



Simple and Flexible Segment Routing Solution by IP Infusion and Vegvisir

Deploying Robust and Reliable SR-TE and Egress Peer Engineering with OcNOS and Traffic Dictator

August 2025

Contents

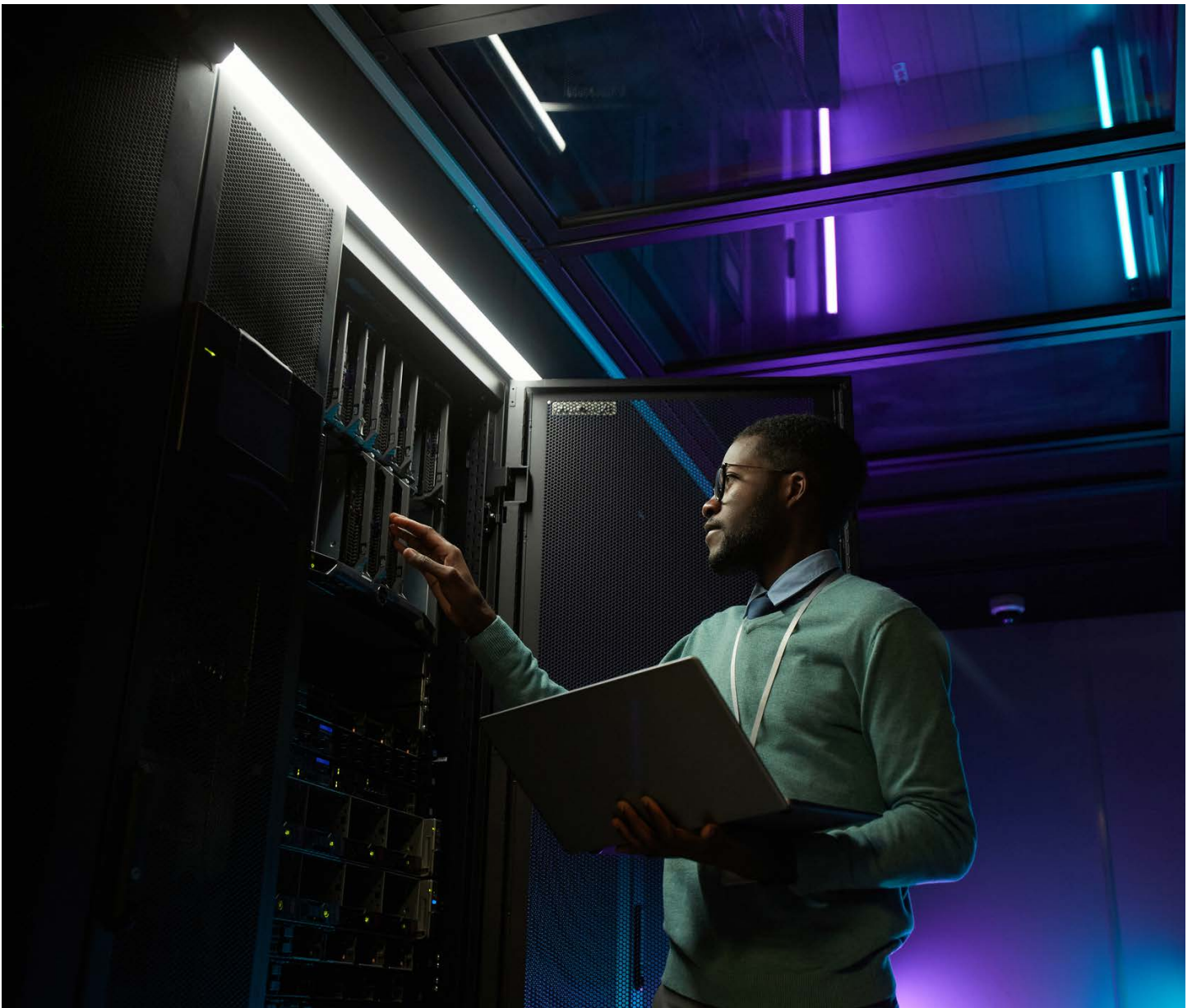
Executive summary	2
Software Defined Networking.....	3
Segment Routing	4
Robust and reliable.....	4
Egress Peer Engineering	5
Disaggregated Networking.....	6
About IP Infusion.....	7
About Vegvisir Systems	7
Design considerations	7
IS-IS.....	7
BGP Link-State.....	7
PCEP.....	7
Coupling MPLS services to transport planes.....	8
Lab topology	9
TE extensions and LSDB export	9
Node NLRI	13
Link NLRI.....	14
Prefix NLRI	15
PCEP session	15
SR-TE policy with explicit path.....	17
PCEP advertisement.....	19
Data plane verification.....	20
Bandwidth constraint and reservation	21
SR-TE policy with link affinity	23
Mapping traffic to SR-TE policies	28
Redundancy and failure scenarios.....	32
Dual PCE redundancy	32
Complete PCE failure	33
Path Diversity	34
ECMP and Anycast SID.....	36
Egress Peer Engineering.....	39
Affinity and bandwidth constraints in EPE policies.....	42
TD Graphical Interface.....	44
Conclusion.....	46
Authors.....	46
Appendix A: references	46
Appendix B: lab configs.....	47
TD.....	47
R1	49
R2	50
R3	51
R4	53
R5	55

Executive summary

Modern Carrier networks are required to support a variety of services, including Broadband, Cloud, 5G and must have the ability to quickly deploy new service functions, while also meeting the constantly increasing bandwidth demands and remaining robust during outages and cyberattacks, quickly rerouting traffic around failed network elements without causing congestion.

It is essential that network configuration tasks can be automated – which ensures fast deployment of services and eliminates the risk of human error.

Network operators prefer to avoid vendor lock-in whenever possible, and use open-standard solutions they can better understand, customize and replace each component when required, without having to replace an entire network. This approach also reduces costs for the network operator.



Software Defined Networking

Software Defined Networking (SDN) is a concept whereby advanced networking features are handled by an SDN controller, so that deploying a new service requires just configuration or software upgrade of the controller, rather than making changes on hundreds or thousands of devices. SDN also allows for more efficient routing, because the controller can see the state of the entire network and calculate more optimal paths.

However, with SDN, it is essential to have a balanced approach, so that the SDN controller does not cause vendor lock-in and can be replaced when required, also the network must remain operational should the SDN controller fail.

The communication interface between the SDN controller and the network is known as the *Southbound interface*.

The communication interface between the SDN controller and the higher-level application is known as the *Northbound interface*.

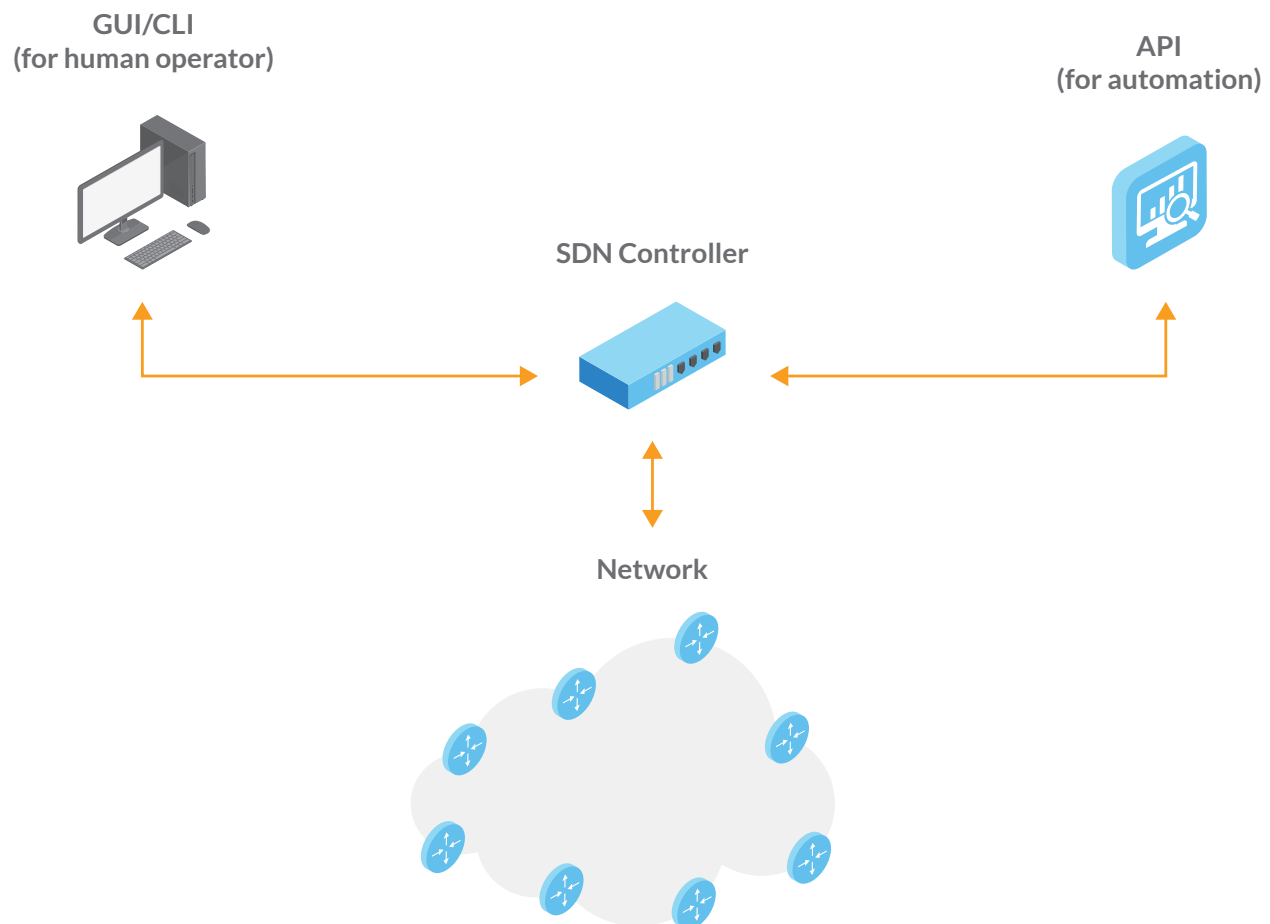


Fig. 1: The illustration of the SDN concept.

Segment Routing

Segment Routing (SR) is a method of source routing that provides a simple, stateless mechanism to program the path the packet takes through the network. SR accomplishes the same thing as MPLS but is much simpler, scalable, flexible and has support for many routing techniques the traditional MPLS struggles with (such as ECMP and Anycast routing).

Segment Routing Traffic Engineering (SR-TE) allows the network operator to optimize routing path for different types of traffic by different criteria (e.g. cheaper bandwidth or minimum delay) and efficiently utilize existing network resources.

Robust and reliable

In the present geopolitical situation, networks regularly experience not only technical failures, but also targeted attacks on infrastructure, such as submarine cable cuts, wireless signal jamming and cyberattacks. The network must adequately react to those failures and attacks, providing the best possible service even with reduced capacity.

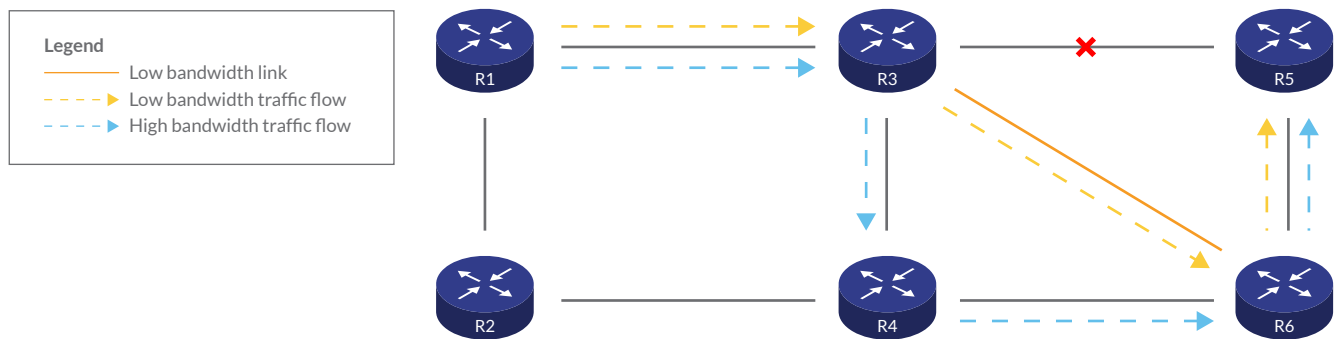


Fig. 2: SR reroutes traffic after link failure.

SR provides sub-50ms reroute after link or node failures with TI-LFA. An SDN controller can help route traffic more optimally, taking bandwidth into account and preventing congestion after link failures. In the example on figure 2, if the link between R3 and R5 fails, rerouting all traffic via the shortest path would cause congestion on the link between R3 and R6, the SDN controller prevents that, by taking required bandwidth into account when calculating SR-TE policies.

SR finds the optimal balance between distributed and centralized networking: basic network connectivity is provided by regular routing protocols such as IS-IS and OSPF and doesn't depend on an SDN controller, but an optional controller can be deployed to provide advanced traffic engineering features.

Egress Peer Engineering

Egress Peer Engineering (EPE) allows the network operator to override default BGP routing behaviour by having the ingress router define which exit should the traffic routed outside the network take.

For example, if AS100 has multiple peerings with AS200 and one of those peerings fail, that might cause congestion on the remaining peerings. With bandwidth-aware EPE, it is possible to prevent this by rerouting some traffic via the Internet.

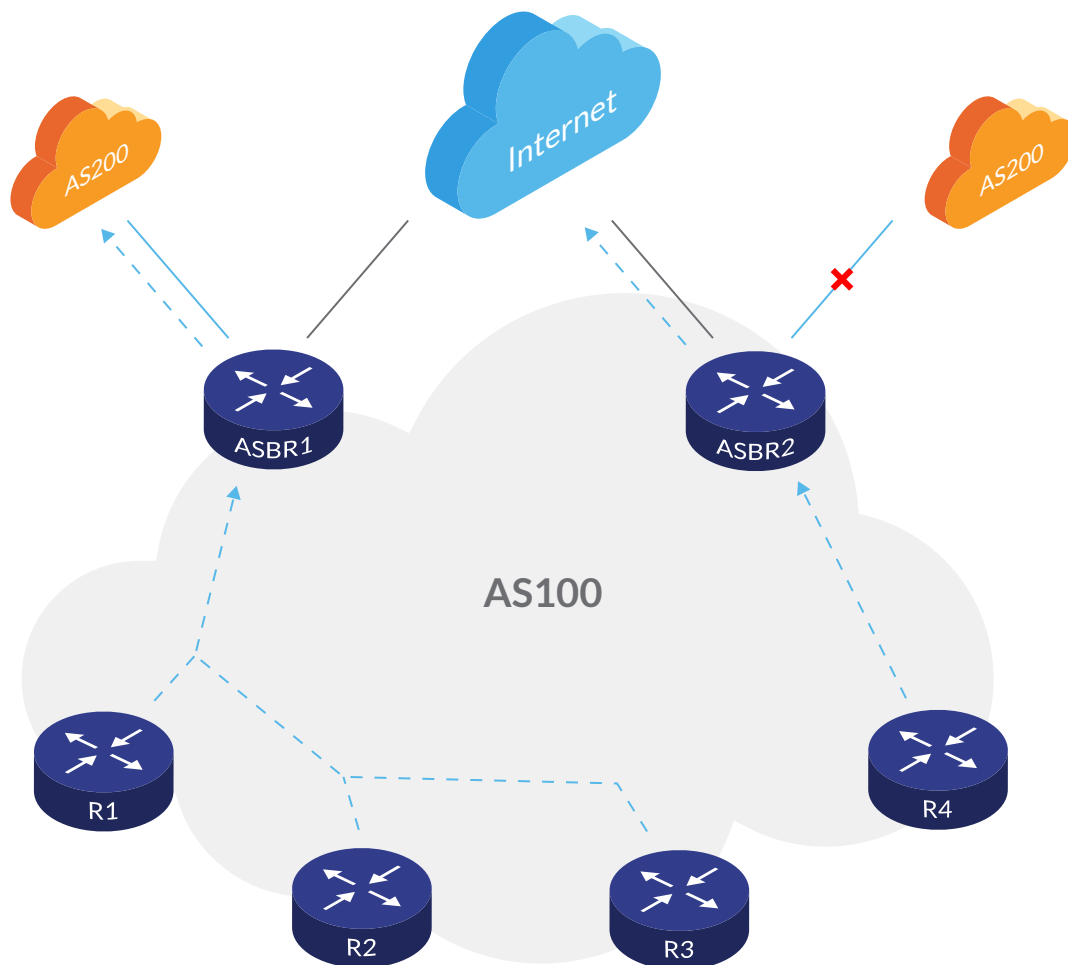


Fig. 3: After one link to AS200 fails, some traffic is rerouted to the Internet, preventing congestion on the remaining link to AS200.

Disaggregated Networking

Disaggregated Networking is a concept whereby network hardware is separated from software, so that the network operator can choose hardware and software vendors separately. This provides more flexibility to the operator, reduces costs, prevents vendor lock-in and reduces the risk of supply chain disruption.

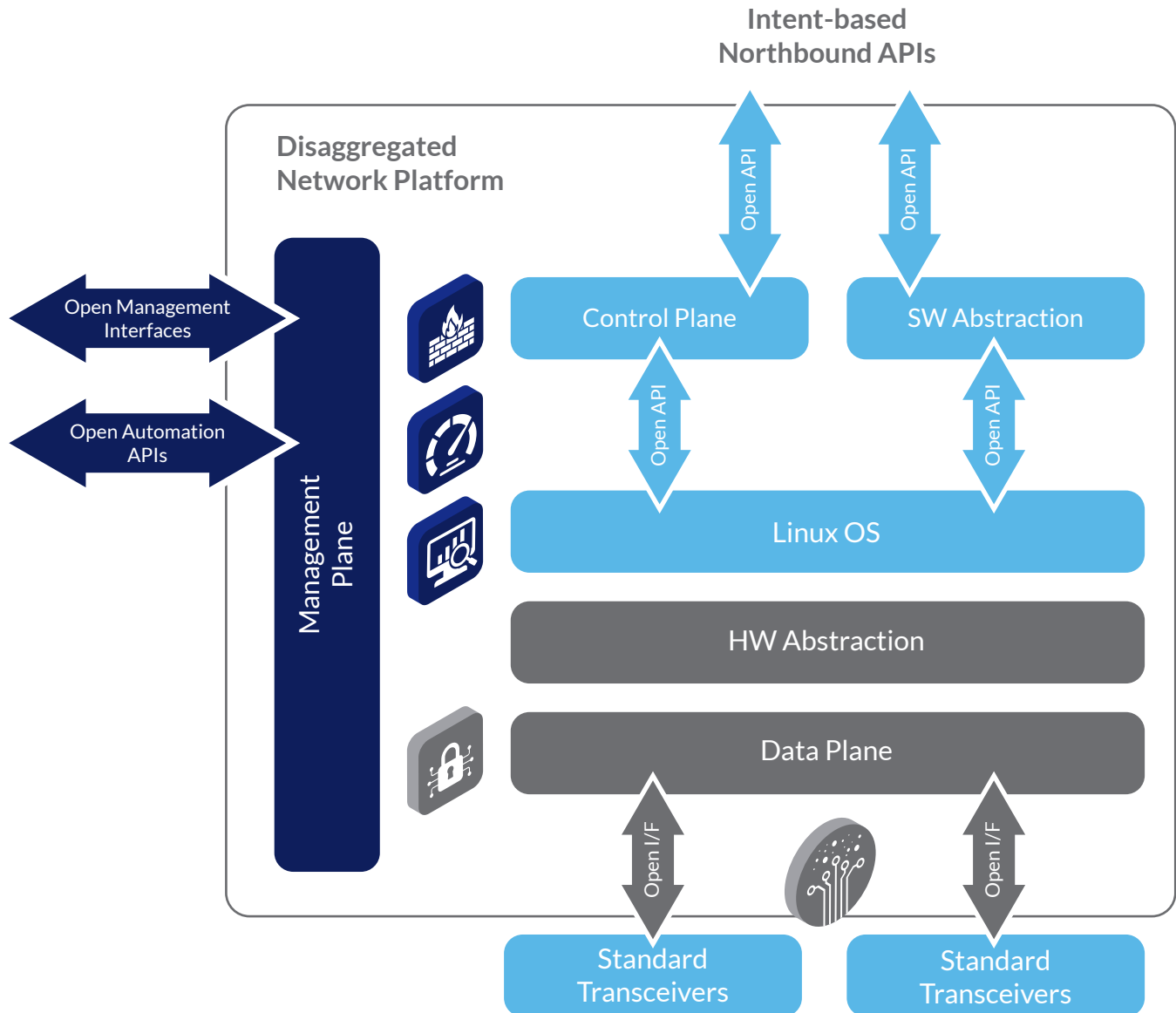


Fig. 4: Disaggregated networking device.

About IP Infusion

IP Infusion is a leading provider of disaggregated networking solutions for telecom and data communications operators. OcNOS, their network operating system, can run on various hardware platforms, including UfiSpace, EdgeCore, Dell and others; and supports a full set of routing features required by modern carrier networks: BGP, IS-IS, MPLS, Segment Routing, SRv6, EVPN, VPLS etc. It also features an industry-standard CLI and Netconf API for network automation. More information can be found at www.ipinfusion.com

About Vegvisir Systems

Vegvisir Systems is a startup that provides open standard and affordable traffic engineering solutions for carrier and cloud networks. Traffic Dictator is a lightweight, user-friendly, multi-vendor SDN controller that supports traffic engineering with Segment Routing, RSVP, and Egress Peer Engineering. More information can be found at vegvisir.ie

Design considerations

Having considered all of the above, let's review the components of the solution and justify each decision.

IS-IS

There are 2 mainstream IGPs: IS-IS and OSPF. While in theory they are very similar, IS-IS is the IGP of choice for most carrier networks, due to its simplicity, extensibility and support for IPv4 and IPv6. Since it is the most commonly used IGP, most vendors develop new features primarily for IS-IS and not OSPF.

We will be using IS-IS with Segment Routing extensions. This eliminates the need for extra label protocols such as LDP or RSVP.

BGP Link-State

BGP Link-State is the standard southbound interface used to export IGP topology to the SDN controller. OcNOS exports the IS-IS topology into BGP-LS and advertises it to Traffic Dictator.

PCEP

PCEP is the most popular protocol for communication between the router (PCC) and controller (PCE). It is stateful (i.e. PCE knows if the PCC has installed the policy and is forwarding traffic), supports both PCE and PCC-initiated policies and is widely supported in the industry. PCEP supports both SR-TE and RSVP-TE LSP, which is useful when migrating from legacy MPLS-TE to SR-TE.

Coupling MPLS services to transport planes

Once TD calculated SR-TE policies with the defined constraint and advertised them to the network, we have to map different services (IP transit, L3VPN, EVPN etc) to the relevant policies. Multiple services configured on the same router are advertised with the same BGP nexthop, but they might have different traffic engineering requirements and have to be mapped to different policies.

With SR-TE, there are 2 possible ways to achieve that:

- 1) Color steering: BGP routes are advertised with color extended community, and the headend maps routes to the SR-TE policies with matching color and nexthop
- 2) Service loopback steering: BGP routes from different services are advertised with different nexthops, and the headend maps those routes to the respective SR-TE policies advertised by controller

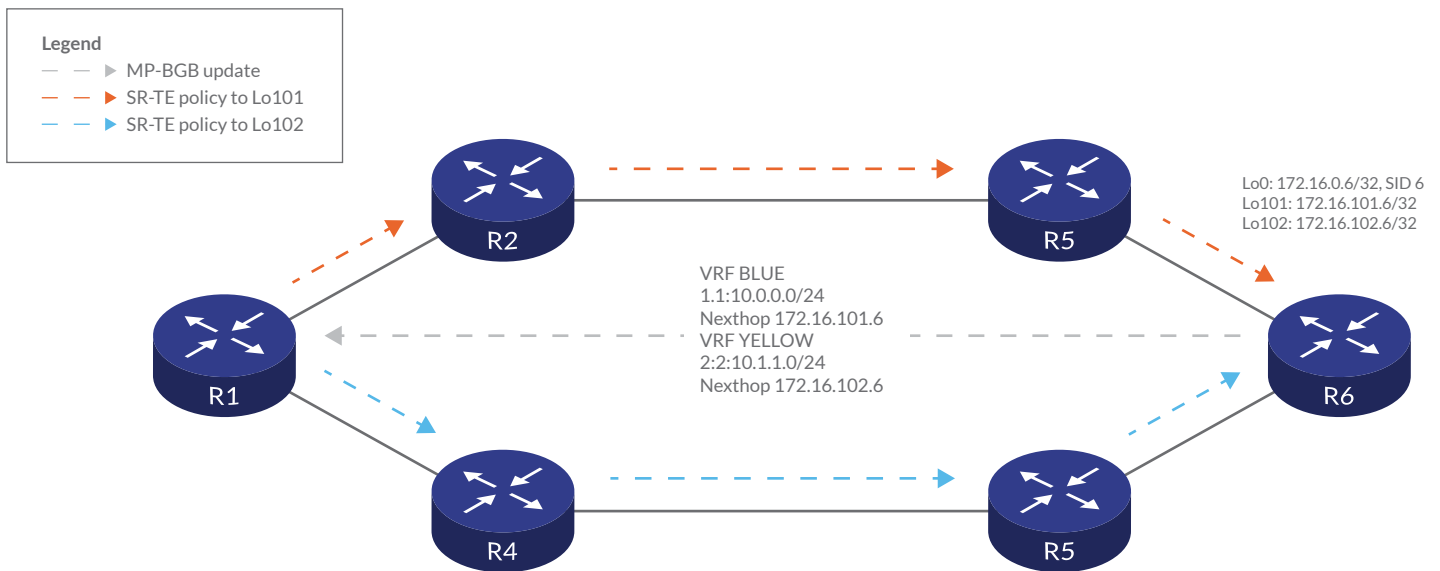


Fig. 5: Mapping different services to different SR-TE policies.

Color steering gives the operator a lot of flexibility and enables features like On-Demand Nexthop (ODN), Null endpoint policies and BGP Classful Transport (BGP-CT). However, it has not been fully standardized yet and requires support not only in PCEP but also in BGP services. It also lacks backwards compatibility with legacy MPLS-TE.

Service loopback steering provides robust coverage in multi-vendor environments allowing for seamless integration across vendors. It also works with RSVP-TE enabling a simple migration path to SR-TE without modifying existing service configurations.

For this demonstration, we will be using service loopback steering.

Lab topology

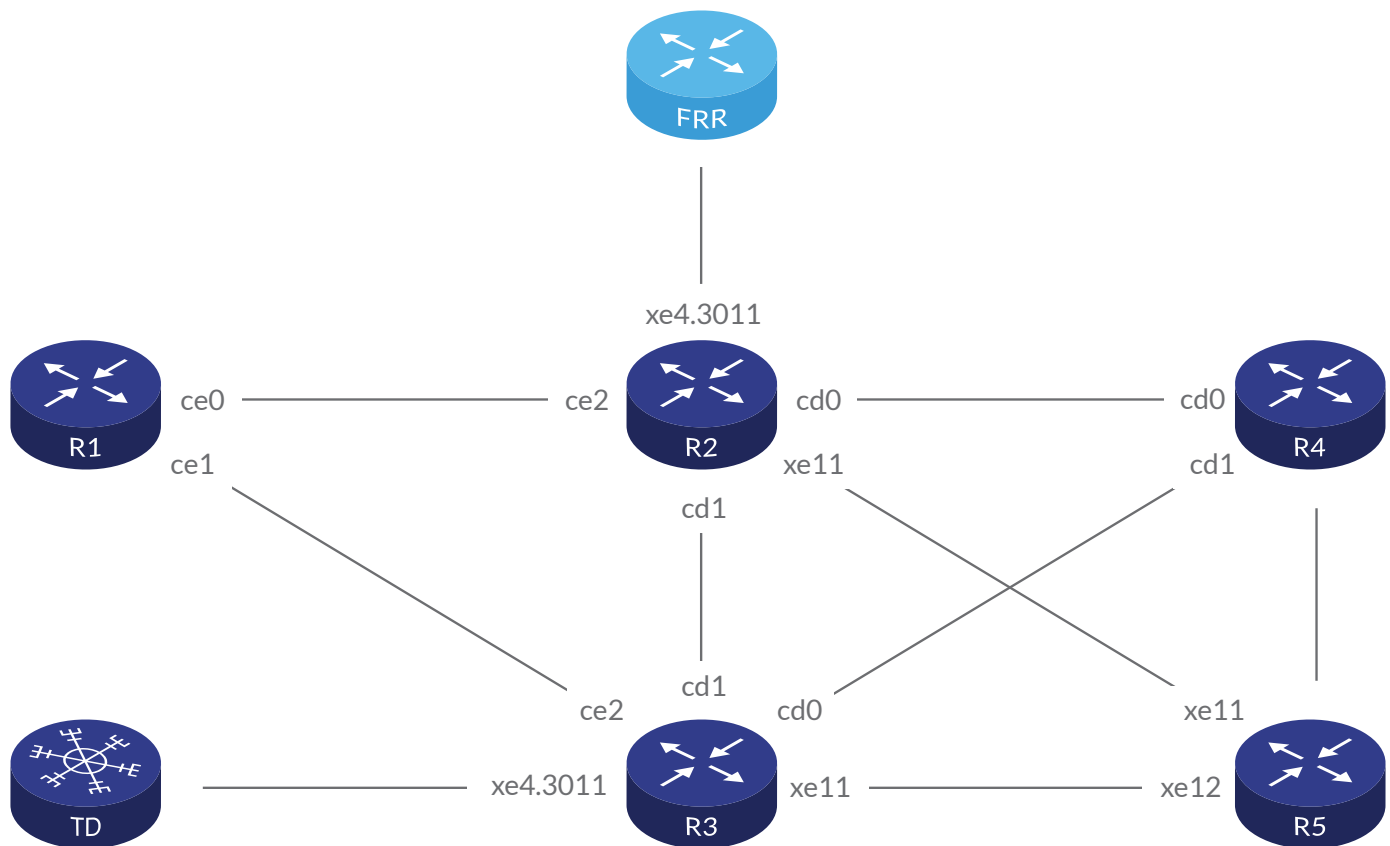


Fig. 6: Lab topology used for the PoC.

To demonstrate SR-TE configuration and operations with TD and OcNOS, we are using UfiSpace 9510-28DC routers with OcNOS 6.6.0 and TD 1.6. A virtual machine with Free Range Routing (FRR) is used to simulate external peers for EPE demonstration.

TE extensions and LSDB export

IS-IS with Segment Routing extensions is enabled on all routers, also Traffic Engineering extensions are enabled to advertise link bandwidth and admin groups (or colors).

Config example:

```
hostname R1
admin-group BLUE 0
admin-group YELLOW 1
!
segment-routing
  global block 16000 23999
!
interface ce0
  ip address 10.100.1.1/24
  label-switching
  admin-group BLUE
  isis network point-to-point
  isis circuit-type level-2-only
  ip router isis 1
!
interface lo
  ip address 127.0.0.1/8
  ip address 1.1.1.1/32 secondary
  prefix-sid index 1
  ip router isis 1
!
router isis 1
  is-type level-2-only
  metric-style wide
  mpls traffic-eng router-id 1.1.1.1
  mpls traffic-eng level-2
  dynamic-hostname
  net 49.0001.0001.0001.00
  segment-routing mpls
```

Note: full configs for all routers are at the end of this document.

R3 has a BGP-LS session with TD. Therefore, we must distribute IS-IS LSDB into BGP-LS in order to advertise it to TD. Let's add the following config to R3:

```
router isis 1
  distribute bgp-ls
!
router bgp 65002
  bgp router-id 3.3.3.3
  neighbor 192.168.123.1 remote-as 65001
!
  address-family link-state link-state
  neighbor 192.168.123.1 activate
  exit-address-family
```

BGP config on TD:

```
router bgp 65001
  router-id 192.168.123.1
  !
  neighbor 192.168.123.3
    remote-as 65002
  address-family link-state
```

Verify BGP-LS routes on OcNOS:

```
R3#show bgp link-state link-state
BGP router identifier 3.3.3.3, local AS number 65002
Origin codes: i - IGP, e - EGP
Prefix codes: E link, V node, T IP reachable route, S SRv6 SID
               I Identifier N local node, R remote node, L link P prefix, S SID
               N local node, R remote node, L link, P prefix
               L1/L2 ISIS level-1/level-2, O OSPF, O6 OSPFv3, a area-ID,
               c confed-ID/ASN, b bgp-identifier, r router-ID, s SID,
               i if-address, n nbr-address, o OSPF Route-type,
               p IP-prefix, d designated router address, s ISO-ID
[V] [L2] [IOx1] [N[c65002] [b3.3.3.3] [s0001.0001.0001.00]]/208
[V] [L2] [IOx1] [N[c65002] [b3.3.3.3] [s0002.0002.0002.00]]/208
[V] [L2] [IOx1] [N[c65002] [b3.3.3.3] [s0003.0003.0003.00]]/208
[V] [L2] [IOx1] [N[c65002] [b3.3.3.3] [s0004.0004.0004.00]]/208
[V] [L2] [IOx1] [N[c65002] [b3.3.3.3] [s0005.0005.0005.00]]/208
[E] [L2] [IOx1] [N[c65002] [b3.3.3.3] [s0001.0001.0001.00]] [R[c65002] [b3.3.3.3]
[s0002.0002.0002.00]] [L[i10.100.1.1] [n10.100.1.2]]/336
[E] [L2] [IOx1] [N[c65002] [b3.3.3.3] [s0002.0002.0002.00]] [R[c65002] [b3.3.3.3]
[s0001.0001.0001.00]] [L[i10.100.1.2] [n10.100.1.1]]/336
[E] [L2] [IOx1] [N[c65002] [b3.3.3.3] [s0002.0002.0002.00]] [R[c65002] [b3.3.3.3]
[s0004.0004.0004.00]] [L[i10.100.4.2] [n10.100.4.4]]/336
[E] [L2] [IOx1] [N[c65002] [b3.3.3.3] [s0002.0002.0002.00]] [R[c65002] [b3.3.3.3]
[s0005.0005.0005.00]] [L[i10.100.5.2] [n10.100.5.5]]/336
[E] [L2] [IOx1] [N[c65002] [b3.3.3.3] [s0003.0003.0003.00]] [R[c65002] [b3.3.3.3]
[s0004.0004.0004.00]] [L[i10.100.5.3] [n10.100.5.4]]/336
[E] [L2] [IOx1] [N[c65002] [b3.3.3.3] [s0003.0003.0003.00]] [R[c65002] [b3.3.3.3]
[s0005.0005.0005.00]] [L[i10.100.7.3] [n10.100.7.5]]/336
[E] [L2] [IOx1] [N[c65002] [b3.3.3.3] [s0004.0004.0004.00]] [R[c65002] [b3.3.3.3]
[s0002.0002.0002.00]] [L[i10.100.4.4] [n10.100.4.2]]/336
[E] [L2] [IOx1] [N[c65002] [b3.3.3.3] [s0004.0004.0004.00]] [R[c65002] [b3.3.3.3]
[s0003.0003.0003.00]] [L[i10.100.5.4] [n10.100.5.3]]/336
[E] [L2] [IOx1] [N[c65002] [b3.3.3.3] [s0005.0005.0005.00]] [R[c65002] [b3.3.3.3]
[s0002.0002.0002.00]] [L[i10.100.5.5] [n10.100.5.2]]/336
[E] [L2] [IOx1] [N[c65002] [b3.3.3.3] [s0005.0005.0005.00]] [R[c65002] [b3.3.3.3]
[s0003.0003.0003.00]] [L[i10.100.7.5] [n10.100.7.3]]/336
[T] [L2] [IOx1] [N[c65002] [b3.3.3.3] [s0001.0001.0001.00]] [P[p10.100.1.0/24]]/240
[T] [L2] [IOx1] [N[c65002] [b3.3.3.3] [s0001.0001.0001.00]] [P[p10.100.2.0/24]]/240
[T] [L2] [IOx1] [N[c65002] [b3.3.3.3] [s0001.0001.0001.00]] [P[p1.1.1.1/32]]/248
[T] [L2] [IOx1] [N[c65002] [b3.3.3.3] [s0002.0002.0002.00]] [P[p10.100.1.0/24]]/240
[T] [L2] [IOx1] [N[c65002] [b3.3.3.3] [s0002.0002.0002.00]] [P[p10.100.3.0/24]]/240
[T] [L2] [IOx1] [N[c65002] [b3.3.3.3] [s0002.0002.0002.00]] [P[p10.100.4.0/24]]/240
[T] [L2] [IOx1] [N[c65002] [b3.3.3.3] [s0002.0002.0002.00]] [P[p10.100.5.0/24]]/240
[T] [L2] [IOx1] [N[c65002] [b3.3.3.3] [s0002.0002.0002.00]] [P[p2.2.2.2/32]]/248
[T] [L2] [IOx1] [N[c65002] [b3.3.3.3] [s0003.0003.0003.00]] [P[p10.100.2.0/24]]/240
[T] [L2] [IOx1] [N[c65002] [b3.3.3.3] [s0003.0003.0003.00]] [P[p10.100.3.0/24]]/240
[T] [L2] [IOx1] [N[c65002] [b3.3.3.3] [s0003.0003.0003.00]] [P[p10.100.5.0/24]]/240
[T] [L2] [IOx1] [N[c65002] [b3.3.3.3] [s0003.0003.0003.00]] [P[p10.100.7.0/24]]/240
[T] [L2] [IOx1] [N[c65002] [b3.3.3.3] [s0003.0003.0003.00]] [P[p3.3.3.3/32]]/248
[T] [L2] [IOx1] [N[c65002] [b3.3.3.3] [s0004.0004.0004.00]] [P[p10.100.4.0/24]]/240
[T] [L2] [IOx1] [N[c65002] [b3.3.3.3] [s0004.0004.0004.00]] [P[p10.100.5.0/24]]/240
[T] [L2] [IOx1] [N[c65002] [b3.3.3.3] [s0004.0004.0004.00]] [P[p4.4.4.4/32]]/248
[T] [L2] [IOx1] [N[c65002] [b3.3.3.3] [s0005.0005.0005.00]] [P[p10.100.5.0/24]]/240
[T] [L2] [IOx1] [N[c65002] [b3.3.3.3] [s0005.0005.0005.00]] [P[p10.100.7.0/24]]/240
[T] [L2] [IOx1] [N[c65002] [b3.3.3.3] [s0005.0005.0005.00]] [P[p5.5.5.5/32]]/248
NLRI's, Total: 34, Node: 5, Link: 10, Prefix: 19, SRv6-SID: 0
```

Verify received BGP-LS routes on TD:

```

TD1#show bgp link-state
BGP-LS routing table information
Router identifier 192.168.123.3, local AS number 65001
Status codes: * valid, > best
Origin codes: i - IGP, e - EGP, ? - incomplete
Prefix codes: E link, V node, T IP reachable route, S SRv6 SID, u/U unknown,
               I Identifier, N local node, R remote node, L link, P prefix, S SID,
               L1/L2 ISIS level-1/level-2, O OSPF, D direct, S static/peer-node,
               a area-ID, l link-ID, t topology-ID, s ISO-ID,
               c confed-ID/ASN, b bgp-identifier, r router-ID, s SID,
               i if-address, n nbr-address, o OSPF Route-type, p IP-prefix,
               d designated router address

```

Weight	Network	Next Hop	Metric	LocPref
	[V] [L2] [I1] [N[c65002] [b50529027] [s0001.0001.0001.00]]			
*>		192.168.123.3	0	100
0	65002 i			
	[V] [L2] [I1] [N[c65002] [b50529027] [s0002.0002.0002.00]]			
*>		192.168.123.3	0	100
0	65002 i			
	[V] [L2] [I1] [N[c65002] [b50529027] [s0003.0003.0003.00]]			
*>		192.168.123.3	0	100
0	65002 i			
	[V] [L2] [I1] [N[c65002] [b50529027] [s0004.0004.0004.00]]			
*>		192.168.123.3	0	100
0	65002 i			
	[V] [L2] [I1] [N[c65002] [b50529027] [s0005.0005.0005.00]]			
*>		192.168.123.3	0	100
0	65002 i			
	[E] [L2] [I1] [N[c65002] [b50529027] [s0001.0001.0001.00]] [R[c65002] [b50529027] [s0002.0002.0002.00]] [L[i10.100.1.1] [n10.100.1.2]]			
*>		192.168.123.3	0	100
0	65002 i			
	[E] [L2] [I1] [N[c65002] [b50529027] [s0002.0002.0002.00]] [R[c65002] [b50529027] [s0001.0001.0001.00]] [L[i10.100.1.2] [n10.100.1.1]]			
*>		192.168.123.3	0	100
0	65002 i			
	[E] [L2] [I1] [N[c65002] [b50529027] [s0002.0002.0002.00]] [R[c65002] [b50529027] [s0004.0004.0004.00]] [L[i10.100.4.2] [n10.100.4.4]]			
*>		192.168.123.3	0	100
0	65002 i			
	[E] [L2] [I1] [N[c65002] [b50529027] [s0002.0002.0002.00]] [R[c65002] [b50529027] [s0005.0005.0005.00]] [L[i10.100.5.2] [n10.100.5.5]]			
*>		192.168.123.3	0	100
0	65002 i			
	[E] [L2] [I1] [N[c65002] [b50529027] [s0003.0003.0003.00]] [R[c65002] [b50529027] [s0004.0004.0004.00]] [L[i10.100.5.3] [n10.100.5.4]]			
*>		192.168.123.3	0	100
0	65002 i			
	[E] [L2] [I1] [N[c65002] [b50529027] [s0003.0003.0003.00]] [R[c65002] [b50529027] [s0005.0005.0005.00]] [L[i10.100.7.3] [n10.100.7.5]]			
*>		192.168.123.3	0	100
0	65002 i			
	[E] [L2] [I1] [N[c65002] [b50529027] [s0004.0004.0004.00]] [R[c65002] [b50529027] [s0002.0002.0002.00]] [L[i10.100.4.4] [n10.100.4.2]]			
*>		192.168.123.3	0	100
0	65002 i			
	[E] [L2] [I1] [N[c65002] [b50529027] [s0004.0004.0004.00]] [R[c65002] [b50529027] [s0003.0003.0003.00]] [L[i10.100.5.4] [n10.100.5.3]]			
*>		192.168.123.3	0	100
0	65002 i			
	[E] [L2] [I1] [N[c65002] [b50529027] [s0005.0005.0005.00]] [R[c65002] [b50529027]			

```

[s0002.0002.0002.00]] [L[i10.100.5.5] [n10.100.5.2]]
*> 192.168.123.3 0 100
0 65002 i
[E] [L2] [I1] [N[c65002] [b50529027] [s0005.0005.0005.00]] [R[c65002] [b50529027]
[s0003.0003.0003.00]] [L[i10.100.7.5] [n10.100.7.3]]
*> 192.168.123.3 0 100
0 65002 i
[T] [L2] [I1] [N[c65002] [b50529027] [s0001.0001.0001.00]] [P[p10.100.1.0/24]]
*> 192.168.123.3 0 100
0 65002 i
[T] [L2] [I1] [N[c65002] [b50529027] [s0001.0001.0001.00]] [P[p10.100.2.0/24]]
*> 192.168.123.3 0 100
0 65002 i
[T] [L2] [I1] [N[c65002] [b50529027] [s0002.0002.0002.00]] [P[p10.100.1.0/24]]
*> 192.168.123.3 0 100
0 65002 i
[T] [L2] [I1] [N[c65002] [b50529027] [s0002.0002.0002.00]] [P[p10.100.3.0/24]]
*> 192.168.123.3 0 100
0 65002 i
[T] [L2] [I1] [N[c65002] [b50529027] [s0002.0002.0002.00]] [P[p10.100.4.0/24]]
*> 192.168.123.3 0 100
0 65002 i
[T] [L2] [I1] [N[c65002] [b50529027] [s0002.0002.0002.00]] [P[p10.100.5.0/24]]
*> 192.168.123.3 0 100
0 65002 i
[T] [L2] [I1] [N[c65002] [b50529027] [s0003.0003.0003.00]] [P[p10.100.2.0/24]]
*> 192.168.123.3 0 100
0 65002 i
[T] [L2] [I1] [N[c65002] [b50529027] [s0003.0003.0003.00]] [P[p10.100.3.0/24]]
*> 192.168.123.3 0 100
0 65002 i
[T] [L2] [I1] [N[c65002] [b50529027] [s0003.0003.0003.00]] [P[p10.100.5.0/24]]
*> 192.168.123.3 0 100
0 65002 i
[T] [L2] [I1] [N[c65002] [b50529027] [s0003.0003.0003.00]] [P[p10.100.7.0/24]]
*> 192.168.123.3 0 100
0 65002 i
[T] [L2] [I1] [N[c65002] [b50529027] [s0004.0004.0004.00]] [P[p10.100.4.0/24]]
*> 192.168.123.3 0 100
0 65002 i
[T] [L2] [I1] [N[c65002] [b50529027] [s0004.0004.0004.00]] [P[p10.100.5.0/24]]
*> 192.168.123.3 0 100
0 65002 i
[T] [L2] [I1] [N[c65002] [b50529027] [s0005.0005.0005.00]] [P[p10.100.5.0/24]]
*> 192.168.123.3 0 100
0 65002 i
[T] [L2] [I1] [N[c65002] [b50529027] [s0005.0005.0005.00]] [P[p10.100.7.0/24]]
*> 192.168.123.3 0 100
0 65002 i
[T] [L2] [I1] [N[c65002] [b50529027] [s0001.0001.0001.00]] [P[p1.1.1.1/32]]
*> 192.168.123.3 0 100
0 65002 i
[T] [L2] [I1] [N[c65002] [b50529027] [s0002.0002.0002.00]] [P[p2.2.2.2/32]]
*> 192.168.123.3 0 100
0 65002 i
[T] [L2] [I1] [N[c65002] [b50529027] [s0003.0003.0003.00]] [P[p3.3.3.3/32]]
*> 192.168.123.3 0 100
0 65002 i
[T] [L2] [I1] [N[c65002] [b50529027] [s0004.0004.0004.00]] [P[p4.4.4.4/32]]
*> 192.168.123.3 0 100
0 65002 i
[T] [L2] [I1] [N[c65002] [b50529027] [s0005.0005.0005.00]] [P[p5.5.5.5/32]]
*> 192.168.123.3 0 100
0 65002 i

```

There are 3 types of BGP-LS NLRI: Node, Link and Prefix.

Node NLRI

Node NLRI includes IS-IS system-ID, TE router ID and various SR parameters such as Segment Routing Global Block (SRGB) that is used to calculate segment values.

```
TD1#show bgp link-state [V][L2][I1][N[c65002][b50529027][s0001.0001.0001.00]]
detail
BGP-LS routing table information
Router identifier 192.168.123.3, local AS number 65001
Prefix codes: E link, V node, T IP reachable route, S SRv6 SID, u/U unknown,
               I Identifier, N local node, R remote node, L link, P prefix, S SID,
               L1/L2 ISIS level-1/level-2, O OSPF, D direct, S static/peer-node,
               a area-ID, l link-ID, t topology-ID, s ISO-ID,
               c confed-ID/ASN, b bgp-identifier, r router-ID, s SID,
               i if-address, n nbr-address, o OSPF Route-type, p IP-prefix,
               d designated router address

BGP routing table entry for [V][L2][I1][N[c65002][b50529027][s0001.0001.0001.00]]
NLRI Type: node
Protocol: ISIS L2
Identifier: 1
Local Node Descriptor:
  AS Number: 65002
  BGP Identifier: 50529027
  ISO Node ID: 0001.0001.0001.00
Paths: 1 available, best #1
  Last modified: March 03, 2025 16:30:21
  65002
    192.168.123.3 from 192.168.123.3 (3.3.3.3)
      Origin igp, metric 0, localpref 100, weight 0, valid, external, best
      Link-state: ISIS area ID: 49, Local TE IPv4 Router-ID: 1.1.1.1,
                  SRGB Base: 16000, Range: 8000,
                  SR Capability Flags: {'mpls_ipv4': False, 'mpls_ipv6': False,
                  'srv6': False},
                  SR Algorithm: [0]
```

Link NLRI

Link NLRI includes link parameters like local and remote IP addresses, metric, admin group, bandwidth and adjacency SID.

```
TD1#show bgp link-state
[E][L2][I1][N[c65002][b50529027][s0001.0001.0001.00]][R[c65002][b50529027]
[s0002.0002.0002.00]][L[i10.100.1.1][n10.100.1.2]] detail
BGP-LS routing table information
Router identifier 192.168.123.3, local AS number 65001
Prefix codes: E link, V node, T IP reachable route, S SRv6 SID, u/U unknown,
               I Identifier, N local node, R remote node, L link, P prefix, S SID,
               L1/L2 ISIS level-1/level-2, O OSPF, D direct, S static/peer-node,
               a area-ID, l link-ID, t topology-ID, s ISO-ID,
               c confed-ID/ASN, b bgp-identifier, r router-ID, s SID,
               i if-address, n nbr-address, o OSPF Route-type, p IP-prefix,
               d designated router address

BGP routing table entry for
[E][L2][I1][N[c65002][b50529027][s0001.0001.0001.00]][R[c65002][b50529027]
[s0002.0002.0002.00]][L[i10.100.1.1][n10.100.1.2]]
NLRI Type: link
Protocol: ISIS L2
Identifier: 1
Local Node Descriptor:
  AS Number: 65002
  BGP Identifier: 50529027
  ISO Node ID: 0001.0001.0001.00
Remote Node Descriptor:
  AS Number: 65002
  BGP Identifier: 50529027
  ISO Node ID: 0002.0002.0002.00
Link Descriptor:
  Local Interface Address IPv4: 10.100.1.1
  Neighbor Interface Address IPv4: 10.100.1.2
Paths: 1 available, best #1
  Last modified: March 03, 2025 16:30:21
  65002
    192.168.123.3 from 192.168.123.3 (3.3.3.3)
      Origin igp, metric 0, localpref 100, weight 0, valid, external, best
      Link-state: Remote TE IPv4 Router-ID: 2.2.2.2, Admin-group: 0x1,
                  Max-link-bw: 10000000000, Max-reservable-bw: 10000000000,
                  Unreserved-bw: [10000000000, 10000000000, 10000000000, 10000000000,
10000000000, 10000000000, 10000000000, 10000000000],
                  TE default metric: 10, IGP metric: 10, ADJ-SID: 24321
```

Node and Link NLRI together are used to build a graph and run the shortest path algorithm.

Prefix NLRI

Prefix NLRI includes a prefix with the associated prefix SID, which is required to build a segment list.

```
TD1#show bgp link-state
[T][L2][I1][N[c65002][b50529027][s0001.0001.0001.00]][P[p1.1.1.1/32]] detail
BGP-LS routing table information
Router identifier 192.168.123.3, local AS number 65001
Prefix codes: E link, V node, T IP reachable route, S SRv6 SID, u/U unknown,
               I Identifier, N local node, R remote node, L link, P prefix, S SID,
               L1/L2 ISIS level-1/level-2, O OSPF, D direct, S static/peer-node,
               a area-ID, l link-ID, t topology-ID, s ISO-ID,
               c confed-ID/ASN, b bgp-identifier, r router-ID, s SID,
               i if-address, n nbr-address, o OSPF Route-type, p IP-prefix,
               d designated router address

BGP routing table entry for
[T][L2][I1][N[c65002][b50529027][s0001.0001.0001.00]][P[p1.1.1.1/32]]
NLRI Type: ipv4_prefix
Protocol: ISIS L2
Identifier: 1
Local Node Descriptor:
  AS Number: 65002
  BGP Identifier: 50529027
  ISO Node ID: 0001.0001.0001.00
Prefix Descriptor:
  Prefix: 1.1.1.1/32
Paths: 1 available, best #1
  Last modified: March 03, 2025 16:30:21
  65002
  192.168.123.3 from 192.168.123.3 (3.3.3.3)
  Origin igp, metric 0, localpref 100, weight 0, valid, external, best
  Link-state: IGP prefix metric: 10, Prefix-SID: 1
```

PCEP session

Now that BGP-LS is up and TD received topology information, let's configure a PCEP session between TD and OcNOS.

Config on TD:

```
router pcep
!
neighbor 192.168.123.3
```

Config on OcNOS:

```
pce configuration 1
  capability
    segment-routing pcep
  pce instantiation
  exit-capability
!
update-source 192.168.123.3
peer-address ipv4 192.168.123.1
```

Verify PCEP on TD:

```
TD1#show pcep summary
PCEP summary information
Neighbor      V      Session ID      SRP ID      MsgRcvd      MsgSent      InQ
OutQ      Up/Down      State
192.168.123.3      1      0/1      1      4      4      0
0      0:00:51      SessionUp
```

Verify PCEP on OcNOS:

```
R3#show pcep peer

=====
Path Computation Client Connection Details :
=====

PCEP entity id : 1
=====
Peer Info
-----
PCE Server IP      : 192.168.123.1
PCEP Source IP    : 192.168.123.3
PCEP Local session-id : 1
PCEP Remote session-id : 0
Session Connect Retry : 0
Session Connect due in : 0 sec
OpenRetry          : 0
Open wait due in   : 0 sec
Keep wait due in   : 0 sec
Keep alive timer due in : 15 sec
Peer Keep-alive value : 30 sec
Peer Dead timer value : 120 sec
Peer Dead timer due in : 106 sec
Peer Overloaded     : No
Peer Overload due in : 0 sec
LocalOK             : 1
RemoteOK            : 1
Max unknown messages : 0
FSM State           : Up
Total FSM State changes : 4
Peer Up time        : 00:01:14
Flap Limit Timer value : 300 Sec
Local Capabilities :
  Stateful PCE Capability : Yes
  LSP Instantiation       : Yes
  SR PCE Capability       : Yes
  SRv6 PCE Capability     : No
Remote Capabilities :
  Stateful PCE Capability : Yes
  LSP Update Capability   : Yes
  LSP Instantiation       : Yes
  SR PCE Capability       : Yes
  SRv6 PCE Capability     : Yes
```

SR-TE policy with explicit path

The most straightforward, although not the most scalable way of configuring SR-TE policies is explicit path. On TD, add the following config:

```
traffic-eng policies
!
  policy R3_R5_via_R1_R2
    headend 3.3.3.3 topology-id 1
    endpoint 5.5.5.5 service-loopback 172.16.101.5
    priority 7 7
    install direct pcep 192.168.123.3
    !
    candidate-path preference 200
      explicit-path via_R1_R2
        bandwidth 2 gbps
    !
!
traffic-eng explicit-paths
!
  explicit-path via_R1_R2
    index 10 strict 10.100.2.1
    index 20 strict 10.100.1.2
```

This instructs TD to calculate a policy that steers traffic from R3 to R5 via R1 and R2, as illustrated by the diagram:

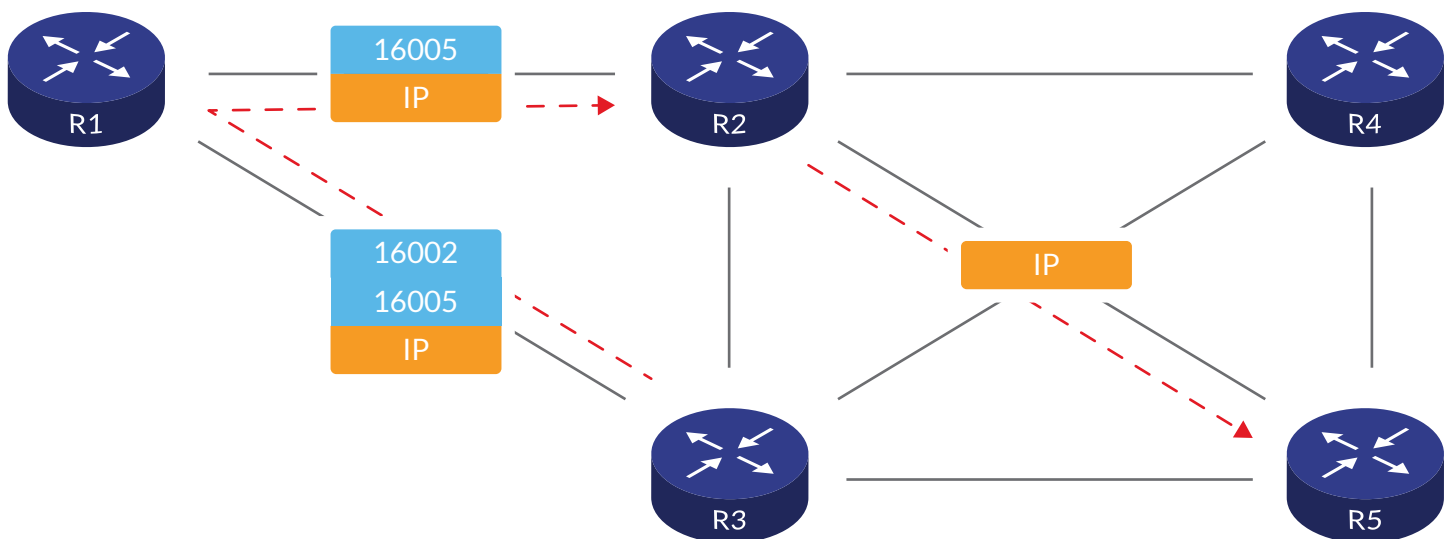


Fig. 7: Traffic steering using explicit path.

Verify calculated policy:

```
TD1#show traffic-eng policy R3_R5_via_R1_R2 detail
Detailed traffic-eng policy information:

Traffic engineering policy "R3_R5_via_R1_R2"

  Valid config, Active
  Headend 3.3.3.3, topology-id 1
  Endpoint 5.5.5.5, service-loopback 172.16.101.5
  Endpoint type: Node, Topology-id: 1, Protocol: isis, Router-id:
0005.0005.0005.00

  Setup priority: 7, Hold priority: 7
  Reserved bandwidth bps: 2000000000
  Install direct, protocol pcep, peer 192.168.123.3
  Policy index: 0, SR-TE distinguisher: 16777216

Candidate paths:
  Candidate-path preference 200
  Path config valid
  Metric: igp
  Path-option: explicit
    Explicit path name: via_R1_R2
  This path is currently active

Calculation results:
  Aggregate metric: 30
  Topologies: ['1']
  Segment lists:
    [16001, 16002, 16005]

Policy statistics:
  Last config update: 2025-06-13 11:57:41,752
  Last recalculation: 2025-06-13 11:58:18.297
  Policy calculation took 0 milliseconds
```

TD calculated label stack [16001, 16002, 16005] - i.e. node SIDs of R1, R2 and R5. Since R1 is directly connected to the headend, node SID of R1 is not sent in dataplane and is used only for nexthop resolution by R3.

PCEP advertisement

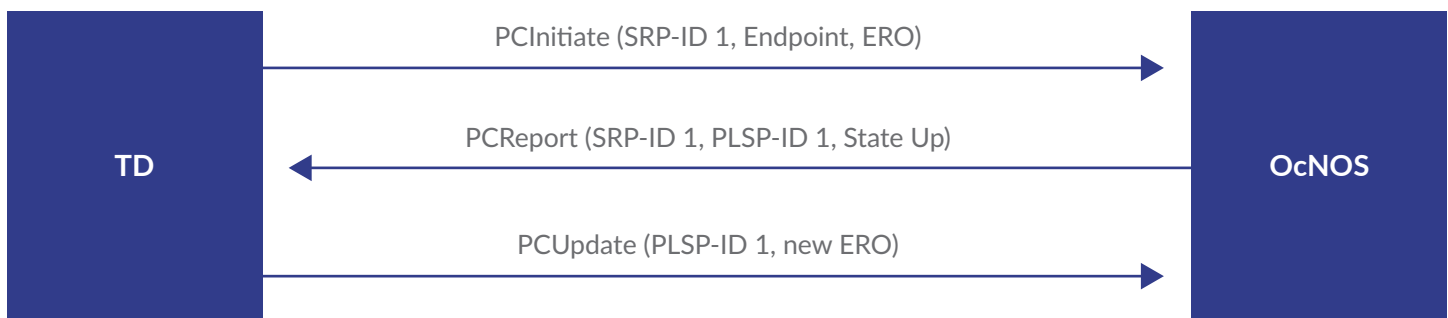
TD calculates the policy towards the endpoint 5.5.5.5 which must be advertised into the IGP, but then it creates a PCEP LSP towards the service loopback 172.16.101.5. The service loopback doesn't need to be advertised into the IGP.

```
TD1#show pcep ipv4 sr-te detail
PCEP SR-TE routing table information

PCEP routing table entry for [16777216][172.16.101.5][5.5.5.5]
Policy name: R3_R5_via_R1_R2
Headend: 3.3.3.3
Endpoint: 172.16.101.5, Color None
Install peer: 192.168.123.3
Last modified: June 13, 2025 11:58:18
Route acked by PCC, PLSP-ID 1
  LSP-ID    Oper status
    0        Up (1)
Metric type igp, metric 30
Segment list: [16001, 16002, 16005]
```

Note that the LSP Oper Status is Up - this means, R3 installed the policy and sent a PCReport to TD, acknowledging the policy. R3 also assigned PLSP-ID 1 to this policy which will be used for all subsequent updates to this policy (e.g. new Explicit Route Object due to changes in topology).

PCEP logic illustrated:



OcNOS received the PCEP route and creates a relevant SR-TE policy:

```
R3#show segment-routing policy detail

Policy-Name: policy1      Color 1      End-point 172.16.101.5      Tunnel-ID: 1
Admin-Status: UP      Oper-Status: UP for 00:01:06
State Transition Count: 1
CSPF Retry Limit: 100      CSPF Retry Interval: 10
Colored Service-FTN Dependent Count: 0
Binding SID :
  BSID: 26240
  Alloc mode: Dynamic
  Oper State: Programmed

CP ID: 0, Active
Preference: 100      Path Type: Explicit      CP Origin: PCE
CP state: Valid
Segment List:
Total no. of segments: 3
  Segment0[LABEL]: Label :16001
  Segment1[LABEL]: Label :16002
  Segment2[LABEL]: Label :16005
Out-if: ce2      Out-label-stack: 3/16002/16005
Computed IGP Metric: 30
Attributes:
```

Note how the first label 16001 is not used in the outgoing label stack. This is because R1 is directly connected to R3, therefore due to PHP, the first label is not used in data plane. TD still sends the entire label stack [16001, 16002, 16005] because the first label is required for nexthop resolution on R3.

Data plane verification

Now let's check the forwarding entry on OcNOS:

```
R3#show mpls ftn-table 172.16.101.5/32
Primary FTN entry with FEC: 172.16.101.5/32, id: 5, row status: Active, Tunnel-
Policy: N/A, State: Installed
  CreateTime: 00:07:08, UpTime: 00:07:08, LastUpdate: N/A
  Owner: SR_POLICY, distance: 0, Action-type: Redirect to Tunnel, Exp-bits: 0x0,
Incoming DSCP: none, Algorithm Number:0
  Tunnel id: 1, Protected LSP id: 0, LSP-type: Primary, QoS Resource id: 0,
Description: policy1, , Color: 1
  Cross connect ix: 15, in intf: - in label: 0 out-segment ix: 18 refcount: 1
  Owner: N/A, Persistent: No, Admin Status: Up, Oper Status: Up
  Out-segment with ix: 18, owner: N/A, Stale: NO, refcount: 1, out intf: ce2, out
label: 3\16002\16005
  Nexthop addr: 10.100.2.1      cross connect ix: 15, op code: Push
```

Configure sniff on R1 and send some traffic:

```
R1(config)#int ce1
R1(config-if)#sniff rx 10 sniff1
R1(config-if)#commit
R1(config-if)#end
R1#start-shell
bash-5.2$ su
Password:
root@R1:/home/ocnos# tcpdump -nvvi sniff1
tcpdump: listening on sniff1, link-type EN10MB (Ethernet), snapshot length 262144 bytes

14:11:45.223276 MPLS (label 16002, tc 5, ttl 255)
                (label 16005, tc 5, [S], ttl 255)
                IP (tos 0x0, ttl 1, id 0, offset 0, flags [none], proto UDP (17), length 80,
options (RA))
                3.3.3.3.3503 > 127.0.0.12.3503: [udp sum ok]
                LSP-PINGv1, msg-type: MPLS Echo Request (1), length: 48
                reply-mode: Reply via an IPv4/IPv6 UDP packet (2)
                Return Code: No return code or return code contained in the Error Code TLV
(0)
                Return Subcode: (0)
                Sender Handle: 0x00000002, Sequence: 1
                Sender Timestamp: 3558.000105250 (1900-01-01T00:59:18Z) Receiver Timestamp:
no timestamp
                Target FEC Stack TLV (1), length: 12
                Unknown subTLV (34), length: 8
                0x0000: ac10 6505 2000 0000
                0x0000: 0022 0008 ac10 6505 2000 0000
```

R3 sends traffic with the correct label stack [16002, 16005].

Bandwidth constraint and reservation

The policy above was also configured with the 2 Gbps bandwidth constraint. TD checks for bandwidth availability when calculating the policy and does bandwidth reservation in the internal database. If the bandwidth is not available, the policy would fail. If there are lower priority policies taking the required bandwidth, the higher priority policy would kick them out.

```
TD1#show topology
-----

ISIS links

0001.0001.0001.00 [E] [L2] [I1] [N[c65002] [b50529027] [s0001.0001.0001.00]]
[R[c65002] [b50529027] [s0002.0002.0002.00]] [L[i10.100.1.1] [n10.100.1.2]]
IGP metric: 10
TE metric: 10
Affinity: 0x1
Max-bw: 100000000000
Unrsv-bw priority 0: 100000000000
Unrsv-bw priority 1: 100000000000
Unrsv-bw priority 2: 100000000000
Unrsv-bw priority 3: 100000000000
Unrsv-bw priority 4: 100000000000
Unrsv-bw priority 5: 100000000000
Unrsv-bw priority 6: 100000000000
Unrsv-bw priority 7: 98000000000
Policies priority 0: []
```

```

Policies priority 1:      []
Policies priority 2:      []
Policies priority 3:      []
Policies priority 4:      []
Policies priority 5:      []
Policies priority 6:      []
Policies priority 7:      ['R3_R5_via_R1_R2']

```

```

-----
[E] [L2] [I1] [N[c65002] [b50529027] [s0002.0002.0002.00]] [R[c65002] [b50529027]
[s0005.0005.0005.00]] [L[i10.100.5.2] [n10.100.5.5]]

```

```

IGP metric:              10
TE metric:               10
Affinity:                0x1
Max-bw:                 100000000000
Unrsv-bw priority 0:    100000000000
Unrsv-bw priority 1:    100000000000
Unrsv-bw priority 2:    100000000000
Unrsv-bw priority 3:    100000000000
Unrsv-bw priority 4:    100000000000
Unrsv-bw priority 5:    100000000000
Unrsv-bw priority 6:    100000000000
Unrsv-bw priority 7:    80000000000
Policies priority 0:     []
Policies priority 1:     []
Policies priority 2:     []
Policies priority 3:     []
Policies priority 4:     []
Policies priority 5:     []
Policies priority 6:     []
Policies priority 7:     ['R3_R5_via_R1_R2']

```

```

-----
0003.0003.0003.00
[E] [L2] [I1] [N[c65002] [b50529027] [s0003.0003.0003.00]] [R[c65002] [b50529027]
[s0001.0001.0001.00]] [L[i10.100.2.3] [n10.100.2.1]]

```

```

IGP metric:              10
TE metric:               10
Affinity:                0x1
Max-bw:                 100000000000
Unrsv-bw priority 0:    100000000000
Unrsv-bw priority 1:    100000000000
Unrsv-bw priority 2:    100000000000
Unrsv-bw priority 3:    100000000000
Unrsv-bw priority 4:    100000000000
Unrsv-bw priority 5:    100000000000
Unrsv-bw priority 6:    100000000000
Unrsv-bw priority 7:    98000000000
Policies priority 0:     []
Policies priority 1:     []
Policies priority 2:     []
Policies priority 3:     []
Policies priority 4:     []
Policies priority 5:     []
Policies priority 6:     []
Policies priority 7:     ['R3_R5_via_R1_R2']

```

SR-TE policy with link affinity

Link affinity (also known as admin group or color) is a more flexible and scalable way of traffic engineering.

In this topology, the links are configured with the respective affinity values (bit position 0 for BLUE and bit position 1 for YELLOW), and this is used as a constraint for SR-TE policies.

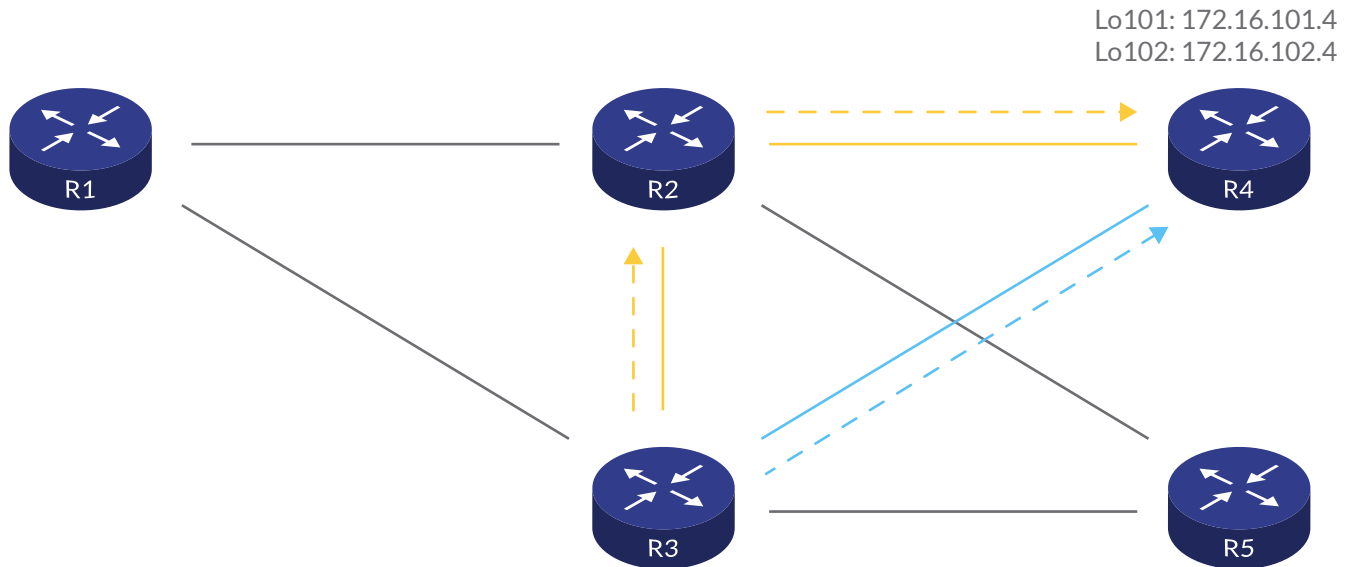


Fig. 8: Traffic engineering with affinity.

Config on OcNOS:

```
admin-group BLUE 0
admin-group YELLOW 1
!
interface cd0
  admin-group BLUE
!
interface cd1
  admin-group YELLOW
```

On TD, we configure affinity-set with the relevant constraint, and then use that set in SR-TE policies:

```
traffic-eng affinities
  affinity-map
    name BLUE bit-position 0
    name YELLOW bit-position 1
  !
  affinity-set BLUE_ONLY
    constraint include-all
    name BLUE
  !
  affinity-set YELLOW_ONLY
    constraint include-all
    name YELLOW
!
traffic-eng policies
!
policy R3_R4_YELLOW
  headend 3.3.3.3 topology-id 1
  endpoint 4.4.4.4 service-loopback 172.16.101.4
  priority 7 7
  install direct pcep 192.168.123.3
  !
  candidate-path preference 200
    affinity-set YELLOW_ONLY
    bandwidth 40 gbps
!
policy R3_R4_BLUE
  headend 3.3.3.3 topology-id 1
  endpoint 4.4.4.4 service-loopback 172.16.102.4
  priority 7 7
  install direct pcep 192.168.123.3
  !
  candidate-path preference 200
    affinity-set BLUE_ONLY
    bandwidth 40 gbps
```

Verify policies:

```
TD1#show traffic-eng policy R3_R4_BLUE detail
Detailed traffic-eng policy information:

Traffic engineering policy "R3_R4_BLUE"

Valid config, Active
Headend 3.3.3.3, topology-id 1
Endpoint 4.4.4.4, service-loopback 172.16.102.4
Endpoint type: Node, Topology-id: 1, Protocol: isis, Router-id:
0004.0004.0004.00

Setup priority: 7, Hold priority: 7
Reserved bandwidth bps: 40000000000
Install direct, protocol pcep, peer 192.168.123.3
Policy index: 2, SR-TE distinguisher: 16777218

Candidate paths:
Candidate-path preference 200
Path config valid
Metric: igp
Path-option: dynamic
Affinity-set: BLUE_ONLY
Constraint: include-all
```

```
      List: ['BLUE']
      Value: 0x1
This path is currently active
```

```
Calculation results:
  Aggregate metric: 10
  Topologies: ['1']
  Segment lists:
    [16004]
```

```
Policy statistics:
  Last config update: 2025-06-13 12:03:40,113
  Last recalculation: 2025-06-13 12:05:11.399
  Policy calculation took 0 milliseconds
```

```
TD1#show traffic-eng policy R3_R4_YELLOW detail
Detailed traffic-eng policy information:
```

```
Traffic engineering policy "R3_R4_YELLOW"
```

```
  Valid config, Active
  Headend 3.3.3.3, topology-id 1
  Endpoint 4.4.4.4, service-loopback 172.16.101.4
  Endpoint type: Node, Topology-id: 1, Protocol: isis, Router-id:
  0004.0004.0004.00
```

```
  Setup priority: 7, Hold priority: 7
  Reserved bandwidth bps: 40000000000
  Install direct, protocol pcep, peer 192.168.123.3
  Policy index: 1, SR-TE distinguisher: 16777217
```

```
  Candidate paths:
    Candidate-path preference 200
    Path config valid
    Metric: igp
    Path-option: dynamic
    Affinity-set: YELLOW_ONLY
    Constraint: include-all
    List: ['YELLOW']
    Value: 0x2
    This path is currently active
```

```
  Calculation results:
    Aggregate metric: 20
    Topologies: ['1']
    Segment lists:
      [16002, 16004]
```

```
  Policy statistics:
    Last config update: 2025-06-13 12:05:11,104
    Last recalculation: 2025-06-13 12:05:11.400
    Policy calculation took 0 milliseconds
```

Verify PCEP routes on TD:

```
TD1#show pcep ipv4 sr-te
PCEP SR-TE routing table information
Status codes: * acked, > up/active, + - inserted, z - zombie

          NLRI                                PLSP-ID    Oper status
*>+      [16777218][172.16.102.4][4.4.4.4]      2         Up (1)
*>+      [16777217][172.16.101.4][4.4.4.4]      3         Up (1)

TD1#show pcep ipv4 sr-te detail
PCEP SR-TE routing table information

PCEP routing table entry for [16777218][172.16.102.4][4.4.4.4]
  Policy name: R3_R4_BLUE
  Headend: 3.3.3.3
  Endpoint: 172.16.102.4, Color None
  Install peer: 192.168.123.3
  Last modified: June 13, 2025 12:05:11
  Route acked by PCC, PLSP-ID 2
    LSP-ID    Oper status
    0         Up (1)
  Metric type igp, metric 10
  Segment list: [16004]
PCEP routing table entry for [16777217][172.16.101.4][4.4.4.4]
  Policy name: R3_R4_YELLOW
  Headend: 3.3.3.3
  Endpoint: 172.16.101.4, Color None
  Install peer: 192.168.123.3
  Last modified: June 13, 2025 12:05:11
  Route acked by PCC, PLSP-ID 3
    LSP-ID    Oper status
    0         Up (1)
  Metric type igp, metric 20
  Segment list: [16002, 16004]
```

Verify SR-TE policies on R3:

```
R3#show segment-routing policy detail
```

Policy-Name: policy2 Color 1 End-point 172.16.102.4 Tunnel-ID: 2

Admin-Status: UP Oper-Status: UP for 00:06:22

State Transition Count: 1

CSPF Retry Limit: 100 CSPF Retry Interval: 10

Colored Service-FTN Dependent Count: 0

Binding SID :

BSID: 26241

Alloc mode: Dynamic

Oper State: Programmed

CP ID: 0, Active

Preference: 100 Path Type: Explicit CP Origin: PCE

CP state: Valid

Segment List:

Total no. of segments: 1

Segment0[LABEL]: Label :16004

Out-if: cd0 Out-label-stack: 3

Computed IGP Metric: 10

Attributes:

Policy-Name: policy3 Color 1 End-point 172.16.101.4 Tunnel-ID: 3

Admin-Status: UP Oper-Status: UP for 00:04:51

State Transition Count: 1

CSPF Retry Limit: 100 CSPF Retry Interval: 10

Colored Service-FTN Dependent Count: 0

Binding SID :

BSID: 26242

Alloc mode: Dynamic

Oper State: Programmed

CP ID: 0, Active

Preference: 100 Path Type: Explicit CP Origin: PCE

CP state: Valid

Segment List:

Total no. of segments: 2

Segment0[LABEL]: Label :16002

Segment1[LABEL]: Label :16004

Out-if: cd1 Out-label-stack: 3/16004

Computed IGP Metric: 20

Attributes:

Mapping traffic to SR-TE policies

The 2 policies above are configured with different service loopbacks. Their purpose is to map services to the respective SR-TE policies. In order to do this, we have to:

- 1) Configure these service loopbacks on R4
- 2) Change BGP nexthop for the service routes we want to map to these policies
- 3) Advertise service loopbacks in BGP-LU from R4 to R3

The last point is required so that in case the controller fails, traffic will be rerouted via the shortest path and won't be blackholed. An alternative to configuring BGP-LU would be advertising service loopbacks into the IGP, but that would require a prefix SID advertisement with each service loopback. BGP-LU is a simpler and more scalable option.

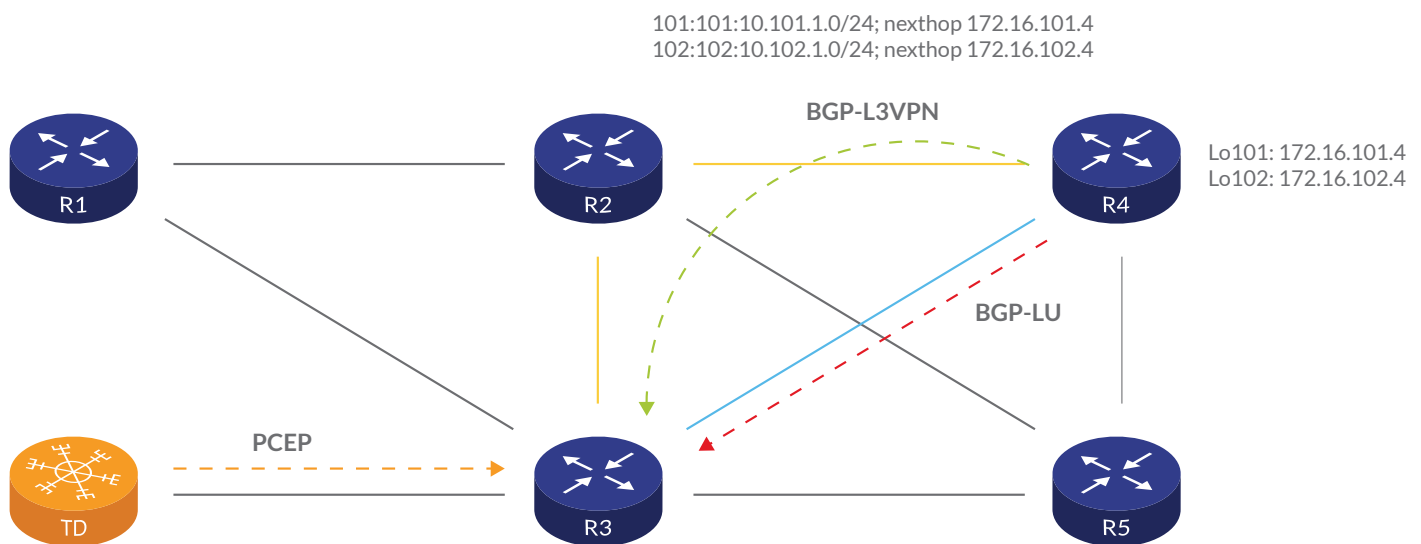


Fig. 9: Mapping MPLS VPN routes to service loopbacks.

R3 MPLS VPN config:

```
hostname R3
!
ip vrf vrf101
  rd 101:101
  route-target both 101:101
!
ip vrf vrf102
  rd 102:102
  route-target both 102:102
!
interface xe12
  ip vrf forwarding vrf101
  ip address 10.101.0.3/24
!
interface xe13
```

```

ip vrf forwarding vrf102
ip address 10.102.0.3/24
!
router bgp 65002
  bgp router-id 3.3.3.3
  neighbor 4.4.4.4 remote-as 65002
  !
  address-family ipv4 labeled-unicast
    neighbor 4.4.4.4 activate
  exit-address-family
  !
  address-family vpnv4 unicast
    neighbor 4.4.4.4 activate
  exit-address-family
  !
  address-family ipv4 vrf vrf101
    redistribute connected
  exit-address-family
  !
  address-family ipv4 vrf vrf102
    redistribute connected
  exit-address-family

```

R4 MPLS VPN config:

```

hostname R4
!
ip vrf vrf101
  rd 101:101
  route-target both 101:101
!
ip vrf vrf102
  rd 102:102
  route-target both 102:102
!
ip prefix-list VRF101
  seq 5 permit 10.101.1.0/24
!
ip prefix-list VRF102
  seq 5 permit 10.102.1.0/24
!
route-map SET_VPN_NH permit 10
  match ip address prefix-list VRF101
  set vpnv4 next-hop 172.16.101.4
!
route-map SET_VPN_NH permit 20
  match ip address prefix-list VRF102
  set vpnv4 next-hop 172.16.102.4
!
interface loopback101
  ip address 172.16.101.4/32 secondary
!
interface loopback102
  ip address 172.16.102.4/32 secondary
!
interface xe7
  ip vrf forwarding vrf101
  ip address 10.101.1.4/24
!
interface xe8
  ip vrf forwarding vrf102
  ip address 10.102.1.4/24
!

```

```

router bgp 65002
  bgp router-id 4.4.4.4
  allocate-label all
  neighbor 3.3.3.3 remote-as 65002
  !
  address-family ipv4 unicast
  network 172.16.101.4/32
  network 172.16.102.4/32
  exit-address-family
  !
  address-family ipv4 labeled-unicast
  neighbor 3.3.3.3 activate
  exit-address-family
  !
  address-family vpnv4 unicast
  neighbor 3.3.3.3 activate
  neighbor 3.3.3.3 route-map SET_VPN_NH out
  exit-address-family
  !
  address-family ipv4 vrf vrf101
  exit-address-family
  !
  address-family ipv4 vrf vrf102
  exit-address-family

```

Now let's see how R3 resolves MPLS VPN routes via the SR-TE policy:

```
R3#show ip bgp vpnv4 all neighbors 4.4.4.4 routes
```

```

For address family: VPNv4 Unicast
Status codes: s suppressed, d damped, h history, a add-path, b back-up, * valid,
> best, i - internal, l - labeled
              S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

```

Network	Next Hop	Metric	LocPrf	Weight	Path
Route Distinguisher: 101:101					
*>i 10.101.1.0/24	172.16.101.4	0	100	0	? -
Accepted routes count = 1					
Route Distinguisher: 102:102					
*>i 10.102.1.0/24	172.16.102.4	0	100	0	? -
Accepted routes count = 1					

```
R3#show mpls vrf-forwarding-table
```

```

Codes: > - installed FTN, * - selected FTN, p - stale FTN, ! - using backup, B -
BGP FTN
(m) - Service mapped over multipath transport
(e) - Service mapped over LDP ECMP or SR ECMP
Ext-Color - Extended-community color advertised by BGP

```

Code	FEC	FTN-ID	VRF-ID	Nhlfe-ID	Pri	Out-Label	Out-
Intf	Nexthop	UpTime	Ext-Color				
B> 10.101.1.0/24		1	3	29	Yes	24960	-
172.16.101.4	00:10:09	-					
B> 10.102.1.0/24		1	4	30	Yes	24961	-
172.16.102.4	00:11:40	-					

Ping from R3 to R4 in VRF101:

```
R3#ping 10.101.1.4 vrf vrf101
Press CTRL+C to exit
PING 10.101.1.4 (10.101.1.4) 100(128) bytes of data.
108 bytes from 10.101.1.4: icmp_seq=1 ttl=64 time=0.552 ms
108 bytes from 10.101.1.4: icmp_seq=2 ttl=64 time=0.507 ms
```

Tcpdump on R2 cd1 interface shows that traffic is indeed forwarded via the yellow path with appropriate transport and VPN labels:

```
root@R2:/home/ocnos# tcpdump -nvvi sniff1
tcpdump: listening on sniff1, link-type EN10MB (Ethernet), snapshot length 262144 bytes
14:17:59.409837 MPLS (label 16004, tc 0, ttl 64)
    (label 24960, tc 0, [S], ttl 64)
    IP (tos 0x0, ttl 64, id 61130, offset 0, flags [DF], proto ICMP (1), length 128)
    10.101.0.3 > 10.101.1.4: ICMP echo request, id 3975, seq 1, length 108
```

Redundancy and failure scenarios

Since centralized traffic engineering relies on a controller, the natural question is what happens if the controller fails.

A well-designed network should react to failures gracefully and retain some level of connectivity whenever it's physically possible. A common problem with many SDN technologies is that controller failure leads to a network outage, which is unacceptable.

Dual PCE redundancy

One obvious solution is deploying multiple controllers. PCEP handles this through the LSP delegation mechanism. Initially, each LSP is delegated to the PCE (controller) that created it. If that PCE fails, the PCC (router) redelegates the LSP to another PCE. See figure 10:

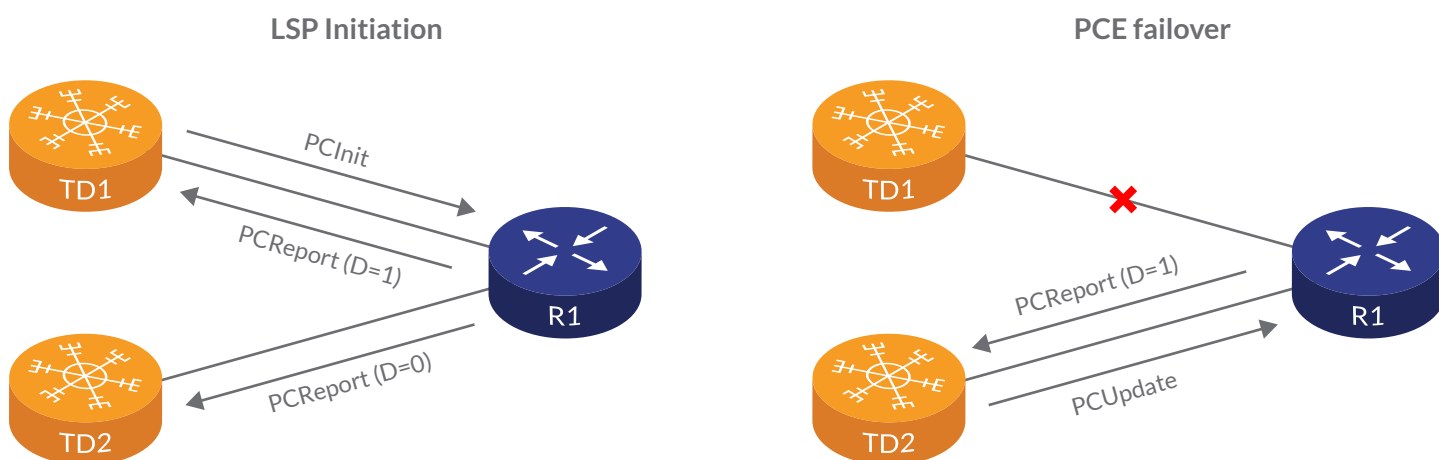


Fig. 10: PCEP delegation and failover.

Output on OcNOS before the primary PCE went down:

```
R3#show pcep segment-routing lsp brief
```

PCEP Entity : 1 Database

Policy	CP	PLSP-ID	Oper	Admin	Report	Delegated	PCE
Local-Policy	Redelegation	State-timeout	Status	Status	Enabled	PCE	Initiated
Name	ID	per lsp					
Status	in secs	in secs					
policy2	0	2	UP	UP	Yes	192.168.123.1	Yes
No	1	31					
policy1	0	1	UP	UP	Yes	192.168.123.1	Yes
No	1	31					
policy3	0	3	UP	UP	Yes	192.168.123.1	Yes
No	1	31					

Redelegation after failover:

```
R3#show pcep segment-routing lsp brief
```

PCEP Entity : 1 Database

Policy	CP	PLSP-ID	Oper	Admin	Report	Delegated	PCE
Local-Policy	Redelegation	State-timeout					
Name	ID	per lsp	Status	Status	Enabled	PCE	Initiated
Status	in secs	in secs					

policy2		0	2	UP	UP	Yes	192.168.123.2 Yes
No	NA		NA				
policy1		0	1	UP	UP	Yes	192.168.123.2 Yes
No	NA		NA				
policy3		0	3	UP	UP	Yes	192.168.123.2 Yes
No	NA		NA				

It is possible to deploy more than 2 controllers this way. The PCEP delegation and failover mechanism is PCC-centric and does not require any state synchronization between different PCE, therefore it can work with any number of PCE.

Complete PCE failure

If all PCE fail at the same time, all SR-TE LSP are going to disappear. In this case, the MPLS VPN routes are going to be resolved via the shortest path route.

Before failure:

```
R3#show mpls forwarding-table
```

Codes: > - installed FTN, * - selected FTN, p - stale FTN, ! - using backup
B - BGP FTN, K - CLI FTN, (t) - tunnel, P - SR Policy FTN, (b) - bypass,
L - LDP FTN, R - RSVP-TE FTN, S - SNMP FTN, I - IGP-Shortcut,
U - unknown FTN, O - SR-OSPF FTN, i - SR-ISIS FTN, k - SR-CLI FTN
(m) - FTN mapped over multipath transport, (e) - FTN is ECMP

FTN-ECMP LDP: Disabled, SR: Disabled

Code	FEC	FTN-ID	Nhlfe-ID	Tunnel-ID	Pri	Out-Label
Out-Intf	ELC	Nexthop	Algo-Num	UpTime		
P>	172.16.101.4/32	8	34	3	Yes	3
cd1	No	10.100.3.2	N/A	00:14:21		
P>	172.16.102.4/32	6	32	2	Yes	3
cd0	No	10.100.5.4	N/A	00:15:52		

After failure:

```
R3#show mpls forwarding-table
Codes: > - installed FTN, * - selected FTN, p - stale FTN, ! - using backup
      B - BGP FTN, K - CLI FTN, (t) - tunnel, P - SR Policy FTN, (b) - bypass,
      L - LDP FTN, R - RSVP-TE FTN, S - SNMP FTN, I - IGP-Shortcut,
      U - unknown FTN, O - SR-OSPF FTN, i - SR-ISIS FTN, k - SR-CLI FTN
      (m) - FTN mapped over multipath transport, (e) - FTN is ECMP

FTN-ECMP LDP: Disabled, SR: Disabled
Code   FEC          FTN-ID   Nhlfe-ID  Tunnel-ID  Pri   Out-Label
Out-Intf  ELC      Nexthop   Algo-Num   UpTime
cd0      B> 172.16.101.4/32  5         43         -        Yes   24962
          No      4.4.4.4      N/A        00:00:16
cd0      B> 172.16.102.4/32  6         44         -        Yes   24963
          No      4.4.4.4      N/A        00:00:16
```

The nexthop for the MPLS VPN routes doesn't change, but now it's resolved via IS-IS SR (4.4.4.4).

Path Diversity

In the previous example we used link affinities to steer 2 policies between the same 2 routers via different paths. This solves the problem but might not be the most optimal solution if the requirement is just to make sure that policies don't use the same links.

The simpler solution is just to configure multiple policies with the same disjoint group on TD. Config example:

```
traffic-eng policies
!
policy R3_R4_PATH1
  headend 3.3.3.3 topology-id 1
  endpoint 4.4.4.4 service-loopback 172.16.102.4
  priority 7 7
  install direct pcep 192.168.123.3
  !
  candidate-path preference 200
  disjoint-group 100 link
!
policy R3_R4_PATH2
  headend 3.3.3.3 topology-id 1
  endpoint 4.4.4.4 service-loopback 172.16.101.4
  priority 7 7
  install direct pcep 192.168.123.3
  !
  candidate-path preference 200
  disjoint-group 100 link
```

Verify policy status

```
TD1#show traffic-eng policy R3_R4_PATH1 detail
Detailed traffic-eng policy information:

Traffic engineering policy "R3_R4_PATH1"

  Valid config, Active
  Headend 3.3.3.3, topology-id 1
  Endpoint 4.4.4.4, service-loopback 172.16.102.4
  Endpoint type: Node, Topology-id: 1, Protocol: isis, Router-id:
0004.0004.0004.00

  Setup priority: 7, Hold priority: 7
  Reserved bandwidth bps: 0
  Install direct, protocol pcep, peer 192.168.123.3
  Policy index: 4, SR-TE distinguisher: 16777220

  Candidate paths:
    Candidate-path preference 200
    Path config valid
    Metric: igp
    Disjoint-group: 100
    Path-option: dynamic
    This path is currently active

  Calculation results:
    Aggregate metric: 10
    Topologies: ['1']
    Segment lists:
      [16004]

  Policy statistics:
    Last config update: 2025-06-13 12:43:41,698
    Last recalculation: 2025-06-13 12:43:42.007
    Policy calculation took 0 milliseconds

TD1#show traffic-eng policy R3_R4_PATH2 detail
Detailed traffic-eng policy information:

Traffic engineering policy "R3_R4_PATH2"

  Valid config, Active
  Headend 3.3.3.3, topology-id 1
  Endpoint 4.4.4.4, service-loopback 172.16.101.4
  Endpoint type: Node, Topology-id: 1, Protocol: isis, Router-id:
0004.0004.0004.00

  Setup priority: 7, Hold priority: 7
  Reserved bandwidth bps: 0
  Install direct, protocol pcep, peer 192.168.123.3
  Policy index: 5, SR-TE distinguisher: 16777221

  Candidate paths:
    Candidate-path preference 200
    Path config valid
    Metric: igp
    Disjoint-group: 100
    Path-option: dynamic
    This path is currently active

  Calculation results:
    Aggregate metric: 20
    Topologies: ['1']
    Segment lists:
```

[16002, 16004]

Policy statistics:

Last config update: 2025-06-13 12:43:41,700

Last recalculation: 2025-06-13 12:43:42.007

Policy calculation took 0 milliseconds

Disjoint groups ensure that these policies will always use different paths. There is no need for any extra configuration of affinities, explicit paths etc.

ECMP and Anycast SID

A significant advantage of Segment Routing over older MPLS-TE is full support for Equal Cost Multi-Path (ECMP). SR does ECMP load balancing whenever multiple routes with equal cost are available.

Note: On OcNOS, the following config is required to enable ECMP in hardware. Command: “mpls ftn-ecmp sr” on source node and “mpls ilm-ecmp sr” on transit node.

However, in some topologies, an SR-TE policy with ECMP might require multiple segment lists.

What if we want to steer traffic from R1 to R3, balancing it over the 2 paths:

- 1) R1-R2-R4-R3
- 2) R1-R2-R5-R3

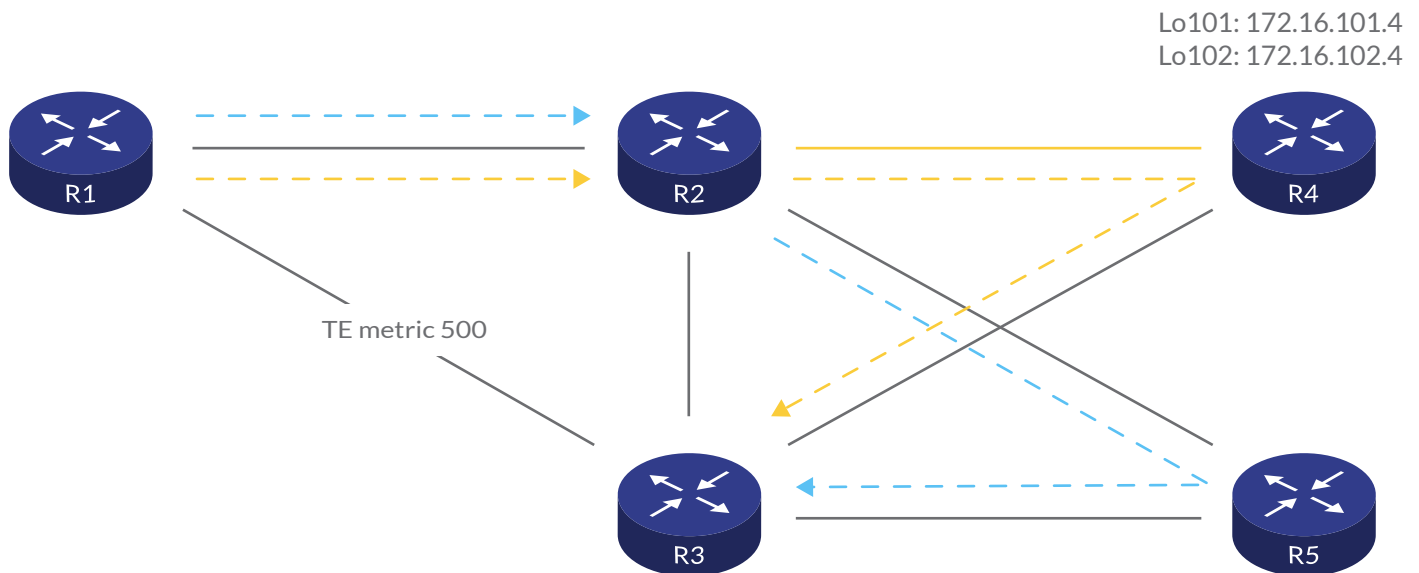


Fig. 11: ECMP load balancing.

Such a constraint would require TD to generate 2 segment lists: [16002, 16004, 16003] and [16002, 16005, 16003].

Advertising multiple segment lists is suboptimal because they will consume more hardware resources.

Segment Routing allows us to configure anycast SID: just a prefix SID with the same value, configured on multiple routers.

Config on R4 and R5:

```
interface loopback45
ip address 172.20.45.45/32
prefix-sid index 45 n-flag-clear
ip router isis 1
```

The “**n-flag-clear**” instructs R4 and R5 to advertise this SID as a regular prefix SID (not a node SID). Node SID must be unique in the IGP domain, but a regular prefix SID can be advertised by multiple routers.

SR-TE policy config on TD:

```
traffic-eng policies
!
policy R1_R3_ANYCAST
  headend 1.1.1.1 topology-id 1
  endpoint 3.3.3.3 service-loopback 172.16.101.3
  priority 7 7
  install direct pcep 192.168.123.1
  !
  candidate-path preference 200
    explicit-path R45_ANYCAST
    metric te
!
traffic-eng explicit-paths
!
explicit-path R45_ANYCAST
index 10 loose 172.20.45.45
```

In this example, we use explicit path, with anycast SID as a loose nexthop, but TD can use anycast SID with any type of constraint (e.g. affinity, without explicit path).

When an SR-TE policy uses ECMP, TD will try to use anycast SID whenever possible, in order to combine multiple segment lists into one. It might not be possible to use anycast SID due to some reasons like different SRGB on routers, or if an adjacency SID is required after suggested anycast SID.

TD output:

```
TD1#show traffic-eng policy R1_R3_ANYCAST detail
Detailed traffic-eng policy information:

Traffic engineering policy "R1_R3_ANYCAST"

Valid config, Active
Headend 1.1.1.1, topology-id 1
Endpoint 3.3.3.3, service-loopback 172.16.101.3
  Endpoint type: Node, Topology-id: 1, Protocol: isis, Router-id:
0003.0003.0003.00

Setup priority: 7, Hold priority: 7
Reserved bandwidth bps: 0
Install direct, protocol pcep, peer 192.168.123.3
Policy index: 3, SR-TE distinguisher: 16777219

Candidate paths:
```

```
Candidate-path preference 200
Path config valid
Metric: te
Path-option: explicit
  Explicit path name: R45_ANYCAST
This path is currently active
```

```
Calculation results:
Aggregate metric: 30
Topologies: ['1']
```

```
Segment lists:
[16002, 16045, 16003]
```

```
Policy statistics:
Last config update: 2025-06-13 12:46:52,493
Last recalculation: 2025-06-13 12:46:52.721
Policy calculation took 0 milliseconds
```

In real network design, a common use case for anycast SID is multi-domain traffic engineering, where an SR-TE policy must go through multiple IGP topologies, like on figure 12:

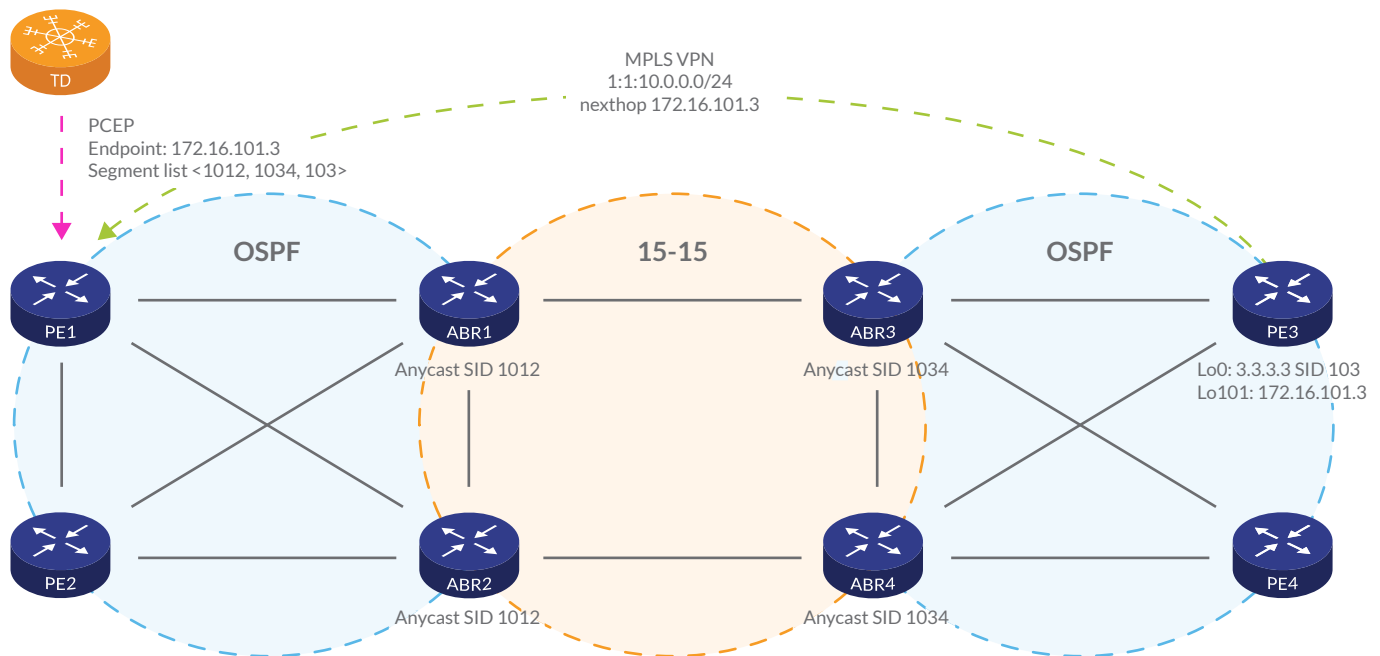


Fig. 12: Anycast SID in multi-domain traffic engineering.

Egress Peer Engineering

Egress peer engineering is a traffic engineering technique that steers traffic to a specific egress peer of a specific node, overriding the normal BGP best path selection mechanism. This can be useful in situations where extra flexibility is required to balance outgoing traffic over available exits.

Consider the topology:

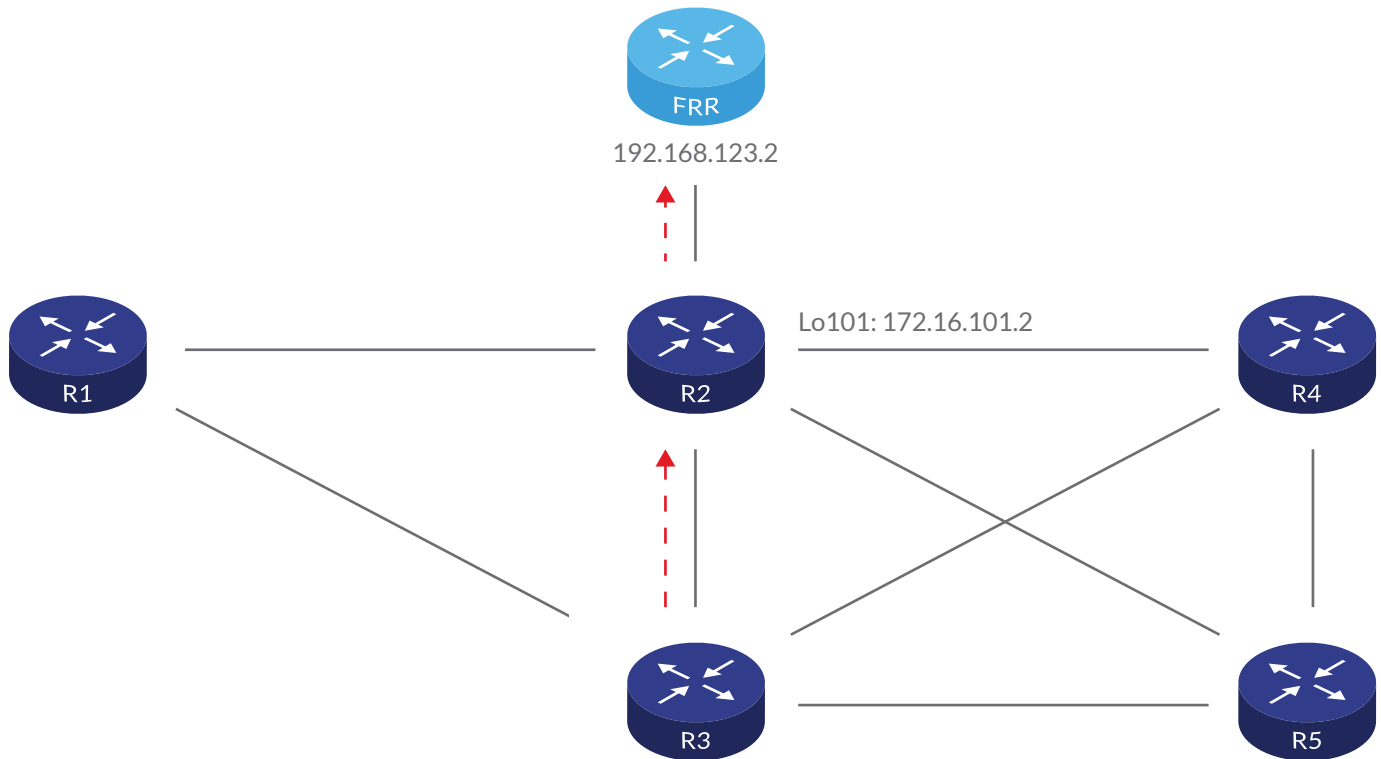


Fig. 13: Egress Peer Engineering example.

An FRR instance is used to simulate the external peer. Config added to R2:

```
router bgp 65002
  bgp router-id 2.2.2.2
  neighbor 3.3.3.3 remote-as 65002
  neighbor 3.3.3.3 update-source 2.2.2.2
  neighbor 192.168.123.2 remote-as 101
  !
  address-family ipv4 unicast
  neighbor 192.168.123.2 activate
  exit-address-family
  !
  address-family link-state link-state
  neighbor 3.3.3.3 activate
  exit-address-family
  egress-engineering
  neighbor 192.168.123.2 peer-node
  exit-egress-engineering
```

With this config, R2 creates a BGP Peer SID for the egress peer:

```
R2#show bgp link-state link-state | in B
BGP router identifier 2.2.2.2, local AS number 65002
[E][B][I0x0][N[c65002][b2.2.2.2][b2.2.2.2][c65002]][R[c65002][b2.2.2.2]
[b192.168.123.2][c101]][L[i192.168.123.102][n192.168.123.2]]/344
```

We can either set up a BGP-LS session between TD and R2, or between R2 and R3, so that R3 advertises all BGP-LS data over an already existing session with TD.

The same route received on TD:

```
TD1#show bgp link-state [E][B][I0][N[c65002][b0][q2.2.2.2]][R[c101][b0]
[q192.168.123.2]][L[i192.168.123.102][n192.168.123.2]] detail
BGP-LS routing table information
Router identifier 192.168.123.3, local AS number 65001
Prefix codes: E link, V node, T IP reachable route, S SRv6 SID, u/U unknown,
               I Identifier, N local node, R remote node, L link, P prefix, S SID,
               L1/L2 ISIS level-1/level-2, O OSPF, D direct, S static/peer-node,
               a area-ID, l link-ID, t topology-ID, s ISO-ID,
               c confed-ID/ASN, b bgp-identifier, r router-ID, s SID,
               i if-address, n nbr-address, o OSPF Route-type, p IP-prefix,
               d designated router address

BGP routing table entry for [E][B][I0][N[c65002][b0][q2.2.2.2]][R[c101][b0]
[q192.168.123.2]][L[i192.168.123.102][n192.168.123.2]]
NLRI Type: link
Protocol: BGP
Identifier: 0
Local Node Descriptor:
  AS Number: 65002
  BGP Identifier: 0.0.0.0
  BGP Router Identifier: 2.2.2.2
Remote Node Descriptor:
  AS Number: 101
  BGP Identifier: 0.0.0.0
  BGP Router Identifier: 192.168.123.2
Link Descriptor:
  Local Interface Address IPv4: 192.168.123.102
  Neighbor Interface Address IPv4: 192.168.123.2
Paths: 1 available, best #1
  Last modified: March 09, 2025 17:03:16
  65002
  192.168.123.3 from 192.168.123.3 (3.3.3.3)
  Origin igp, metric 0, localpref 100, weight 0, valid, external, best
  Link-state: Peer-SID: 25600
```

Now to create an SR-EPE policy, we just have to specify egress peer IP as an endpoint:

```
traffic-eng policies
!
policy R3_R2_EPE
  headend 3.3.3.3 topology-id 1
  endpoint 192.168.123.2 service-loopback 172.16.101.2
  priority 7 7
  install direct pcep 192.168.123.3
  !
  candidate-path preference 200
```

TD will resolve 192.168.123.2 as a BGP-LS route with Peer SID, map BGP router ID to the IS-IS router-ID of R2 and build an SR-EPE policy:

```
TD1#show traffic-eng policy R3_R2_EPE detail
Detailed traffic-eng policy information:

Traffic engineering policy "R3_R2_EPE"

Valid config, Active
Headend 3.3.3.3, topology-id 1
Endpoint 192.168.123.2, service-loopback 172.16.101.2
Endpoint type: Egress peer, Topology-id: 1, Protocol: isis, Router-id:
0002.0002.0002.00

Setup priority: 7, Hold priority: 7
Reserved bandwidth bps: 0
Install direct, protocol pcep, peer 192.168.123.3
Policy index: 5, SR-TE distinguisher: 16777221

Candidate paths:
  Candidate-path preference 200
    Path config valid
    Metric: igp
    Path-option: dynamic
    This path is currently active

Calculation results:
  Aggregate metric: 10
  Topologies: ['1']
  Segment lists:
    [16002, 25600]

Policy statistics:
  Last config update: 2025-03-09 17:19:40.821
  Last recalculation: 2025-03-09 17:19:41.125
  Policy calculation took 0 milliseconds
```

Similarly to regular SR-TE policies, the SR-EPE policy is advertised to R3 using PCEP, and traffic can be mapped to it using service loopbacks as described above in this document.

Affinity and bandwidth constraints in EPE policies

IS-IS and OSPF support link attributes such as bandwidth and affinity, which can be used for traffic engineering as shown above in this document.

BGP Peer SID does not have similar attributes, however on TD it is possible to configure affinities and bandwidth for egress peers.

TD config:

```
traffic-eng nodes
!
node 2.2.2.2
!
neighbor 192.168.123.2
  affinity BLUE
  bandwidth 10 gbps
```

This configuration assigns affinity and bandwidth to the egress peer of R2. When trying to build an EPE policy with affinity or bandwidth constraints, TD will correlate this static config with dynamically advertised BGP Peer SID. If the Peer SID is withdrawn (e.g. because the neighbor goes down), the EPE policy will fail.

Let's configure another SR-EPE policy, steering traffic via only BLUE links:

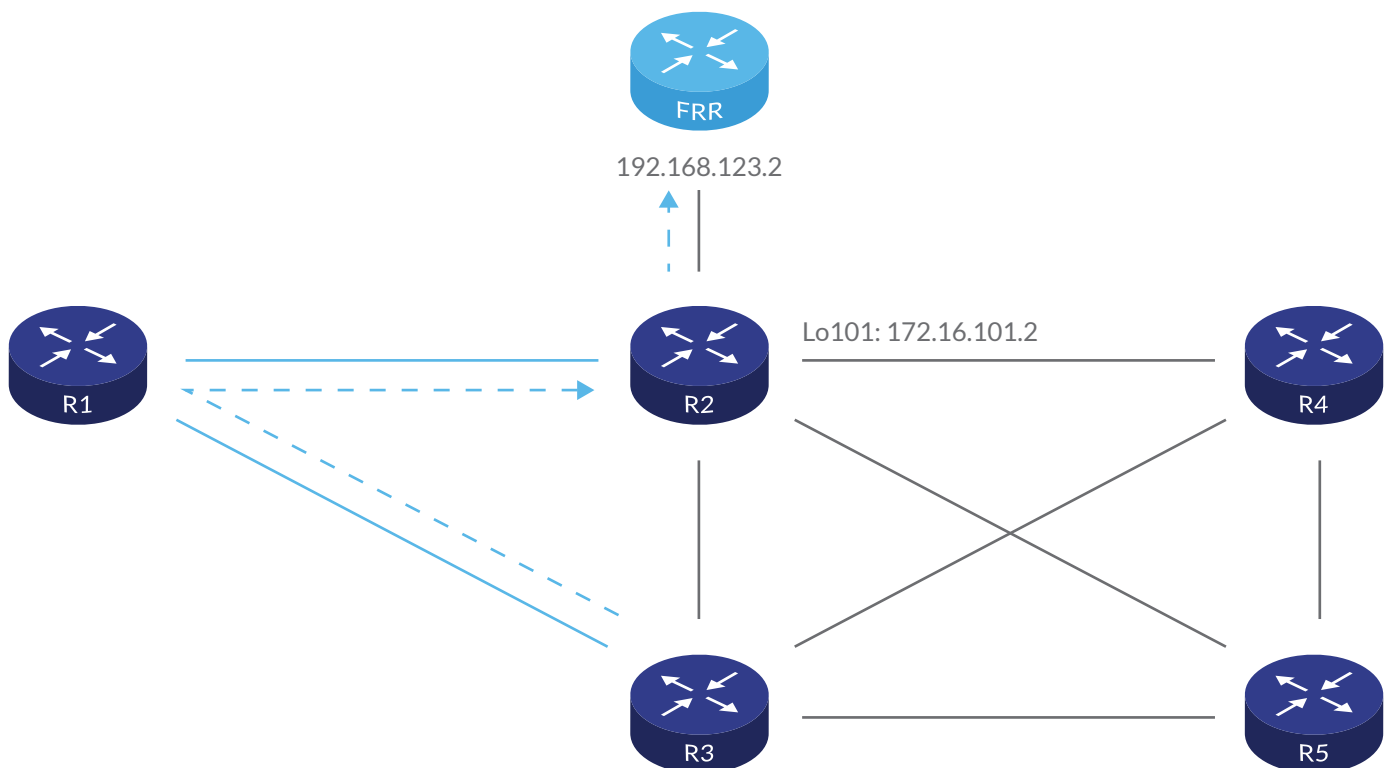


Fig. 14: Egress Peer Engineering with affinity.

```

traffic-eng policies
!
policy R3_R2_EPE_BLUE
  headend 3.3.3.3 topology-id 1
  endpoint 192.168.123.2 service-loopback 172.16.102.2
  priority 7 7
  install direct pcep 192.168.123.3
  !
  candidate-path preference 200
  affinity-set BLUE_ONLY

```

Result:

```

TD1#show traffic-eng policy R3_R2_EPE_BLUE detail
Detailed traffic-eng policy information:

Traffic engineering policy "R3_R2_EPE_BLUE"

  Valid config, Active
  Headend 3.3.3.3, topology-id 1
  Endpoint 192.168.123.2, service-loopback 172.16.102.2
  Endpoint type: Egress peer, Topology-id: 1, Protocol: isis, Router-id:
0002.0002.0002.00

  Setup priority: 7, Hold priority: 7
  Reserved bandwidth bps: 0
  Install direct, protocol pcep, peer 192.168.123.3
  Policy index: 6, SR-TE distinguisher: 16777222

  Candidate paths:
    Candidate-path preference 200
    Path config valid
    Metric: igp
    Path-option: dynamic
    Affinity-set: BLUE_ONLY
    Constraint: include-all
    List: ['BLUE']
    Value: 0x1
    This path is currently active

  Calculation results:
    Aggregate metric: 20
    Topologies: ['1']
    Segment lists:
      [16001, 16002, 25600]

  Policy statistics:
    Last config update: 2025-03-09 17:47:50,298
    Last recalculation: 2025-03-09 17:52:05.606
    Policy calculation took 0 milliseconds

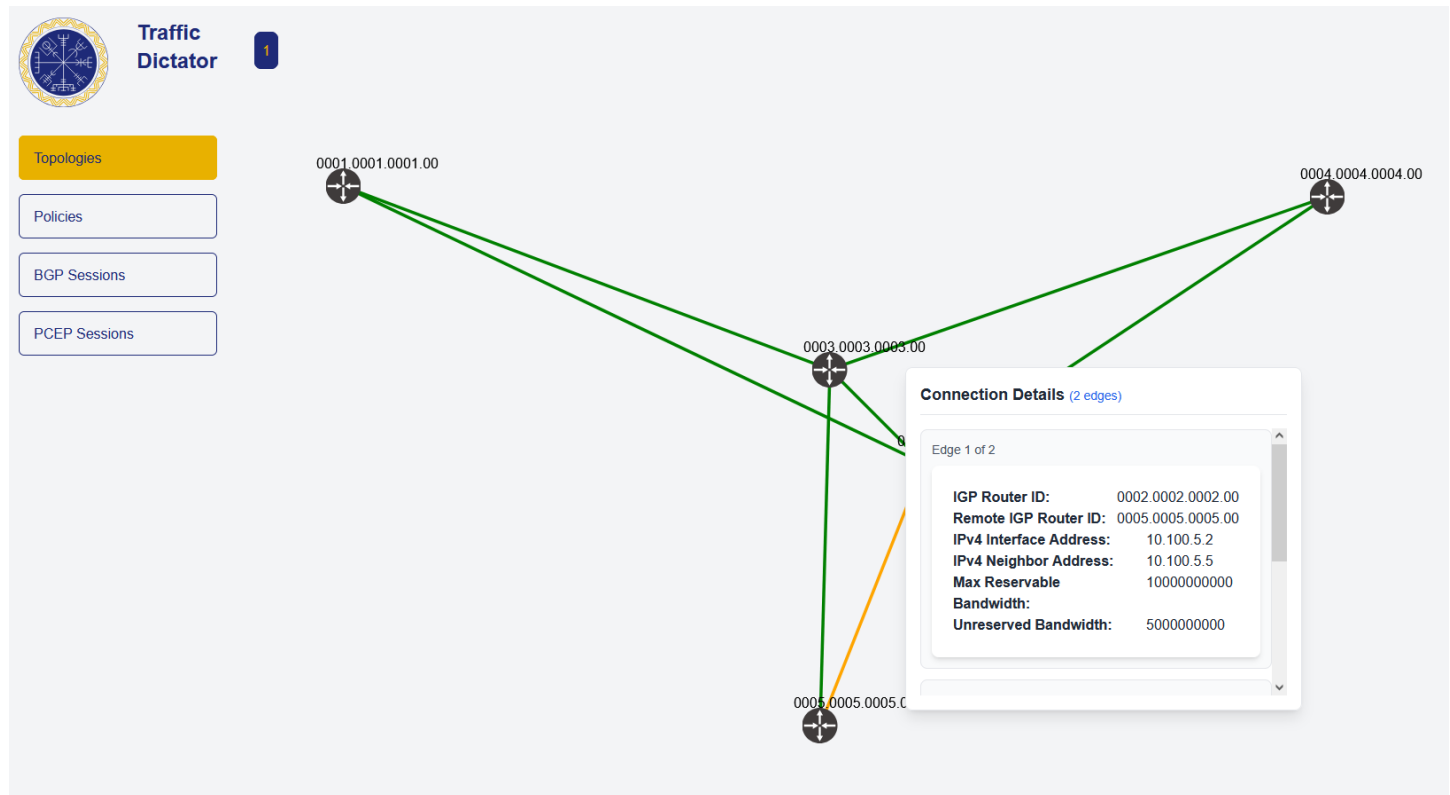
```

Similarly, it is possible to use the bandwidth constraint for EPE policies, to ensure optimal rerouting after some egress peers fail.

TD Graphical Interface

In addition for CLI and HTTP API, Traffic Dictator provides a GUI which allows the operator to see the network topology drawn from BGP-LS data, and create/edit/view SR-TE policies.

Topology view:



Note that the link between R2 and R5 is orange - which reflects the current bandwidth reservation status. Other links are green which means >75% bandwidth is available.

Policy view:



Traffic Dictator

Topologies

Policies

BGP Sessions

PCEP Sessions

POLICY NAME	HEADEND	ENDPOINT	COLOR/SERVICE LOOPBACK	PROTOCOL	RESERVED BANDWIDTH	PRIORITY	STATUS/REASON	ACTIONS
R1_R3_ANYCAST	1.1.1.1	3.3.3.3	172.16.101.3	PCEP/direct	N/A	7/7	Active	  
R3_R4_PATH1	3.3.3.3	4.4.4.4	172.16.102.4	PCEP/direct	N/A	7/7	Active	  
R3_R4_PATH2	3.3.3.3	4.4.4.4	172.16.101.4	PCEP/direct	N/A	7/7	Active	  
R3_R5_via_R1_R2	3.3.3.3	5.5.5.5	172.16.101.5	PCEP/direct	5000000000	7/7	Active	  

Create New Policies

General routing config

SPF delay

200

SPF holddown

5000

SPF rapid runs

3

Global SR-TE distinguisher

1

Reoptimization interval

3600

Traffic engineering statistic

Total number of policies

4

Policies with valid config

4

Policies admin down

N/A

Active policies

4

Policies recalculated

35

SPF caches hit count

18

Last reoptimization

June 13, 2025 12:34:03

Reoptimization count

1

Last recalculation statistic

Recalculation completed at

2025-06-13 13:14:57.380

Number of updated policies

4

Recalculation took

1

Edit policy view:



Traffic Dictator

Topologies

Policies

BGP Sessions

PCEP Sessions

POLICY NAME	HEADEND	ENDPOINT	COLOR/SERVICE LOOPBACK	PROTOCOL	RESERVED BANDWIDTH	PRIORITY	STATUS/REASON	ACTIONS
R1_R3_ANYCAST	1.1.1.1	3.3.3.3	172.16.101.3	PCEP/direct	N/A	7/7	Active	  
R3_R4_PATH1	3.3.3.3	4.4.4.4	172.16.102.4	PCEP/direct	N/A	7/7	Active	  
R3_R4_PATH2	3.3.3.3	4.4.4.4	172.16.101.4	PCEP/direct	N/A	7/7	Active	  
R3_R5_via_R1_R2	3.3.3.3	5.5.5.5	172.16.101.5	PCEP/direct	5000000000	7/7	Active	  

Edit Policy R3_R5_via_R1_R2

*Policy name

R3_R5_via_R1_R2

Description

*Headend

3.3.3.3

*Topology-id

1

*Endpoint

5.5.5.5

Color/Service-loopback

☐ Color

☒ Service loopback

Service loopback

172.16.101.5

Binding SID

ENLP

☐ IPv4

☐ IPv6

☐ Both

☐ None

☐ Default

*Setup priority

7

*Hold priority

7

Candidate Paths

Sort paths

New Candidate Path

index

200

...

Endpoint-override

--

Explicit path name

--

Metric *

IGP

Affinity set name

--

Bandwidth

5

Bandwidth Unit

Gbps

Confirm

46 | SEGMENT ROUTING SOLUTION APPLICATION NOTE



Conclusion

The leaders in disaggregated networking have teamed up to provide an elegant solution for your traffic engineering needs. TD and OcNOS both provide robust capabilities that enable seamless end to end traffic engineering in your networks today.

Authors

Dmytro Shypoalov – Founder, Vegvisir Systems

Vincent Schuele – Director, Partner and Customer Engineering, IP Infusion

Appendix A: references

- 1) Traffic Dictator white paper
vegvisir.ie/wp-content/uploads/2024/06/Traffic-Dictator-White-Paper-v1.0.pdf
- 2) Traffic Dictator documentation
vegvisir.ie/documentation/
- 3) OcNOS Segment Routing application note
www.ipinfusion.com/wp-content/uploads/2021/03/03-21-OcNOS-SP-Segment-Routing-v1.pdf
- 4) OcNOS Segment Routing documentation
docs.ipinfusion.com/service-provider-6.6/#page/OcNOS_SP/SegRoute-Config/Overview.html

Appendix B: lab configs

TD

```
router general
  spf delay 200
  spf holddown 5000
  spf rapid-runs 3
  srte distinguisher 1
  reoptimization interval 3600
!
router bgp 65001
  router-id 192.168.123.3
  !
  neighbor 192.168.123.3
    remote-as 65002
    address-family link-state
!
router pcep
  !
  neighbor 192.168.123.3
!
traffic-eng affinities
  affinity-map
    name BLUE bit-position 0
    name YELLOW bit-position 1
  !
  affinity-set BLUE_ONLY
    constraint include-all
    name BLUE
  !
  affinity-set YELLOW_ONLY
    constraint include-all
    name YELLOW
!
traffic-eng nodes
  !
  node 2.2.2.2
    !
    neighbor 192.168.123.2
      affinity BLUE
      bandwidth 10 gbps
!
traffic-eng policies
  !
  policy R1_R3_ANYCAST
    headend 1.1.1.1 topology-id 1
    endpoint 3.3.3.3 service-loopback 172.16.101.3
    priority 7 7
    install direct pcep 192.168.123.1
    !
    candidate-path preference 200
    explicit-path R45_ANYCAST
  !
  policy R3_R2_EPE
    headend 3.3.3.3 topology-id 1
    endpoint 192.168.123.2 service-loopback 172.16.101.2
    priority 7 7
    install direct pcep 192.168.123.3
    !
    candidate-path preference 200
  !
  policy R3_R2_EPE_BLUE
```

```

    headend 3.3.3.3 topology-id 1
    endpoint 192.168.123.2 service-loopback 172.16.102.2
    priority 7 7
    install direct pcep 192.168.123.3
    !
    candidate-path preference 200
        affinity-set BLUE_ONLY
    !
policy R3_R4_BLUE
    headend 3.3.3.3 topology-id 1
    endpoint 4.4.4.4 service-loopback 172.16.102.4
    priority 7 7
    install direct pcep 192.168.123.3
    !
    candidate-path preference 200
        affinity-set BLUE_ONLY
        bandwidth 40 gbps
    !
policy R3_R4_PATH1
    headend 3.3.3.3 topology-id 1
    endpoint 4.4.4.4 service-loopback 172.16.102.4
    priority 7 7
    install direct pcep 192.168.123.3
    !
    candidate-path preference 200
        disjoint-group 100 link
    !
policy R3_R4_PATH2
    headend 3.3.3.3 topology-id 1
    endpoint 4.4.4.4 service-loopback 172.16.101.4
    priority 7 7
    install direct pcep 192.168.123.3
    !
    candidate-path preference 200
        disjoint-group 100 link
    !
policy R3_R4_YELLOW
    headend 3.3.3.3 topology-id 1
    endpoint 4.4.4.4 service-loopback 172.16.101.4
    priority 7 7
    install direct pcep 192.168.123.3
    !
    candidate-path preference 200
        affinity-set YELLOW_ONLY
        bandwidth 4 gbps
    !
policy R3_R5_via_R1_R2
    headend 3.3.3.3 topology-id 1
    endpoint 5.5.5.5 service-loopback 172.16.101.5
    priority 7 7
    install direct pcep 192.168.123.3
    !
    candidate-path preference 200
        explicit-path via_R1_R2
        bandwidth 2 gbps
    !
traffic-eng peer-groups
!
traffic-eng explicit-paths
!
explicit-path R45_ANYCAST
    index 10 loose 172.20.45.45
    !
explicit-path via_R1_R2
    index 10 strict 10.100.2.1
    index 20 strict 10.100.1.2

```

```

!
management api http-commands
  protocol http
  port 80
  protocol https
  port 443
!
management syslog
!
management users
!
management licenses
  license general
!
end

```

R1

```

hostname R1
admin-group BLUE 0
admin-group YELLOW 1
!
segment-routing
  global block 16000 23999
!
interface ce0
  ip address 10.100.1.1/24
  label-switching
  admin-group BLUE
  isis network point-to-point
  isis circuit-type level-2-only
  ip router isis 1
!
interface cel
  ip address 10.100.2.1/24
  label-switching
  admin-group BLUE
  isis network point-to-point
  isis circuit-type level-2-only
  ip router isis 1
!
interface lo
  ip address 127.0.0.1/8
  ip address 1.1.1.1/32 secondary
  ipv6 address ::1/128
  prefix-sid index 1
  ip router isis 1
!
router isis 1
  is-type level-2-only
  metric-style wide
  mpls traffic-eng router-id 1.1.1.1
  mpls traffic-eng level-2
  dynamic-hostname
  net 49.0001.0001.0001.00
  segment-routing mpls

```

```

hostname R2
admin-group BLUE 0
admin-group YELLOW 1
!
segment-routing
  global block 16000 23999
!
interface cd0
  ip address 10.100.4.2/24
  label-switching
  admin-group YELLOW
  isis network point-to-point
  isis circuit-type level-2-only
  ip router isis 1
!
interface cd1
  ip address 10.100.3.2/24
  label-switching
  admin-group YELLOW
  isis network point-to-point
  isis circuit-type level-2-only
  ip router isis 1
!
interface ce2
  ip address 10.100.1.2/24
  label-switching
  admin-group BLUE
  isis network point-to-point
  isis circuit-type level-2-only
  ip router isis 1
!
interface eth0
  ip vrf forwarding management
  ip address dhcp
!
interface lo
  ip address 127.0.0.1/8
  ip address 2.2.2.2/32 secondary
  ipv6 address ::1/128
  prefix-sid index 2
  ip router isis 1
!
interface xe4.3011
  encapsulation dot1q 3011
  ip address 192.168.123.102/24
!
interface xel1
  ip address 10.100.5.2/24
  label-switching
  admin-group BLUE
  isis network point-to-point
  isis circuit-type level-2-only
  ip router isis 1
!
router isis 1
  is-type level-2-only
  metric-style wide
  mpls traffic-eng router-id 2.2.2.2
  mpls traffic-eng level-2
  dynamic-hostname
  net 49.0002.0002.0002.00
  segment-routing mpls

```

```

!
router bgp 65002
  bgp router-id 2.2.2.2
  neighbor 3.3.3.3 remote-as 65002
  neighbor 3.3.3.3 update-source 2.2.2.2
  neighbor 192.168.123.2 remote-as 101
  !
  address-family ipv4 unicast
    neighbor 192.168.123.2 activate
  exit-address-family
  !
  address-family link-state link-state
    neighbor 3.3.3.3 activate
  exit-address-family
  egress-engineering
    neighbor 192.168.123.2 peer-node
  exit-egress-engineering
  !
exit

```

R3

```

hostname R3
admin-group BLUE 0
admin-group YELLOW 1
!
ip vrf vrf101
  rd 101:101
  route-target both 101:101
!
ip vrf vrf102
  rd 102:102
  route-target both 102:102
!
segment-routing
  global block 16000 23999
!
interface cd0
  ip address 10.100.5.3/24
  label-switching
  admin-group BLUE
  isis network point-to-point
  isis circuit-type level-2-only
  ip router isis 1
!
interface cd1
  ip address 10.100.3.3/24
  label-switching
  admin-group YELLOW
  isis network point-to-point
  isis circuit-type level-2-only
  ip router isis 1
!
interface ce2
  ip address 10.100.2.3/24
  label-switching
  admin-group BLUE
  isis network point-to-point
  isis circuit-type level-2-only
  ip router isis 1
!
interface lo

```

```

ip address 127.0.0.1/8
ip address 3.3.3.3/32 secondary
ipv6 address ::1/128
prefix-sid index 3
isis circuit-type level-2-only
ip router isis 1
!
interface xe4.3011
encapsulation dot1q 3011
ip address 192.168.123.3/24
!
interface xe11
ip address 10.100.7.3/24
label-switching
admin-group YELLOW
isis network point-to-point
isis circuit-type level-2-only
ip router isis 1
!
interface xe12
ip vrf forwarding vrf101
ip address 10.101.0.3/24
!
interface xe13
ip vrf forwarding vrf102
ip address 10.102.0.3/24
!
router isis 1
is-type level-2-only
distribute bgp-ls
metric-style wide
mpls traffic-eng router-id 3.3.3.3
mpls traffic-eng level-2
dynamic-hostname
net 49.0003.0003.0003.00
segment-routing mpls
!
router bgp 65002
bgp router-id 3.3.3.3
neighbor 2.2.2.2 remote-as 65002
neighbor 2.2.2.2 update-source 3.3.3.3
neighbor 4.4.4.4 remote-as 65002
neighbor 5.5.5.5 remote-as 65002
neighbor 192.168.123.1 remote-as 65001
!
address-family ipv4 labeled-unicast
neighbor 4.4.4.4 activate
neighbor 5.5.5.5 activate
exit-address-family
!
address-family vpnv4 unicast
neighbor 4.4.4.4 activate
exit-address-family
!
address-family link-state link-state
neighbor 2.2.2.2 activate
neighbor 192.168.123.1 activate
exit-address-family
!
address-family ipv4 vrf vrf101
redistribute connected
exit-address-family
!
address-family ipv4 vrf vrf102
redistribute connected
exit-address-family

```

```

!
exit
!
pce configuration 1
  capability
    segment-routing pcep
    pce instantiation
  exit-capability
!
  update-source 192.168.123.3
  peer-address ipv4 192.168.123.1
!

```

R4

```

hostname R4
admin-group BLUE 0
admin-group YELLOW 1
!
ip vrf vrf101
  rd 101:101
  route-target both 101:101
!
ip vrf vrf102
  rd 102:102
  route-target both 102:102
!
segment-routing
  global block 16000 23999
!
ip prefix-list VRF101
  seq 5 permit 10.101.1.0/24
!
ip prefix-list VRF102
  seq 5 permit 10.102.1.0/24
!
route-map SET_VPN_NH permit 10
  match ip address prefix-list VRF101
  set vpnv4 next-hop 172.16.101.4
!
route-map SET_VPN_NH permit 20
  match ip address prefix-list VRF102
  set vpnv4 next-hop 172.16.102.4
!
interface cd0
  ip address 10.100.4.4/24
  label-switching
  admin-group YELLOW
  isis network point-to-point
  isis circuit-type level-2-only
  ip router isis 1
!
interface cd1
  ip address 10.100.5.4/24
  label-switching
  admin-group BLUE
  admin-group YELLOW
  isis network point-to-point
  isis circuit-type level-2-only
  ip router isis 1
!
interface lo
  ip address 127.0.0.1/8

```

```

ip address 4.4.4.4/32 secondary
ipv6 address ::1/128
prefix-sid index 4
ip router isis 1
!
interface loopback45
ip address 172.20.45.45/32
prefix-sid index 45 n-flag-clear
ip router isis 1
!
interface loopback101
ip address 172.16.101.4/32
!
interface loopback102
ip address 172.16.102.4/32
!
interface xe7
ip vrf forwarding vrf101
ip address 10.101.1.4/24
!
interface xe8
ip vrf forwarding vrf102
ip address 10.102.1.4/24
!
router isis 1
is-type level-2-only
metric-style wide
mpls traffic-eng router-id 4.4.4.4
mpls traffic-eng level-2
dynamic-hostname
net 49.0004.0004.0004.00
segment-routing mpls
!
router bgp 65002
bgp router-id 4.4.4.4
allocate-label all
neighbor 3.3.3.3 remote-as 65002
!
address-family ipv4 unicast
network 172.16.101.4/32
network 172.16.102.4/32
exit-address-family
!
address-family ipv4 labeled-unicast
neighbor 3.3.3.3 activate
exit-address-family
!
address-family vpnv4 unicast
neighbor 3.3.3.3 activate
neighbor 3.3.3.3 route-map SET_VPN_NH out
exit-address-family
!
address-family ipv4 vrf vrf101
redistribute connected
exit-address-family
!
address-family ipv4 vrf vrf102
redistribute connected
exit-address-family
!
exit

```

```

hostname R5
admin-group BLUE 0
admin-group YELLOW 1
!
segment-routing
  global block 16000 23999
!
interface lo
  ip address 127.0.0.1/8
  ip address 5.5.5.5/32 secondary
  ipv6 address ::1/128
  prefix-sid index 5
  ip router isis 1
!
interface loopback45
  ip address 172.20.45.45/32
  prefix-sid index 45 n-flag-clear
  ip router isis 1
!
interface loopback101
  ip address 172.16.101.5/32
!
interface xel1
  ip address 10.100.5.5/24
  label-switching
admin-group BLUE
  isis network point-to-point
  isis circuit-type level-2-only
  ip router isis 1
!
interface xel2
  ip address 10.100.7.5/24
  label-switching
admin-group BLUE
  isis network point-to-point
  isis circuit-type level-2-only
  ip router isis 1
!
router isis 1
  is-type level-2-only
  metric-style wide
  mpls traffic-eng router-id 5.5.5.5
  mpls traffic-eng level-2
  dynamic-hostname
  net 49.0005.0005.0005.00
  segment-routing mpls
!
router bgp 65002
  bgp router-id 5.5.5.5
  allocate-label all
  neighbor 3.3.3.3 remote-as 65002
!
  address-family ipv4 labeled-unicast
  neighbor 3.3.3.3 activate
  exit-address-family
!
exit

```