Configuring non-Path Computation Element (PCE)-based Inter-Area Segment Routing Traffic Engineering (SR-TE)

Contents

Introduction

This document describes the aspects of understanding, configuring, and verifying the Inter-Area SR-TE without Path Computation Element controller.

Contributed by Elvin Arias, Cisco TAC Engineer.

Prerequisites

There are no prerequisites for this document.

Requirements

There are no specific requirements for this document.

Components Used

The information in this document is based on Cisco IOS-XR® and IOS-XE®.

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

Multi-Domain SR-TE Introduction

Segment Routing Traffic Engineering (SR-TE) provides the capabilities to steer traffic through the core without forming any state sessions in the core. An SR-TE policy is expressed as a list of segments that specifies a path, called Segment ID (SID) list. No signaling is required as state is in the packet and SID list are processed as a set of instructions by the transit SR-enabled routers.

Multi-domain have been traditionally implemented with Resource Reservation Protocol Traffic Engineering (RSVP-TE) via the use of loose next-hop expansion in an explicit path option. When performing computations an administrator would create a path where inter-area Internet Protocol (IP) addresses are loosely defined in order to allow end-to-end computation via Constrained Shortest Path First (CSPF).

SR-TE does not have the concept of loose next-hops, so and for multi-domain computations the question is how can this be performed?. Computations are possible and the de facto design is to place a centralized controller (XTC, WAE, NOS) in order to perform the corresponding multi-domain computations. Offloading of the computations from head-end to tal end will allow devices to compute paths without having visibility to the entire topology. For this Path Computation Element (PCE) entity is used and the idea is that this entity has the entire visibility of the domain, performs computations and keeps track of the LSPs computed.

In cases where having a controller is temporarily not possible and multi-domain computations are necessary in the Segment Routing core, we can perform different configurations to allow tunnels to establish in inter-area scenarios.

Path Types

SR-TE allows us to define multiple path types, generally known as Explicit paths and Dynamic paths. For dynamic and explicit paths this is straightforward, we let SR-TE algorithm to compute the path based on a dynamic criteria, often TE or IGP metric to a tail end. For explicit paths we can define multiple types, among many we can do:

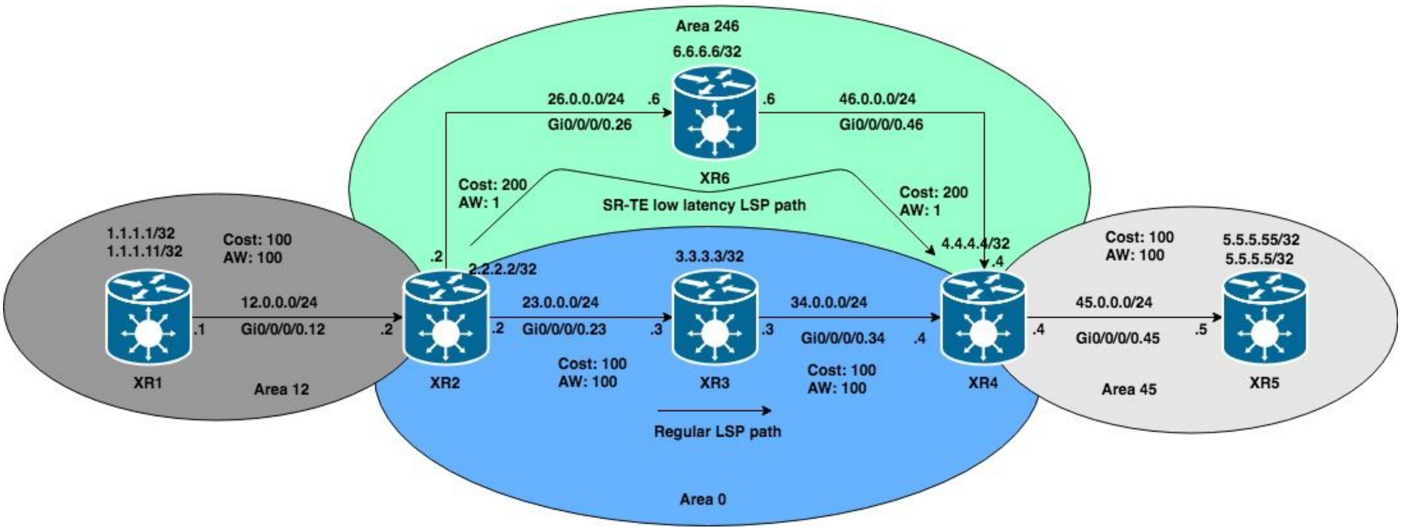
- SID only as label (MPLS only)
- SID only as IPv6 address (SRv6 only)
- IPv4 node address with optional SID
- IPv6 node address with optional SID
- IPv4 address + interface index with optional SID
- IPv4 local and remote addresses with optional SID
- IPv6 + interface index with optional SID
- IPv6 local and remote addresses with optional SID

When defining inter-area SR-TE policies, we must define explicit paths towards the tail end, this is because we don't have the entire visibility of the topology. For inter-area SR-TE we need to configure the policies as follows:

- Explicit-path with tail end SID-label
- Explicit path with transit + SID-label
- Explicit path with local IPv4 addresses + SID-label

Note: If dynamic inter-area path-options are required then the path computation must be delegated to a PCE entity.

Topology diagram



For the next cases, we will use this OSPF inter-area topology, and examples will be based on trying to compute SR-TE tunnels from XR1 to XR5 crossing the area boundaries.

Spoiler

Note: Examples for SR-TE are based on OSPF, but it is also applicable to IS-IS.

Note: Examples for SR-TE are based on OSPF, but it is also applicable to IS-IS.

Initial Configurations

XR1

```
hostname XR1
icmp ipv4 rate-limit unreachable disable
interface Loopback0
 ipv4 address 1.1.1.1 255.255.255.255
!
interface Loopback1
 ipv4 address 1.1.1.11 255.255.255.255
!
interface GigabitEthernet0/0/0/0.12
 ipv4 address 12.0.0.1 255.255.255.0
 encapsulation dot1q 12
!
router ospf 1
 router-id 1.1.1.1
 segment-routing mpls
 segment-routing forwarding mpls
 segment-routing sr-prefer
 address-family ipv4
 area 12
  mpls traffic-eng
  interface Loopback0
   prefix-sid index 1
  !
  interface Loopback1
   prefix-sid index 11
  !
  interface GigabitEthernet0/0/0/0.12
   cost 100
   network point-to-point
  !
```

```
 !
 mpls traffic-eng router-id Loopback0
!
mpls traffic-eng
 interface GigabitEthernet0/0/0/0.12
  admin-weight 100
 !
!
end
```

XR2

```
hostname XR2
logging console debugging
explicit-path identifier 4
 index 10 next-label 16004
!
interface Loopback0
 ipv4 address 2.2.2.2 255.255.255.255
!
interface GigabitEthernet0/0/0/0.12
 ipv4 address 12.0.0.2 255.255.255.0
 encapsulation dot1q 12
!
interface GigabitEthernet0/0/0/0.23
 ipv4 address 23.0.0.2 255.255.255.0
 encapsulation dot1q 23
!
interface GigabitEthernet0/0/0/0.26
 ipv4 address 26.0.0.2 255.255.255.0
 encapsulation dot1q 26
!
router ospf 1
 router-id 2.2.2.2
 segment-routing mpls
 segment-routing forwarding mpls
 segment-routing sr-prefer
 address-family ipv4
 area 0
  mpls traffic-eng
  interface Loopback0
   prefix-sid index 2
  !
  interface GigabitEthernet0/0/0/0.23
   cost 100
   network point-to-point
  !
 !
 area 12
  mpls traffic-eng
  interface GigabitEthernet0/0/0/0.12
   cost 100
   network point-to-point
  !
 !
 area 246
  mpls traffic-eng
  interface GigabitEthernet0/0/0/0.26
   cost 200
   network point-to-point
  !
 !
 mpls traffic-eng router-id Loopback0
!
```

```
mpls oam
!
mpls traffic-eng
 interface GigabitEthernet0/0/0/0.12
  admin-weight 100
 !
 interface GigabitEthernet0/0/0/0.23
  admin-weight 100
 !
 interface GigabitEthernet0/0/0/0.26
  admin-weight 1
 !
!
end
```

XR3

```
hostname XRv3
interface Loopback0
 ipv4 address 3.3.3.3 255.255.255.255
!
interface MgmtEth0/0/CPU0/0
 shutdown
!
interface GigabitEthernet0/0/0/0.23
 ipv4 address 23.0.0.3 255.255.255.0
 encapsulation dot1q 23
!
interface GigabitEthernet0/0/0/0.34
 ipv4 address 34.0.0.3 255.255.255.0
 encapsulation dot1q 34
!
router ospf 1
 router-id 3.3.3.3
 segment-routing mpls
 segment-routing forwarding mpls
 segment-routing sr-prefer
 address-family ipv4
 area 0
  mpls traffic-eng
  interface Loopback0
   prefix-sid index 3
  !
  interface GigabitEthernet0/0/0/0.23
   cost 100
   network point-to-point
  !
  interface GigabitEthernet0/0/0/0.34
   cost 100
   network point-to-point
  !
 !
 mpls traffic-eng router-id Loopback0
!
mpls oam
!
mpls traffic-eng
 interface GigabitEthernet0/0/0/0.23
  admin-weight 100
 !
 interface GigabitEthernet0/0/0/0.34
  admin-weight 100
 !
```

```
!
end
```

XR4

```
hostname XR4
interface Loopback0
 ipv4 address 4.4.4.4 255.255.255.255
!
interface GigabitEthernet0/0/0/0.34
 ipv4 address 34.0.0.4 255.255.255.0
 encapsulation dot1q 34
!
interface GigabitEthernet0/0/0/0.45
 ipv4 address 45.0.0.4 255.255.255.0
 encapsulation dot1q 45
!
interface GigabitEthernet0/0/0/0.46
 ipv4 address 46.0.0.4 255.255.255.0
 encapsulation dot1q 46
!
router ospf 1
 distribute bgp-ls
 router-id 4.4.4.4
 segment-routing mpls
 segment-routing forwarding mpls
 segment-routing sr-prefer
 address-family ipv4
 area 0
  mpls traffic-eng
  interface Loopback0
   prefix-sid index 4
  !
  interface GigabitEthernet0/0/0/0.34
   cost 100
   network point-to-point
  !
 !
 area 45
  mpls traffic-eng
  interface GigabitEthernet0/0/0/0.45
   cost 100
   network point-to-point
  !
 !
 area 246
  mpls traffic-eng
  interface GigabitEthernet0/0/0/0.46
   cost 200
   network point-to-point
  !
 !
 mpls traffic-eng router-id Loopback0
!
mpls oam
!
mpls traffic-eng
 interface GigabitEthernet0/0/0/0.34
  admin-weight 100
 !
 interface GigabitEthernet0/0/0/0.45
  admin-weight 100
 !
 interface GigabitEthernet0/0/0/0.46
```

```
  admin-weight 1
 !
!
end
```

XR5

```
hostname XRv5
interface Loopback0
 ipv4 address 5.5.5.5 255.255.255.255
!
interface Loopback1
 ipv4 address 5.5.5.55 255.255.255.255
!
interface GigabitEthernet0/0/0/0.45
 ipv4 address 45.0.0.5 255.255.255.0
 encapsulation dot1q 45
!
router ospf 1
 router-id 5.5.5.5
 segment-routing mpls
 segment-routing forwarding mpls
 segment-routing sr-prefer
 address-family ipv4
 area 45
  mpls traffic-eng
  interface Loopback0
   prefix-sid index 5
  !
  interface Loopback1
   prefix-sid index 55
  !
  interface GigabitEthernet0/0/0/0.45
   cost 100
   network point-to-point
  !
 !
 mpls traffic-eng router-id Loopback0
!
mpls oam
!
mpls traffic-eng
 interface GigabitEthernet0/0/0/0.45
  admin-weight 100
 !
!
end
```

XR6

```
hostname XR6
icmp ipv4 rate-limit unreachable disable
interface Loopback0
 ipv4 address 6.6.6.6 255.255.255.255
!
interface GigabitEthernet0/0/0/0.26
 ipv4 address 26.0.0.6 255.255.255.0
 encapsulation dot1q 26
!
interface GigabitEthernet0/0/0/0.46
 ipv4 address 46.0.0.6 255.255.255.0
 encapsulation dot1q 46
!
router ospf 1
```

```
 router-id 6.6.6.6
 segment-routing mpls
 segment-routing forwarding mpls
 segment-routing sr-prefer
 address-family ipv4
 area 246
  mpls traffic-eng
  interface Loopback0
   prefix-sid index 6
  !
  interface GigabitEthernet0/0/0/0.26
   cost 200
   network point-to-point
  !
  interface GigabitEthernet0/0/0/0.46
   cost 200
   network point-to-point
  !
 !
 mpls traffic-eng router-id Loopback0
!
mpls oam
!
mpls traffic-eng
 interface GigabitEthernet0/0/0/0.26
  admin-weight 1
 !
 interface GigabitEthernet0/0/0/0.46
  admin-weight 1
 !
!
end
```

Devices in the OSPF domain have built LSPs between them, we can verify this by checking the LSP between XR1 to XR5.

```
RP/0/0/CPU0:XR1#ping mpls ipv4 5.5.5.5/32 fec-type generic verbose
Sending 5, 100-byte MPLS Echos to 5.5.5.5/32, timeout is 2 seconds, send interval is 0 msec:
Codes: '!' - success, 'Q' - request not sent, '.' - timeout, 'L' - labeled output interface, 'B'
- unlabeled output interface, 'D' - DS Map mismatch, 'F' - no FEC mapping, 'f' - FEC mismatch,
'M' - malformed request, 'm' - unsupported tlvs, 'N' - no rx label, 'P' - no rx intf label prot,
'p' - premature termination of LSP, 'R' - transit router, 'I' - unknown upstream index, 'X' -
unknown return code, 'x' - return code 0 Type escape sequence to abort.
!      size 100, reply addr 45.0.0.5, return code 3
!      size 100, reply addr 45.0.0.5, return code 3
!      size 100, reply addr 45.0.0.5, return code 3
!      size 100, reply addr 45.0.0.5, return code 3
!      size 100, reply addr 45.0.0.5, return code 3

Success rate is 100 percent (5/5), round-trip min/avg/max = 1/6/10 ms
```

SR-TE Policy Configurations

Case #1: Inter-area SR-TE tunnel with explicit path with prefix-SID of tail end

We will create an SR-TE policy from XR1 to compute a path towards XR5 prefix-SID corresponding to 5.5.5.5/32. Prefix 5.5.5.5/32 has been configured with an index of 5, this is the only label that we will provide to PCALC to compute the path.

Note: All routers in the topology have the same SRGB block.

```
RP/0/0/CPU0:XR1#ping mpls ipv4 5.5.5.5/32 fec-type generic verbose
Sending 5, 100-byte MPLS Echos to 5.5.5.5/32, timeout is 2 seconds, send interval is 0 msec:
Codes: '!' - success, 'Q' - request not sent, '.' - timeout, 'L' - labeled output interface, 'B'
- unlabeled output interface, 'D' - DS Map mismatch, 'F' - no FEC mapping, 'f' - FEC mismatch,
'M' - malformed request, 'm' - unsupported tlvs, 'N' - no rx label, 'P' - no rx intf label prot,
'p' - premature termination of LSP, 'R' - transit router, 'I' - unknown upstream index, 'X' -
unknown return code, 'x' - return code 0 Type escape sequence to abort.
!      size 100, reply addr 45.0.0.5, return code 3
!      size 100, reply addr 45.0.0.5, return code 3
!      size 100, reply addr 45.0.0.5, return code 3
!      size 100, reply addr 45.0.0.5, return code 3
!      size 100, reply addr 45.0.0.5, return code 3

Success rate is 100 percent (5/5), round-trip min/avg/max = 1/6/10 ms
```

Spoiler

Note: Autoroute announce does not work in inter-area cases.

Note: Autoroute announce does not work in inter-area cases.

Verifications

When we provide a SID list as the input for the computation, only the first label is verified, and if this condition is met the tunnel will be up. If we verify the tunnel, we can see that is up and routing is being performed.

```
RP/0/0/CPU0:XR1#show mpls traffic-eng tunnels segment-routing p2p 15

Name: tunnel-te15  Destination: 5.5.5.5  Ifhandle:0x130
  Signalled-Name: XR1_t15
  Status:
    Admin:    up Oper:   up   Path:  valid   Signalling: connected

    path option 1, (Segment-Routing) type explicit CASE1 (Basis for Setup)
    G-PID: 0x0800 (derived from egress interface properties)
    Bandwidth Requested: 0 kbps  CT0
    Creation Time: Mon Nov 26 02:14:33 2018 (00:14:34 ago)
  Config Parameters:
    Bandwidth:        0 kbps (CT0) Priority:  7  7 Affinity: 0x0/0xffff
    Metric Type: TE (interface)
    Path Selection:
      Tiebreaker: Min-fill (default)
      Protection: Unprotected Adjacency
    Hop-limit: disabled
    Cost-limit: disabled
    Path-invalidation timeout: 10000 msec (default), Action: Tear (default)
    AutoRoute: disabled  LockDown: disabled   Policy class: not set
    Forward class: 0 (default)
    Forwarding-Adjacency: disabled
    Autoroute Destinations: 1
    Loadshare:          0 equal loadshares
    Auto-bw: disabled
    Path Protection: Not Enabled
    BFD Fast Detection: Disabled
    Reoptimization after affinity failure: Enabled
    SRLG discovery: Disabled
  History:
    Tunnel has been up for: 00:04:43 (since Mon Nov 26 02:24:24 UTC 2018)
    Current LSP:
      Uptime: 00:04:43 (since Mon Nov 26 02:24:24 UTC 2018)
    Prior LSP:
```

```
      ID: 5 Path Option: 1
      Removal Trigger: tunnel shutdown

  Segment-Routing Path Info (OSPF 1 area 12)
    Segment0[Node]: 5.5.5.5, Label: 16005
Displayed 1 (of 1) heads, 0 (of 0) midpoints, 0 (of 0) tails
Displayed 1 up, 0 down, 0 recovering, 0 recovered heads
```

Spoiler

Note: PCALC events can be verified with debug mpls traffic-eng path lookup command.

Note: PCALC events can be verified with debug mpls traffic-eng path lookup command.

If we check the global RIB, we can see that routing to 5.5.5.5/32 is set via tunnel interface 15.

```
RP/0/0/CPU0:XR1#show route 5.5.5.5
Routing entry for 5.5.5.5/32
  Known via "te-client", distance 2, metric 401 (connected)
  Installed Nov 26 02:24:24.336 for 00:07:03
  Routing Descriptor Blocks
    directly connected, via tunnel-te15
      Route metric is 401
  No advertising protos.
```

If we check the LFIB, we can see that tunnel-te15 has been installed and is ready for forwarding.

```
RP/0/0/CPU0:XR1#ping 5.5.5.5 source 1.1.1.1 repeat 100 size 1500

Type escape sequence to abort.
Sending 100, 1500-byte ICMP Echos to 5.5.5.5, timeout is 2 seconds:
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
Success rate is 100 percent (100/100), round-trip min/avg/max = 9/12/19 ms

RP/0/0/CPU0:XR1#show mpls forwarding tunnels detail
Tunnel        Outgoing    Outgoing     Next Hop        Bytes
Name          Label       Interface                    Switched
------------- ----------- ------------ --------------- ------------
tt15         (SR) 16005       Gi0/0/0/0.12 12.0.0.2        150400
     Updated: Nov 26 02:24:24.357
     Version: 200, Priority: 2
     Label Stack (Top -> Bottom): { 16005 }
     NHID: 0x0, Encap-ID: N/A, Path idx: 0, Backup path idx: 0, Weight: 0
     MAC/Encaps: 18/22, MTU: 1500
     Packets Switched: 100

  Interface Name: tunnel-te15, Interface Handle: 0x00000130, Local Label: 24003
  Forwarding Class: 0, Weight: 0
  Packets/Bytes Switched: 100/150000
```

Case #2: Inter-area SR-TE tunnel with explicit path with IPv4 addresses locally + prefix-SIDs

When defining SR-TE policies for inter-area, we have the option to mix labels and IPv4 addresses. For the PCALC to successfully compute a path to the tail end, the IPv4 addresses provided for the calculation must be local of the area, and for elements that are outside the area, we must provide either prefix adjacency SIDs.

```
RP/0/0/CPU0:XR1#show mpls forwarding tunnels detail
Tunnel        Outgoing    Outgoing     Next Hop        Bytes
Name          Label       Interface                    Switched
------------- ----------- ------------ --------------- ------------
tt15         (SR) 16005       Gi0/0/0/0.12 12.0.0.2        150400
     Updated: Nov 26 02:24:24.357
     Version: 200, Priority: 2
     Label Stack (Top -> Bottom): { 16005 }
     NHID: 0x0, Encap-ID: N/A, Path idx: 0, Backup path idx: 0, Weight: 0
     MAC/Encaps: 18/22, MTU: 1500
     Packets Switched: 100

  Interface Name: tunnel-te15, Interface Handle: 0x00000130, Local Label: 24003
  Forwarding Class: 0, Weight: 0
  Packets/Bytes Switched: 100/150000
```

Verifications

As observed, we have indicated to PCALC that path must go through via XR6 (16006) and then to the final prefix SID (16005). Verifying the tunnel computation results we can see how it was computed.

```
RP/0/0/CPU0:XR1#show mpls traffic-eng tunnels segment-routing p2p 15

Name: tunnel-te15  Destination: 5.5.5.5  Ifhandle:0x130
  Signalled-Name: XR1_t15
  Status:
    Admin:    up Oper:   up   Path:  valid   Signalling: connected

    path option 1, (Segment-Routing) type explicit CASE2 (Basis for Setup)
    G-PID: 0x0800 (derived from egress interface properties)
    Bandwidth Requested: 0 kbps  CT0
    Creation Time: Mon Nov 26 02:14:33 2018 (00:40:44 ago)
  Config Parameters:
    Bandwidth:        0 kbps (CT0) Priority:  7  7 Affinity: 0x0/0xffff
    Metric Type: TE (interface)
    Path Selection:
      Tiebreaker: Min-fill (default)
      Protection: Unprotected Adjacency
    Hop-limit: disabled
    Cost-limit: disabled
    Path-invalidation timeout: 10000 msec (default), Action: Tear (default)
    AutoRoute: disabled  LockDown: disabled   Policy class: not set
    Forward class: 0 (default)
    Forwarding-Adjacency: disabled
    Autoroute Destinations: 1
    Loadshare:          0 equal loadshares
    Auto-bw: disabled
    Path Protection: Not Enabled
    BFD Fast Detection: Disabled
    Reoptimization after affinity failure: Enabled
    SRLG discovery: Disabled
  History:
    Tunnel has been up for: 00:08:47 (since Mon Nov 26 02:46:30 UTC 2018)
    Current LSP:
      Uptime: 00:00:10 (since Mon Nov 26 02:55:07 UTC 2018)
    Reopt. LSP:
      Last Failure:
        LSP not signalled, identical to the [CURRENT] LSP
```

```
        Date/Time: Mon Nov 26 02:52:43 UTC 2018 [00:02:34 ago]
    Prior LSP:
      ID: 9 Path Option: 1
      Removal Trigger: reoptimization completed

  Segment-Routing Path Info (OSPF 1 area 12)
    Segment0[Link]: 12.0.0.1 - 12.0.0.2, Label: 24001
    Segment1[Node]: 6.6.6.6, Label: 16006
    Segment2[Node]: 5.5.5.5, Label: 16005
Displayed 1 (of 1) heads, 0 (of 0) midpoints, 0 (of 0) tails
Displayed 1 up, 0 down, 0 recovering, 0 recovered heads
```

If we traceroute, we can see the next-hops we effectively go through XR6.

```
RP/0/0/CPU0:XR1#traceroute 5.5.5.5 source 1.1.1.1

Type escape sequence to abort.
Tracing the route to 5.5.5.5

 1  12.0.0.2 [MPLS: Labels 16006/16005 Exp 0] 9 msec  0 msec  0 msec
 2  26.0.0.6 [MPLS: Label 16005 Exp 0] 0 msec  0 msec  0 msec
 3  46.0.0.4 [MPLS: Label 16005 Exp 0] 0 msec  9 msec  0 msec
 4  45.0.0.5 9 msec  *  9 msec
```

Case #3: Inter-area SR-TE tunnel with explicit path with IPv4 addresses locally + prefix-SID suboptimal routing

We can have situations where we define the prefix-SIDs, but form suboptimal or looping traffic patterns. In this case, we will create this scenario.

```
RP/0/0/CPU0:XR1#traceroute 5.5.5.5 source 1.1.1.1

Type escape sequence to abort.
Tracing the route to 5.5.5.5

 1  12.0.0.2 [MPLS: Labels 16006/16005 Exp 0] 9 msec  0 msec  0 msec
 2  26.0.0.6 [MPLS: Label 16005 Exp 0] 0 msec  0 msec  0 msec
 3  46.0.0.4 [MPLS: Label 16005 Exp 0] 0 msec  9 msec  0 msec
 4  45.0.0.5 9 msec  *  9 msec
```

Based on the prefix-SID, we can see that traffic should go through the prefix SIDs of XR6 -> XR2 -> XR3 -> XR5.

```
RP/0/0/CPU0:XR1#show mpls traffic-eng tunnels segment-routing p2p 15

    Admin:    up Oper:   up   Path:  valid   Signalling: connected

    path option 1, (Segment-Routing) type explicit CASE3 (Basis for Setup)

<<Output omitted>>
  Segment-Routing Path Info (OSPF 1 area 12)
    Segment0[Link]: 12.0.0.1 - 12.0.0.2, Label: 24001
    Segment1[Node]: 6.6.6.6, Label: 16006
    Segment2[Node]: 2.2.2.2, Label: 16002
    Segment3[Node]: 3.3.3.3, Label: 16003
    Segment4[Node]: 5.5.5.5, Label: 16005
Displayed 1 (of 1) heads, 0 (of 0) midpoints, 0 (of 0) tails
```

If we trace the path to 5.5.5.5/32 we can see that we have formed a loop between XR2 and XR6, even though this is suboptiomal, we can still route to XR5 5.5.5.5/32 without issues since the LSP is correctly setup.

```
RP/0/0/CPU0:XR1#traceroute 5.5.5.5 source 1.1.1.1


Type escape sequence to abort.
Tracing the route to 5.5.5.5

 1  12.0.0.2 [MPLS: Labels 16006/16002/16003/16005 Exp 0] 19 msec  19 msec  9 msec
 2  26.0.0.6 [MPLS: Labels 16002/16003/16005 Exp 0] 9 msec  9 msec  9 msec
 3  26.0.0.2 [MPLS: Labels 16003/16005 Exp 0] 9 msec  9 msec  9 msec
 4  23.0.0.3 [MPLS: Label 16005 Exp 0] 9 msec  9 msec  9 msec
 5  34.0.0.4 [MPLS: Label 16005 Exp 0] 9 msec  9 msec  9 msec
 6  45.0.0.5 9 msec  *  9 msec
```

Summary

When creating multi-domain policies without PCEs in Segment Routing Traffic Engineering, we do not have the complete view of the link-state database, due to this, we must set explicit paths meeting specific routing requirements, due to the lack of visbility. Inter-area tunnels are possible and will come up by defining explicit paths with IPv4 addresses, adjacency SIDs and/or prefix SIDs on the local area with prefix SIDs of the transit devices and/or tail end of the SR-TE policy. Other explicit path definitions will fail.