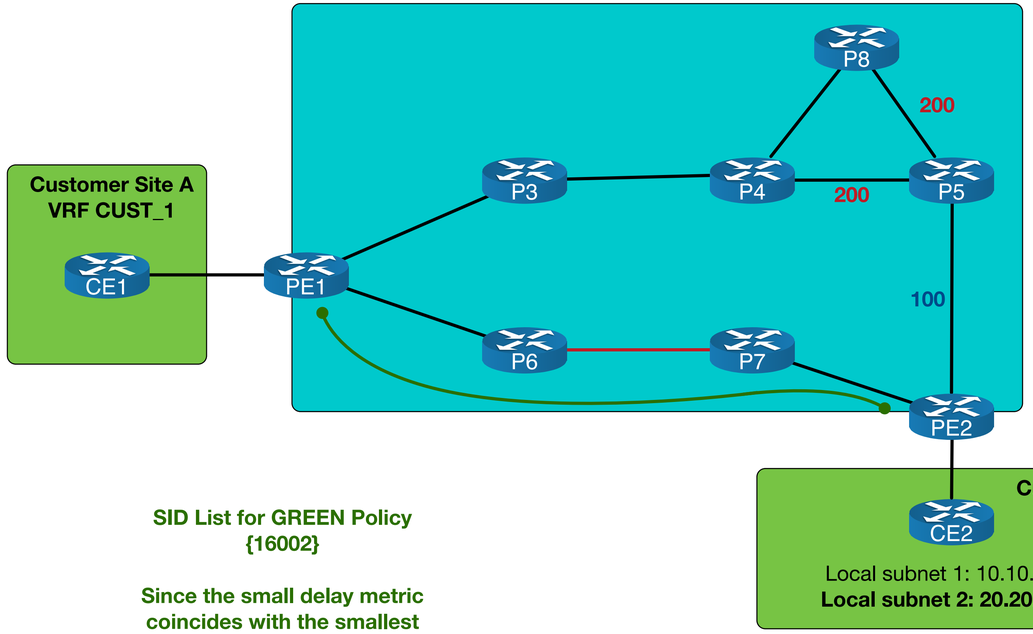


SID List for BLUE Policy
{16005}
{24052}

16005 will transport the packet to node P5 along shortest IGP path. 24052 corresponds to the Adj-SID between P5 and PE2.

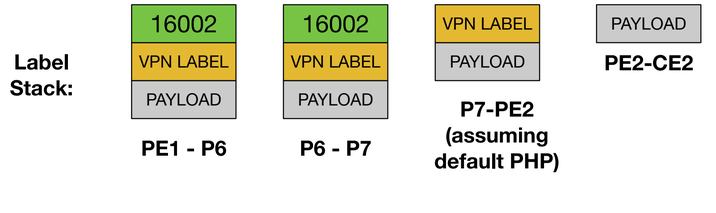


GREEN Policy Traffic Flow
(Smallest Delay Metric)



SID List for GREEN Policy
{16002}

Since the small delay metric coincides with the smallest IGP metric, all that ODN needs to do is create a policy with 16002 in the SID link. This will simply forward the packet on the IGP shortest path to PE2.

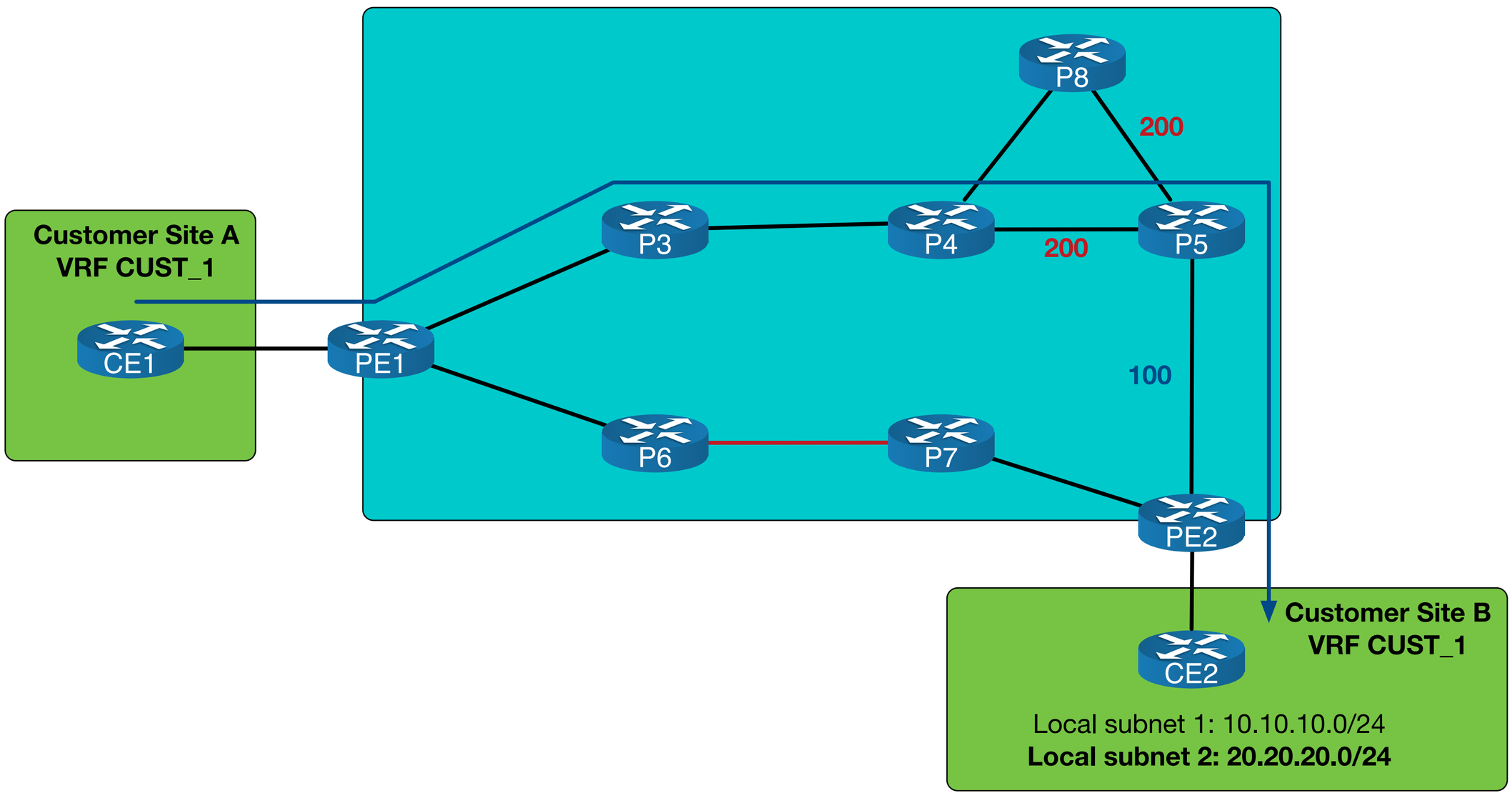


by Steven Crutchley

Segment Routing - On Demand Next Hop

Output - BLUE Policy

This page shows various CLI output from PE1 as it relates to the BLUE policy which specifies the lowest IGP metric avoiding red links.




```
RP/0/0/CPU0:PE1# show segment-routing traffic-eng policy
SR-TE policy database
---------------------

Color: 20, End-point: 2.2.2.2
  Name: srte_c_20_ep_2.2.2.2
    Status:
      Admin: up Operational: up for 00:12:15 (since Aug 3 17:12:47.734)
    Candidate-paths:
    Preference: 200 (BGP ODN) (active)
      Requested BSID: dynamic
      Constraints:
        Affinity:
          exclude-any:
          RED_LINK
        Dynamic (active)
          Metric Type: igp, Path Accumulated Metric: 130
            16005 [Prefix-SID, 5.5.5.5]
            24052 [Adjacency-SID, 99.2.5.5 99.2.5.2]
    Preference: 100 (BGP ODN)
      Requested BSID: dynamic
      PCC info:
        Symbolic name: bgp_c_20_ep_2.2.2.2_discr_100
        PLSP-ID: 20
      Dynamic (pce) (invalid)
        Metric Type: NONE, Path Accumulated Metric: 0
    Attributes:
      Binding SID: 40001
      Forward Class: 0
    Steering BGP disabled: no
    IPv6 caps enable: yes
```

The preference for a path calculated by the head-end will default to 200.

This was dynamically created using ODN

The total IGP metric over the path

A policy will try to be created from a PCE server. However in this case, because a PCE server is not configured, it shows up as invalid and is not considered.

```
RP/0/0/CPU0:PE1# show bgp vrf CUST_1 10.10.10.0/24
BGP routing table entry for 10.10.10.0/24, Route Distinguisher: 1.1.1.1:0
Versions:
  Process  bRIB/RIB  SendTblVer
  Speaker       561         561
Last Modified: Aug 3 16:59:23.321 for 00:12:04
Paths: (1 available, best #1)
  Not advertised to any peer
  Path #1: Received by speaker 0
  Not advertised to any peer
  2
    2.2.2.2 C:20 (bsid:40001) (metric 30) from 2.2.2.2 (2.2.2.2)
      Received Label 90004
      Origin IGP, metric 0, localpref 100, valid, internal, best, group-best, import-
candidate, imported
      Received Path ID 0, Local Path ID 1, version 556
      Extended community: Color:20 RT:1:1
      SR policy color 20, up, registered, bsid 40001, if-handle 0x00000490

      Source AFI: VPNv4 Unicast, Source VRF: default, Source Route Distinguisher: 2.2.2.2:0
```

BGP Color Community

Standard IGP metric to next-hop of 2.2.2.2

```
RP/0/0/CPU0:PE1# show cef vrf CUST_1 10.10.10.0/24
10.10.10.0/24, version 218, internal 0x5000001 0x0 (ptr 0xa13a0d78) [1], 0x0 (0x0), 0x208
(0xa175f44c)
  Updated Aug 3 17:00:01.025
  Prefix Len 24, traffic index 0, precedence n/a, priority 3
    via local-label 40001, 3 dependencies, recursive [flags 0x6000]
      path-idx 0 NHID 0x0 [0xa17d4288 0x0]
      recursion-via-label
      next hop VRF 'default', table 0xe0000000
      next hop via 40001/0/21
        next hop srte_c_20_ep_2.2.2.2 labels imposed {ImplNull 90004}
```

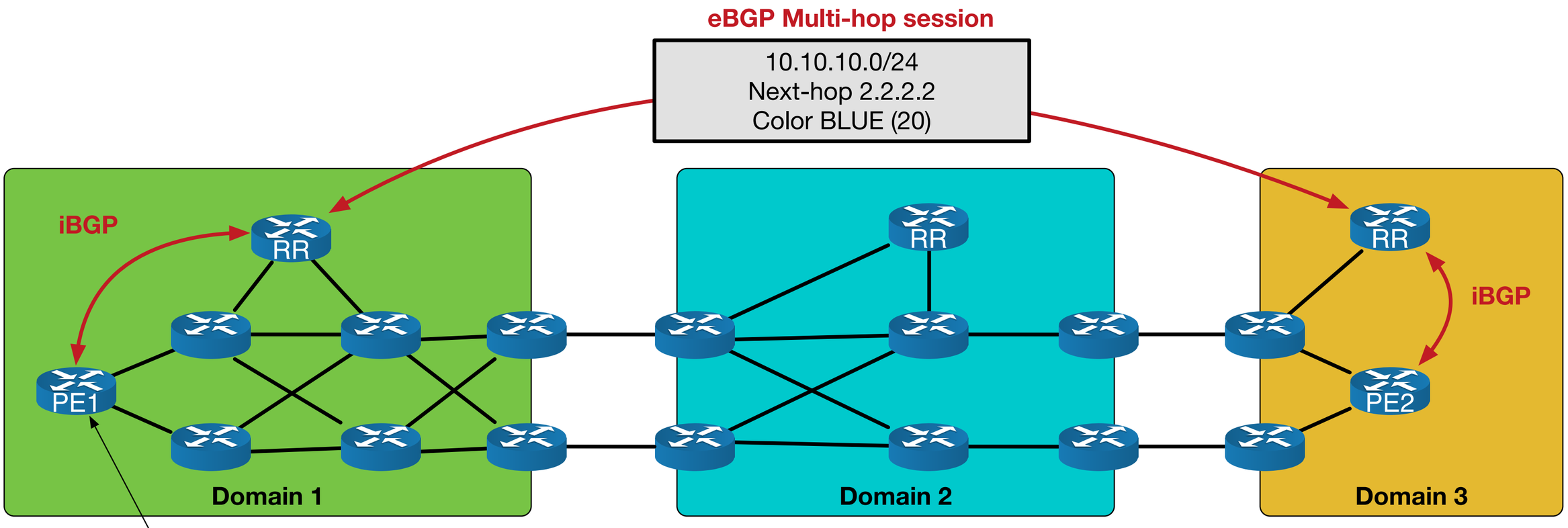
The ImplNull in this output indicates that the local BSID should be referenced (40001). This will in turn cause the calculated SID-List to be place on to the packet (with 90004, the VPN label, as the bottom of stack label)

Segment Routing - On Demand Next Hop

Interdomain Operation

Setup

The below diagram shows 3 separate domains. Full routing information is **not** passed between domains.



PE1 might not have an IP route to the next-hop (in this case 2.2.2.2). This could be provided using an aggregate route or less-specific null static that is redistributed.

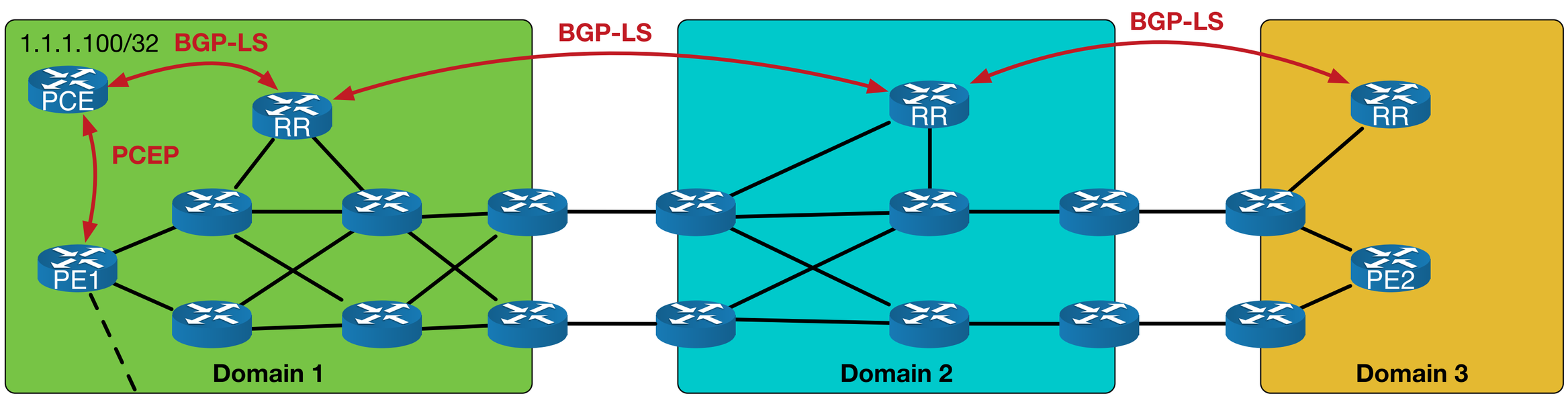
PE1 does not have full view of all the other domains. So cannot calculate the SR policy based on the optimisation and constraint criteria (e.g. It can't optimize the IGP metric or avoid certain links without knowing the IGP costs or affinity bits in Domain 2 and 3). To solve this a PCE is used.

Using a PCE

The PCE in the below topology has a view of all 3 domains (via BGP-LS sessions to multiple Refectors or other PCEs)

What is a PCE?

An SR Path Computation Element, or SR PCE, is a centralised compute engine that will calculate the SID list for the headend routers. Full information about all the domains is typically fed to a PCE using the BGP Link State address-family. Similar to how a route reflector gathers all prefixes so that a full iBGP mesh is not needed, a PCE allows for a central place to gather all the TE information without needing every headend to know the full inter-domain topology. A PCE can be a router or server. PCEP (PCE communication protocol) is, as the name suggests, used to communicate with the headend PCC (Path Computation Client).




```
segment-routing
  traffic-eng
    on-demand color 20
      dynamic
        pcep
        metric
          type delay
    !
    pcc
      pce address ipv4 1.1.1.100
```

This keyword makes PE1 ONLY use the PCEP to compute the path (it will not compute one locally)

```
RP/0/0/CPU0:PE1# show segment-routing traffic-eng policy
SR-TE policy database
---------------------
Color: 20, End-point: 2.2.2.2
  Name: srte_c_20_ep_2.2.2.2
    Status:
      Admin: up Operational: up for 00:12:33 (since Aug 2 08:43:12.118)
    Candidate-paths:
    Preference: 200 (BGP ODN) (shutdown)
      Requested BSID: dynamic
      Dynamic (invalid)
      Last error: No path found
  Preference: 100 (BGP ODN) (active)
    Requested BSID: dynamic
    PCC info:
      Symbolic name: bgp_c_20_ep_2.2.2.2_discr_100
      PLSP-ID: 32
    Dynamic (pce 1.1.1.100) (valid)
      Metric Type: delay, Path Accumulated Metric: 550
        16011 [Prefix-SID, 1.1.1.11]
        16021 [Prefix-SID, 20.2.2.1]
        16025 [Prefix-SID, 20.2.2.5]
        16032 [Prefix-SID, 2.2.2.2]
  Attributes:
    Binding SID: 40068
    Forward Class: 0
    Steering BGP disabled: no
    IPv6 caps enable: yes
```

Showing as shutdown since only PCEP is used

Calculated by PCEP all the way to PE2

ODN can used in combination with a default reachability model. For example if seamless MPLS is used (where routes are distributed between the various domains) ODN templates can be used to put some routes into SLA paths and leave others out.