Segment Routing - Binding SIDs and the SRLB

A Binding SID, or **BSID**, is a local label that is bound to an **SR Policy**.

A **BSID** is an attribute of a **candidate path** with in **SR Policy** and functions as a unique identifier for the policy. Whichever **candidate path** is used in the **SR Policy**, the resulting incoming/local label in LFIB is the **BSID**.

Steering

Local Steering	Remote Steering
In order to forward traffic *locally* on to the **SR Policy**, the service signalling protocol (for example BGP), installs the service route recursing on the **BSID**.	The MPLS forwarding table adds the incoming label as the **BSID** and the outgoing label as "pop + add **SR Policy** Segments"

Allocation

Dynamic (default)	Explicit/Static
By default the **BSID** is allocated automatically from the dynamic IOS-XR label range 24,000 - 1,048,575. Once allocated all **candidate paths** within an **SR Policy** will use this label.	Explicit allocation is used in situations where the label needs to remain stable or where multiple headends, having similar **SR Policies**, have the same **BSID** values. Explicit allocation comes from the Segment Routing Local Block (see later in this document)

Forcing Static Allocation

```
segment-routing
  traffic-eng
    binding-sid dynamic disable
```

This command will require a headend to configure an explicit label. If this command is in effect then any **candidate paths** without a **BSID** is invalid.

```
segment-routing
  traffic-eng
    binding-sid explicit fallback-dynamic
```

If the explicit **BSID** is unavailable (regardless of how the headend allocates it - via CLI, PCEP or BGP) the **candidate paths** with be invalid. To allow it to fall back to dynamic path if the explicit one is unavailable, use this command on IOS-XR.

An explicit **BSID** can't be configured under an ODN color template. This is because **SR Policies** with different endpoints could be instantiated and would all be bound to the same **BSID**. This would mean multiple different policies would have the same **BSID**, which defeats the purpose of having a **BSID** be a unique identifier.

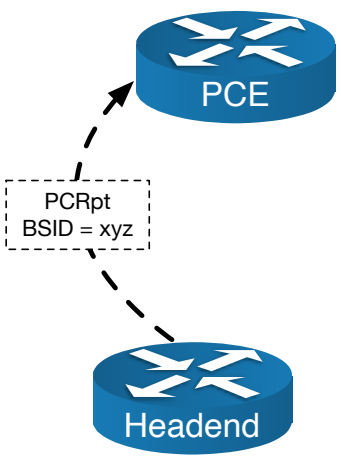
Binding to any tunnel

This document focuses on SR but a BSID can be bound to any tunnel or interface - for example RSVP-TE, GRE or IP tunnels. The below example show how to steer traffic into an RSVP tunnel.

```
interface tunnel-te1
  ipv4 unnumbered Loopback0
  destination 3.3.3.3
  binding-sid mpls label 5001
  path-selection metric te
  path-option 1 dynamic
```

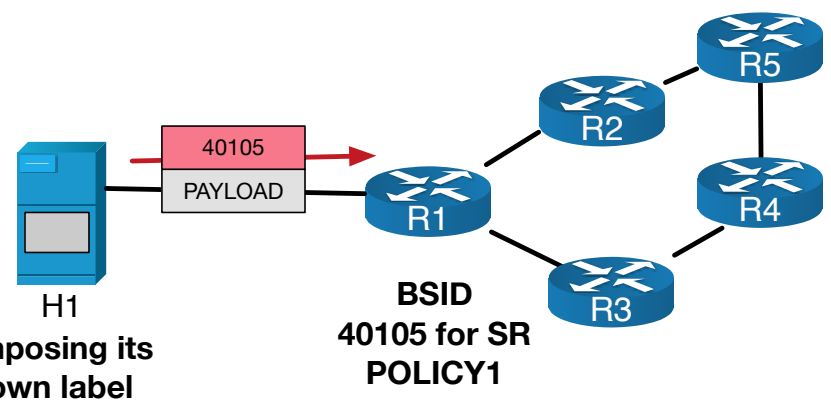
Using a Controller

If a controller is used, for example a PCE, then the BSID will be included in the report message sent to the controller.



Explicit Allocation if Host sets Label

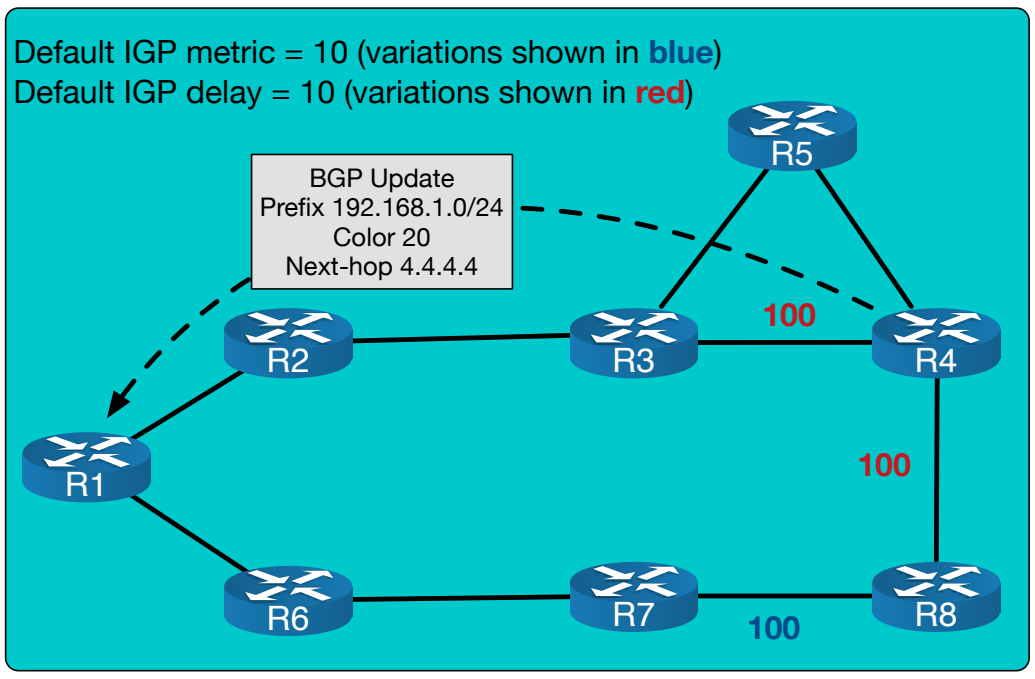
In the below example H1 is setting its label stack, with the BSID on R1 being the top label for particular flows. If the label is dynamically allocated and R1 reloads, the label could change, causing H1s traffic to be lost. This is perfect scenario for static allocation.



Binding SID

Dynamic allocation example

This page shows an example for Segment Routing policy that steers traffic to the prefix 192.168.1.0/24, on R4, via R8. The SID list is explicitly defined and the BSID is dynamically allocated.



The Adj SID from one node to another is of the format ***240xy*** where ***x*** is the node and ***y*** is the neighbor.

Each node has loopback0 x.x.x.x/32 and Node-SID 1600x, where x is the node number.

```
hostname R1
!
segment-routing
  traffic-eng
    segment-list name VIA_R8
      index 10 mpls label 16007
      index 20 mpls label 24078
      index 20 mpls label 16004
    !
  policy POLICY1
    color 20 end-point ipv4 4.4.4.4/32
    candidate-paths
    preference 150
      explicit segment-list VIA_R8
    !
    preference 100
      dynamic
        metric
          type delay
```

```
RP/0/0/CPU0:R1# show segment-routing traffic-eng policy

SR-TE policy database
---------------------

Color: 20, End-point: 4.4.4.4
  Name: srte_c_20_ep_4.4.4.4
  Status:
    Admin: up Operational: up for 00:01:20 (since Aug 5 15:57:65.434)
  Candidate-paths:
    Preference: 150 (configuration) (active)
      Name: POLICY1
      Requested BSID: dynamic
      Explicit: segment-list VIA_R8 (valid)
        Weight: 1, Metric Type: TE
          16007 [Prefix-SID, 7.7.7.7]
          24078
          16004 [Prefix-SID, 4.4.4.4]
    Preference: 100 (configuration)
      Name: POLICY1
      Requested BSID: dynamic
      Dynamic (invalid)
  Attributes:
    Binding SID: 40105
    Forward Class: 0
    Steering BGP disabled: no
    IPv6 caps enable: yes
```

Default is dynamic

40105 has been allocated in this case

Incoming is the BSID. Outgoing is essentially "Pop and add SR policy label stack"

```
RP/0/0/CPU0:R1# show mpls forwarding labels 401045
Local      Outgoing    Prefix      Outgoing                Next Hop        Bytes
Label      Label       or ID       Interface                               Switched
-----      --------    ------      --------------------    -----------     ----------
40105      Pop          No ID      srte_c_20_ep_4.4.4.4    point2point     0

RP/0/0/CPU0:R1# show segment-routing traffic-eng forwarding policy detail
Color      Endpoint    Segment     Outgoing    Outgoing    Next Hop        Bytes
                       List        Label       Interface                   Switched
-----      --------    ------      ----------  ----------  -----------     ----------
20         4.4.4.4     VIA_R8      16007       Gi0/0/0/0   10.1.6.6        0
                   Label Stack (Top > Bottom): { 16007, 24078, 16004 }
                   Path-id: 1, Weight: 64
                   Packets Switched: 0
```

Local steering will recurse on the BSID

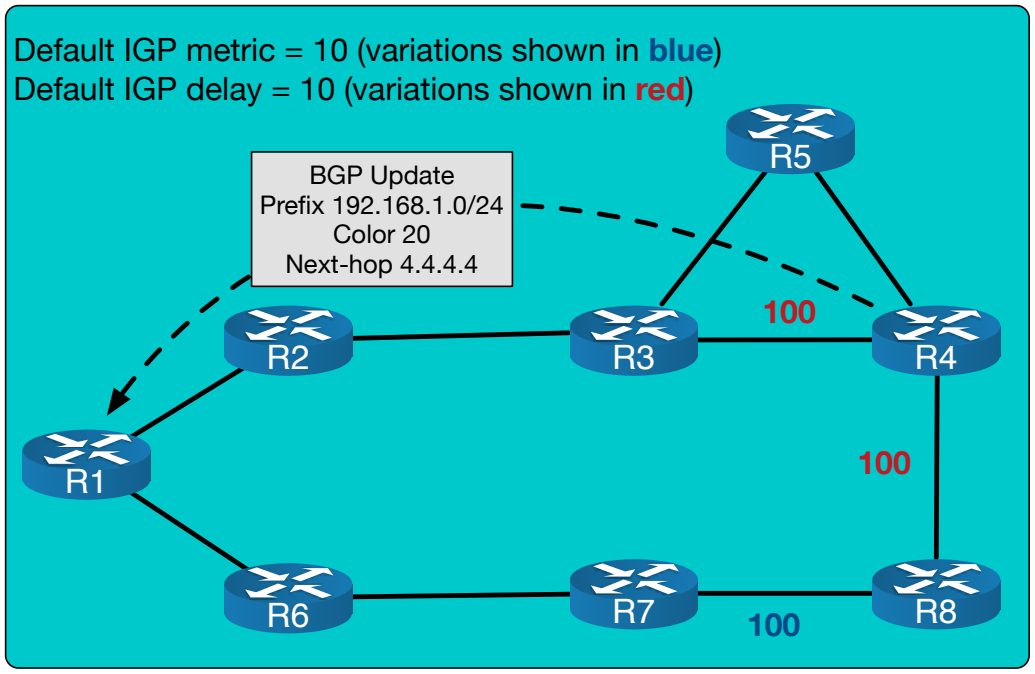
```
RP/0/0/CPU0:R1# show bgp ipv4 unicast 192.168.1.0/24
BGP routing table entry for 192.168.1.0/24
Versions:
  Process           bRIB/RIB   SendTblVer
  Speaker                 6            6
Last Modified: Aug 3 12:32:02.543 for 01:22:00
Paths: (1 available, best #1)
  Not advertised to any peer
  Path #1: Received by speaker 0
  Not advertised to any peer
  Local
    4.4.4.4 C:20 (bsid:40105) (metric 30) from 4.4.4.4 (4.4.4.4)
      Origin IGP, metric 0, localpref 100, valid, internal, best, group-best
      Received Path ID 0, Local Path ID 1, version 6
      Extended community: Color:20
      SR policy color 20, up, registered, bsid 40105 , if-handle 0x000000d0
```

```
RP/0/0/CPU0:R1# show cef 192.168.10/24
192.168.1.0/24, version 46, internal 0x5000002 0x0 (ptr 0xa140fe6e) [1], 0x0
(0x0), 0x0 (0x0)
Updated Aug 3 12:33:46.383
Prefix Len 24, traffic index 0, precedence n/a, priority 4
  via local-label 40105 , 3 dependencies, recursive [flags 0x6000]
    path-idx 0 NHID 0x0 [0xa17cbd10 0x0]
    recursion-via-label
    next hop via 40105 /1/21
      next hop srte_c_20_ep_4.4.4.4
```

Binding SID

Explicit allocation example

This page shows the same setup as that of the previous page, but with explicit SID allocation.



The Adj SID from one node to another is of the format ***240xy*** where ***x*** is the node and ***y*** is the neighbor.

Each node has loopback0 x.x.x.x/32 and Node-SID 1600x, where x is the node number.

```
hostname R1
!
segment-routing
  traffic-eng
    segment-list name VIA_R8
      index 10 mpls label 16007
      index 20 mpls label 24078
      index 20 mpls label 16004
    !
  policy POLICY1
    binding-sid mpls 15000
    color 20 end-point ipv4 4.4.4.4/32
    candidate-paths
    preference 150
      explicit segment-list VIA_R8
    !
    preference 100
      dynamic
        metric
          type delay
```

```
RP/0/0/CPU0:R1# show segment-routing traffic-eng policy

SR-TE policy database
---------------------

Color: 20, End-point: 4.4.4.4
  Name: srte_c_20_ep_4.4.4.4
  Status:
    Admin: up Operational: up for 00:01:20 (since Aug 5 15:57:65.434)
  Candidate-paths:
    Preference: 150 (configuration) (active)
      Name: POLICY1
      Requested BSID: 15000
      Explicit: segment-list VIA_R8 (valid)
        Weight: 1, Metric Type: TE
          16007 [Prefix-SID, 7.7.7.7]
          24078
          16004 [Prefix-SID, 4.4.4.4]
     Preference: 100 (configuration)
       Name: POLICY1
       Requested BSID: dynamic
       Dynamic (invalid)
  Attributes:
    Binding SID: 15000 (SRLB)
    Forward Class: 0
    Steering BGP disabled: no
    IPv6 caps enable: yes
```

Requested shows the explicit value

Label has been allocated from The SRLB

Note the reference to the SRLB here. Index 0 means it is the first label in the SRLB range - namely 15000.

```
RP/0/0/CPU0:R1# show mpls forwarding labels 15000
Local      Outgoing    Prefix          Outgoing                Next Hop       Bytes
Label      Label       or ID           Interface                              Switched
-----      --------    ------          --------------------    -----------    ----------
15000       Pop        SRLB (idx 0)    srte_c_20_ep_4.4.4.4    point2point     0

RP/0/0/CPU0:R1# show segment-routing traffic-eng forwarding policy detail
Color      Endpoint    Segment     Outgoing    Outgoing     Next Hop       Bytes
                       List        Label       Interface                   Switched
-----      --------    ------     ----------  ----------   -----------    ----------
20         4.4.4.4     VIA_R8      16007       Gi0/0/0/0    10.1.6.6       0
                     Label Stack (Top > Bottom): { 16007, 24078, 16004 }
                     Path-id: 1, Weight: 64
                     Packets Switched: 0
```

Segment Routing Local Block

The Segment Routing Local Block (SRLB) is a range of labels reserved for explicit local assignment. This page covers the basics of its configuration and verification.

The default range on IOS-XR is 15,000 - 15,999.

These labels are advertised in IGP extension just like labels from the SRGB (Segment Routing Global Block)

Explicit labels should be allocated from this range to avoid the possibility of the explicit label be unavailable because it has been dynamically allocated beforehand.

```
segment-routing
  local-block 100000 100999
```

This IOS-XR command enables you to change the default SRLB range.

```
segment-routing
  traffic-eng
    binding-sid explicit enforce-srlb
```

With this IOS-XR command, any configuration of an explicit label outside of the SRLB is not allocated and the associated candidate path is invalid.

OSPF Opaque LSA

```
RP/0/0/CPU0:R1# show ospf database opaque-area 4.0.0.0 self-originate

       OSPF Router with ID (1.1.1.1) (Process ID 1)

         Type-10 Opaque Link Area Link States (Area 0)

  LS age: 47
  Options: (No TOS-capability, DC)
  LS Type: Opaque Area Link
  Link State ID: 4.0.0.0
  Opaque Type: 4
  Opaque ID: 0
  Advertising Router: 1.1.1.1
  LS Seq Number: 80000003
  Checksum: 0x47fe
  Length: 88

    Router Information TLV: Length: 4
    Capabilities:
      Graceful Restart Helper Capable
      Stub Router Capable
      All capability bits: 0x60000000

    Segment Routing Algorithm TLV: Length: 2
      Algorithm: 0
      Algorithm: 1

    Segment Routing Range TLV: Length: 12
      Range Size: 8000

        SID sub-TLV: Length 3
          Label: 16000

    Node MSD TLV: Length: 2
      Type: 1, Value 10

    Segment Routing Local Block TLV: Length: 12
      Range Size: 1000

        SID sub-TLV: Length 3
          Label: 15000

    Dynamic Hostname TLV: Length: 6
      Hostname: R1
```

ISIS TLVs

```
RP/0/0/CPU0:R1# show isis database verbose R1

IS-IS 1 (Level-2) Link State Database
LSPID         LSP Seq Num    LSP Checksum  LSP  Holdtime/Rcvd  ATT/P/OL
R1.00-00    * 0x00000178   0x5d23        433  /*             0/0/0
<snip>
  Router Cap: 1.1.1.1 D:0 S:0
    Segment Routing: I:1 V:0, SRGB Base: 16000 Range: 8000
    SR Local Block: Base: 15000 Range: 1000
    SR Algorithm:
      Algorithm: 0
     Algorithm: 1
    Node Maximum SID Depth:
      Label Imposition: 10
<snip>
```

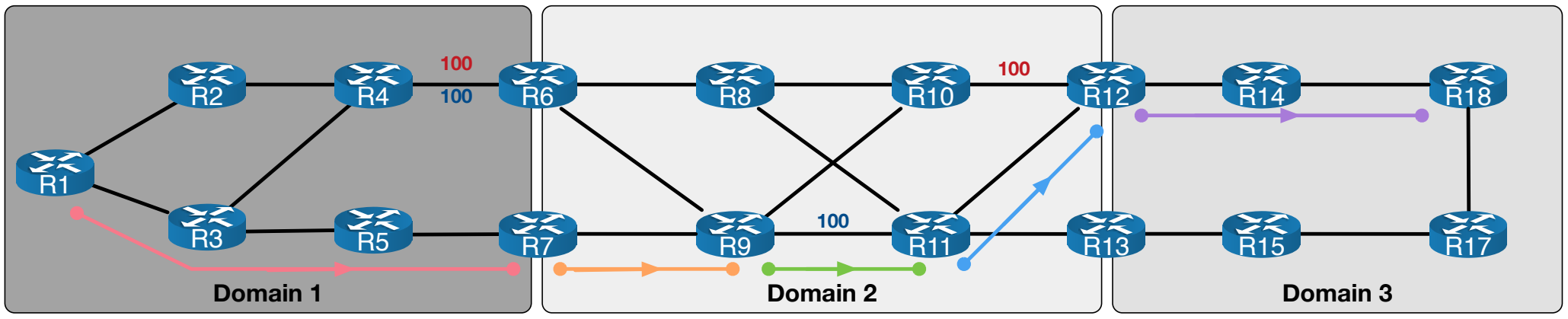
Using these TLVs, a controller can learn both the label ranges and what has already been explicitly allocated.

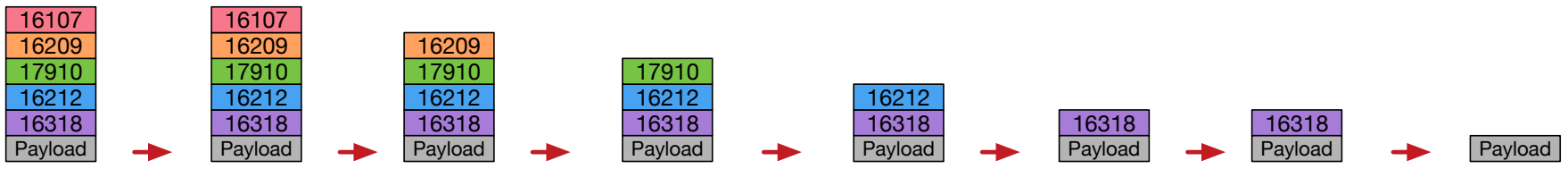
Scaling with a Binding SID

This page shows how inter-domain scaling can be achieved using a Binding SID. In this example, traffic is steered from R1 in Domain 1 to R18 in Domain 3 using the lowest delay metric. Each domain runs its own independent IGP. Because the lowest IGP path doesn't match the lowest delay path within Domain 2, a complicated SID list needs to be used. If a BSID is used by Domain 2 it can simplify the label stack that the headend uses and hide network churn from outside domains.

Multi-domain Policy without Binding SID

Default IGP metric = 10 (variations shown in **blue**)
Default IGP delay = 10 (variations shown in **red**)





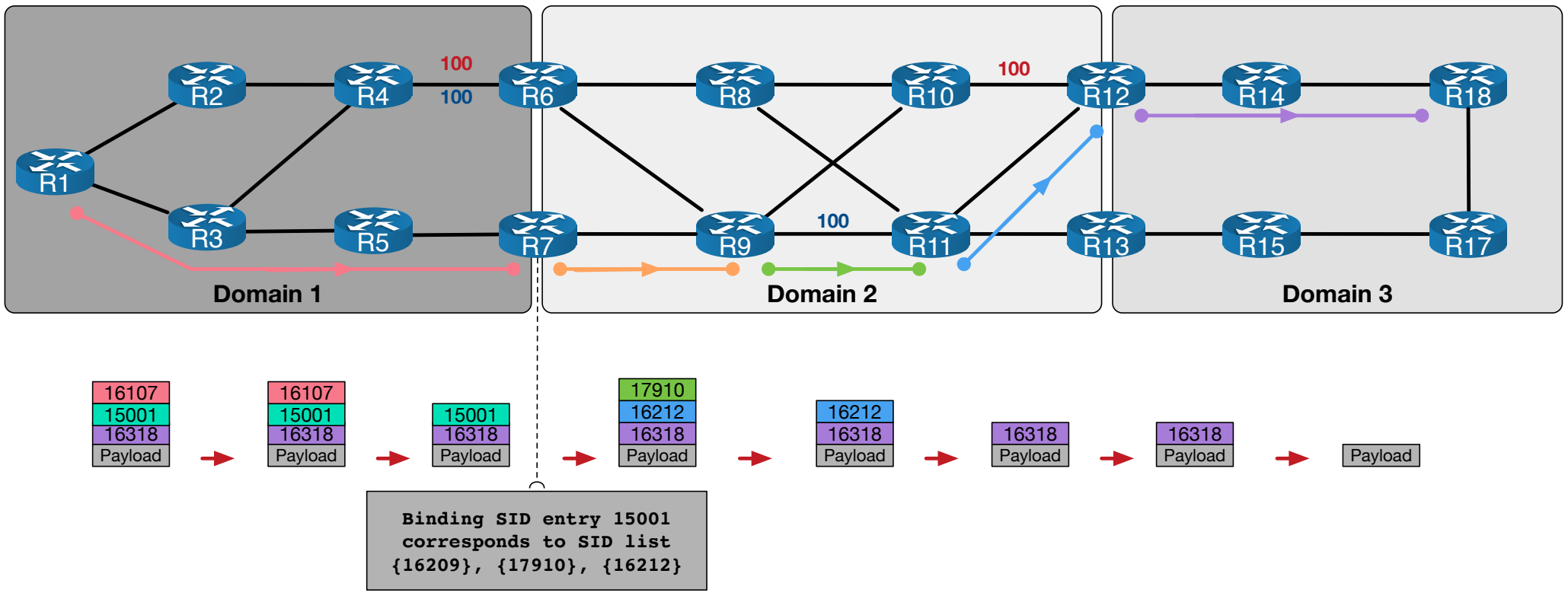
In order to direct traffic from R1 to R18 using the lowest delay metric, the entire path needs to be encoded. Assuming no Flex-Algorithms are used, the SID list would be as follows:

16107 - Node SID for R7 —> sends the packet on the ECMP aware IGP path to R7
16209 - Node SID for R9 —> sends the packet on the ECMP aware IGP path to R9
17910 - Adj SID for R9 to R11 adjacency —> sends the packet from R9 to R11. This is needed since the path with the lowest IGP metric does not match the path with the lowest delay metric.
16212 - Node SID for R12 —> sends the packet on the ECMP aware IGP path to R12
16318 - Node SID for R18 —> sends the packet on the ECMP aware IGP path to R18

How is multi-domain topology information exchanged?
Within an IGP domain, the SR IGP extensions will communicate the various Segment information. But for inter-domain exchanges a different approach will be needed. One option, not shown in these diagrams, is to use eBGP-LS between reflectors, combined with PCEP between the PCE (which will get information from the reflectors) and the Headend.

Multi-domain Policy with Binding SID

Default IGP metric = 10 (variations shown in **blue**)
Default IGP delay = 10 (variations shown in **red**)



With a Binding SID in place, any router in Domain1 can use the Binding SID **15001** to represent the **SR Policy** that is **shortest delay path through Domain 2** from R7. This has a number of advantages.

R1 does not need to know the full SID List. This simplifies signalling and reduces the label stack. It also eliminates the need for R1 to update the SID list if an outage occurs in Domain 2 that prompts an internal SID list change. In short, Domain 2 is opaque from the point of view of Domain 1 and Domain 1 is isolated from any churn.

If R6 and R7 were to use an anycast BSID, R1 could leverage both R6 and R7 using this BSID.