Segment Routing - Flex-Algo

This document gives an example of how SR Flexible Algorithm works based on the below topology.

What is Flex-Algo?
Flexible Algorithms (or Flex-Algos for short) allows you to define different algorithms used to calculate the SPF tree.**The default algorithm is 0 and uses the IGP shortest path. Flex-Algos are represented by a number from 0 - 255. They are defined with their own objectives and set of constraints.**

Routers will configure multiple prefix-SIDs for a given prefix - one for each supported Flex-Algo.

ECMP is supported for each configured Flex-Algo based on the optimisation objective.

A Flex-Algorithm is made up of three parts:

Element	Options in IOS-XR
Calculation Type	SPF (0) or SPF(1). SFP (0) is the standard Djikstra SPF algorithm. SFP(1) or Strict, also uses the same algorithm but doesn't allow local policies to override the SFP-computed path with a different path.
Optimisation Objective	This is used to minimise a particular metric. Options include IGP, TE metric or Link Delay.
Constraint (Optional)	Link resources typically defined with affinity colors or SRLGs.

Installing forwarding entries
MPLS-to-MPLS forwarding entries are installed when a Flex-Algo is configured.
IP-to-MPLS and IP-to-IP forwarding entires are NOT.

By default, unlabelled IP packets are not steering onto Flex-Algo paths - SR-TE steering mechanisms are needed (see Steering onto a Flex-Algo Path *page below)*

Algorithm Numbers Algorithms 0 to 127 are standardised by the IETF.
Algorithms 128-255 are customisable.

An SR enabled node in IOS-XR will always support Algo(0) and Algo(1).Flex-Algo(k) will refer to Flex-Algorithm with number k. Prefix-SID(k) will prefer to a Prefix-SID of Flex-Algo(k)

Advertisement
A Flex-Algo need only be configured on one router that then advertises it throughout the IGP domain. Other routers simply need to receive this advertisement to use the Flex-Algo. For redundancy, multiple nodes can advertise the same Flex-Algo. If these Flex-Algos differ, the one with the highest priority is used. If they have the same priority, System ID (ISIS) or Router ID (OSPF) is used as a tie-breaker.

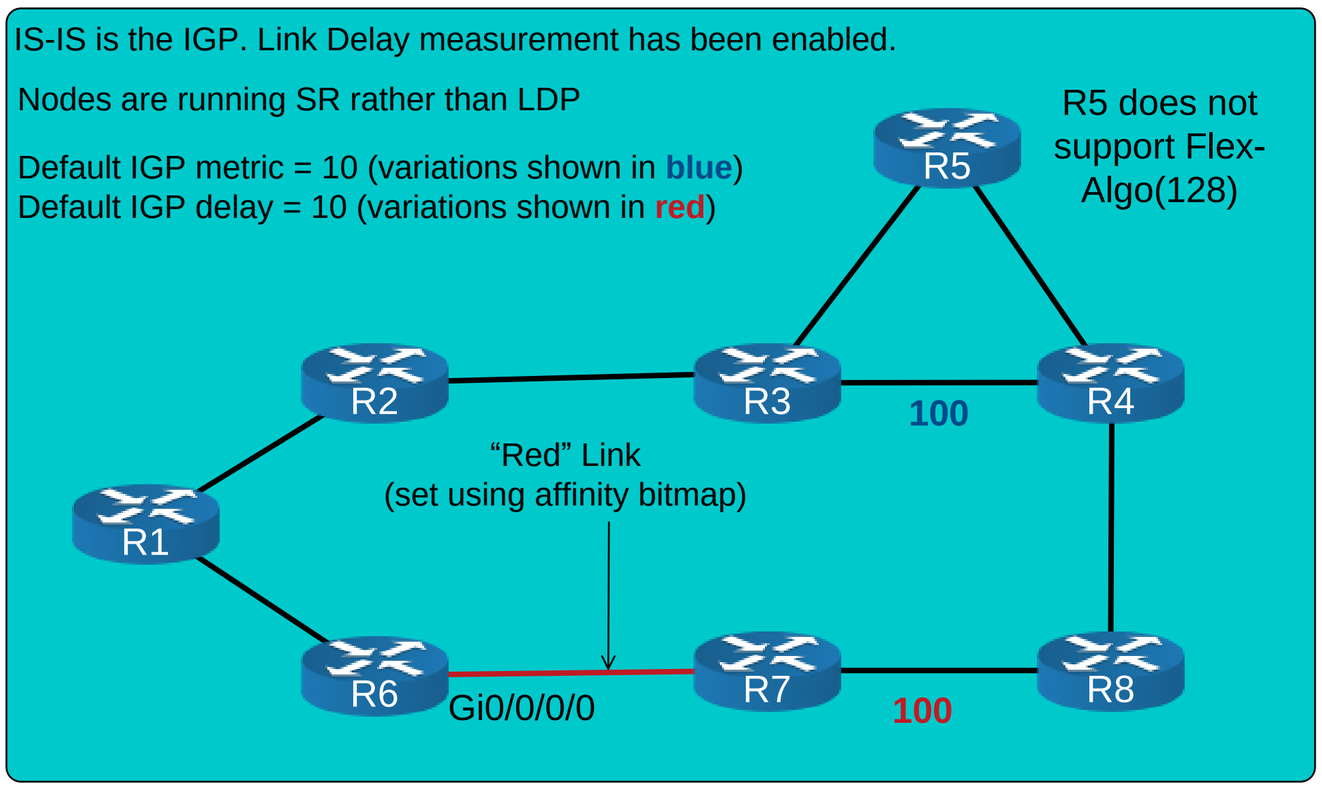
The default priority is 128. The default behaviour by a node is not to advertise a local configuration of a Flex-Algo.

TLVs
The algorithm associated with a Prefix-SID is included in the Prefix-SID TLV when it is advertised by OSPF or IS-IS.

If a Flex-Algo is configured on a node, its support is advertised in the router capability advertisement.

If the Flex-Algo is configured **and** advertisement is enabled, the details of the algorithm itself will be included in the router capability advertisement.

Network Topology



Device	Loopback Address	Flex-Algo(0) SID	Flex-Algo(128) SID	Flex-Algo(129) SID
R1	1.1.1.1/32	16001	16801	16901
R2	2.2.2.2/32	16002	16802	16902
R3	3.3.3.3/32	16003	16803	16903
R4	4.4.4.4/32	16004	16804	16904
R5	5.5.5.5/32	16005	n/a	16905
R6	6.6.6.6/32	16006	16806	16906
R7	7.7.7.7/32	16007	16807	16908
R8	8.8.8.8/32	16008	16808	16908

The Adj SID from one node to another is of the format ***240xy*** where ***x*** is the node and ***y*** is the neighbor. For example P6's Adj-SID from R6 to R7 is 24067.

Flex-Algo(0)
Calculation Type: Algo(0) SPF
Optimisation: IGP Metric
Constraint: None

Flex-Algo(128) Calculation Type:
Algo(0) SPF
Optimisation: Delay Metric
Constraint: None

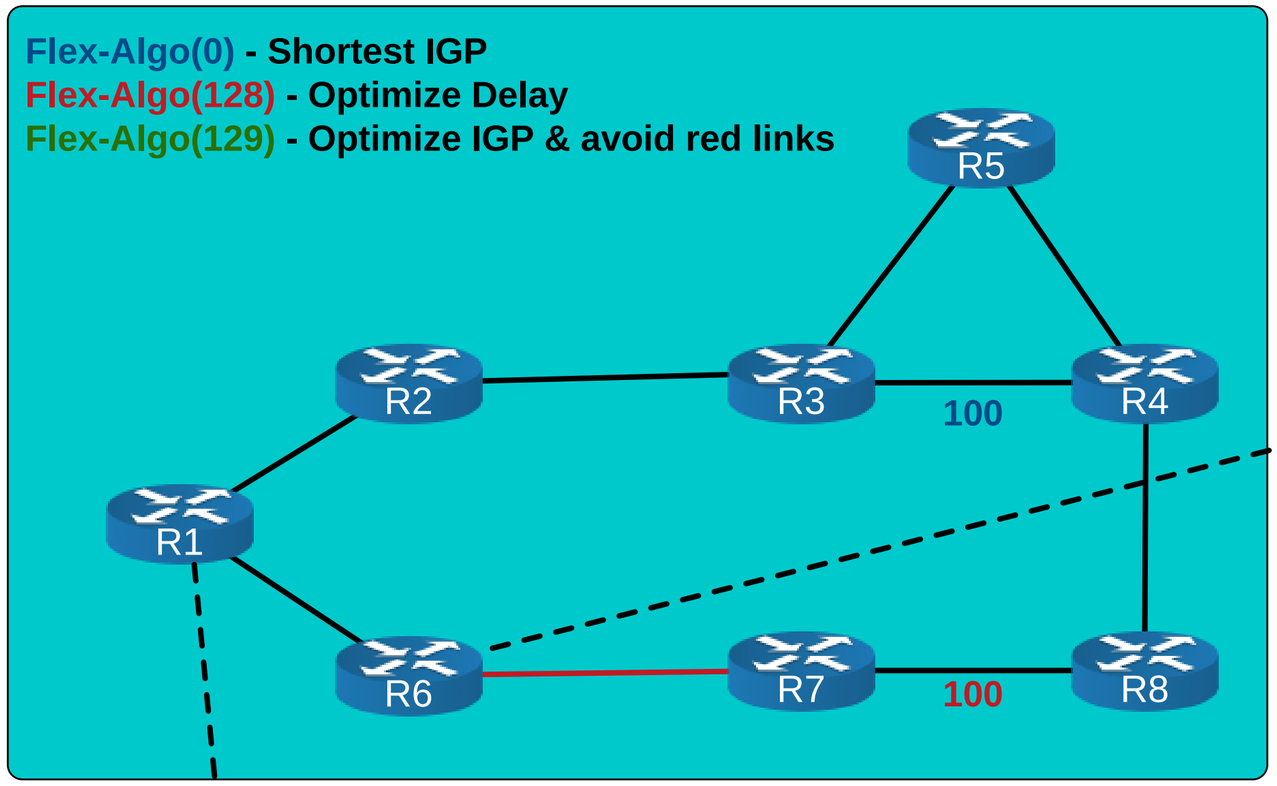
Flex-Algo(129) Calculation Type:
Algo(0) SPF
Optimisation: IGP Metric
Constraint: Avoid RED links

by Steven Crutchley

Segment Routing - Flex-Algo

Configuration

This page shows IOS-XR CLI configuration to setup the policies shown on the previous page from R1s point of view. Each router's loopback 0 is configured with 3 prefix SIDs - one corresponding to each Flex-Algo.




```
segment-routing
 traffic-eng
  affinity-map
   name RED_LINK bit-position 2
  !
  interface Gi0/0/0/0
   affinity name RED_LINK
```

Refers to the position in the affinity bitmap that RED_LINK represents

This name is locally significant

```
interface Loopback0
 ipv4 address 1.1.1.1/32
!
router isis 1
 affinity-map RED bit-position 2
 !
 flex-algo 128
  priority 100
  metric-type delay
  advertise-definition
 !
 flex-algo 129
  priority 100
  advertise-definition
  affinity exclude-any RED
!
 interface Loopback0
  address-family ipv4 unicast
   prefix-sid absolute 16001
   prefix-sid algorithm 128 absolute 16801
   prefix-sid algorithm 129 absolute 16901
```

If different definitions of the same Flex-Algo are defined within an IGP domain, the priority is used to decide which one a router should utilise. The highest priority will take precedence. If multi Flex-Algos exist with the same algorithm number and priority the highest System ID (ISIS) or Router ID (OSPF) is the decider. A locally configured Flex-Algo definition is only considered in this election if it is also advertised.

This enables the advertising of this algorithm's details in the capability advertisements

The metric-type is excluded from this Flex-Algo so it defaults to IGP

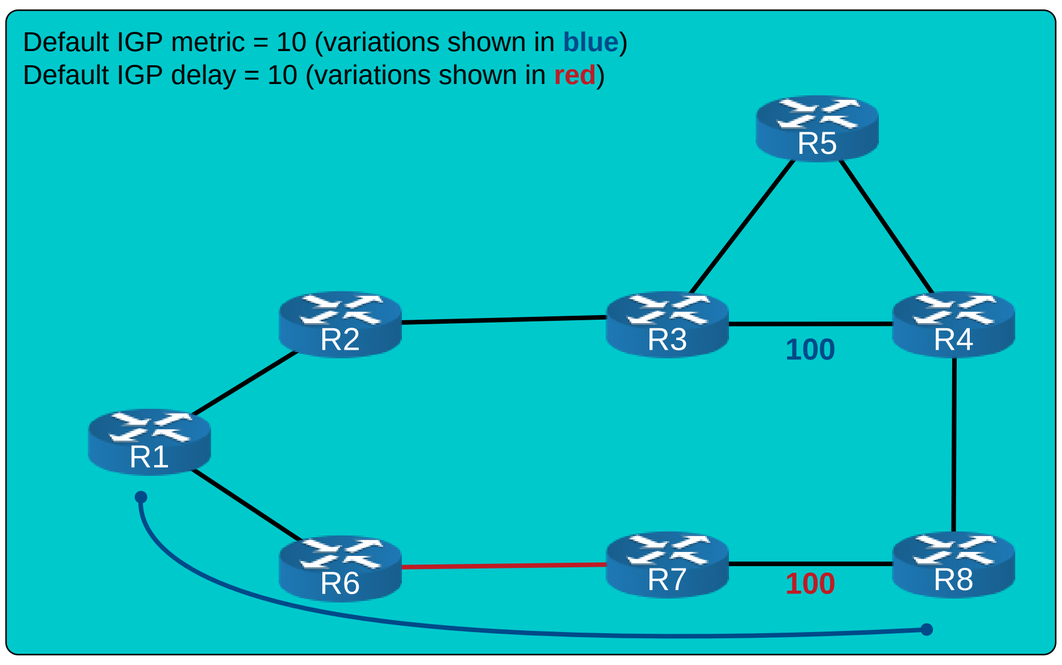
Flex-Algo SIDs are allocated from the SR Global Block (SRGB) shared by all prefix SIDs. They are just as customisable too (setting explicit, removing the node-SID flag etc)

Segment Routing - Flex-Algo

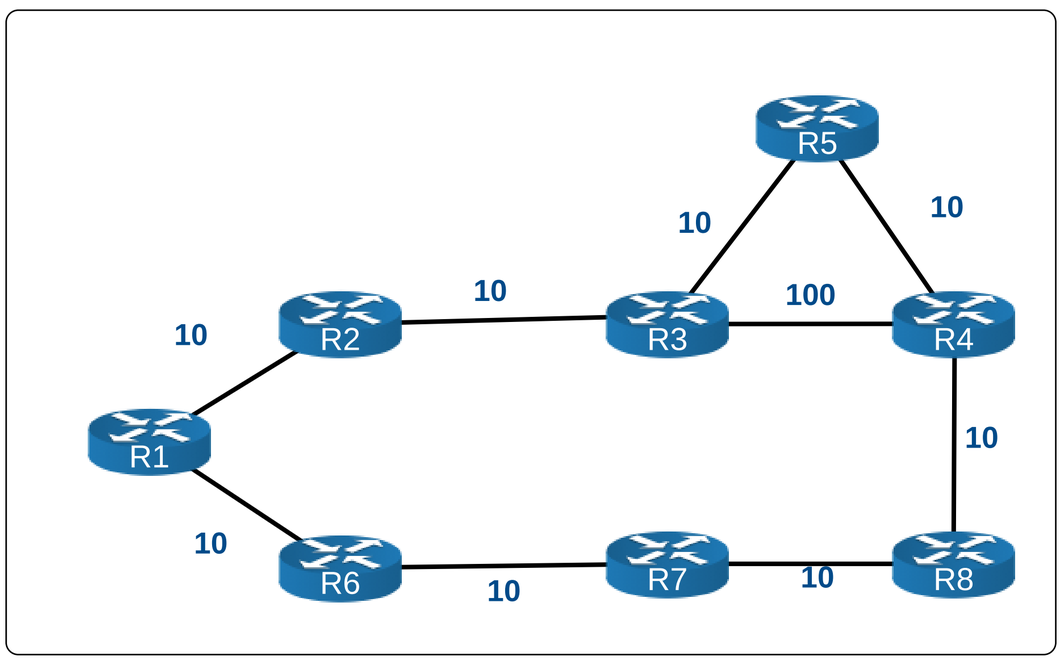
This page examines the traffic flow of packets from R1 to R8 using each of the aforementioned Flex-Algorithms. When a router performs the calculation for a given algorithm, the router removes all the nodes that do not advertise their participation in the topology and any resources that are removed as a result of the constraints defined in the algorithm. For Flex-Algo(k) the resulting topology is referenced as Topo(k)

Flex-Algo (0)
(Smallest IGP Metric)

Traffic Flow



Topo (0)

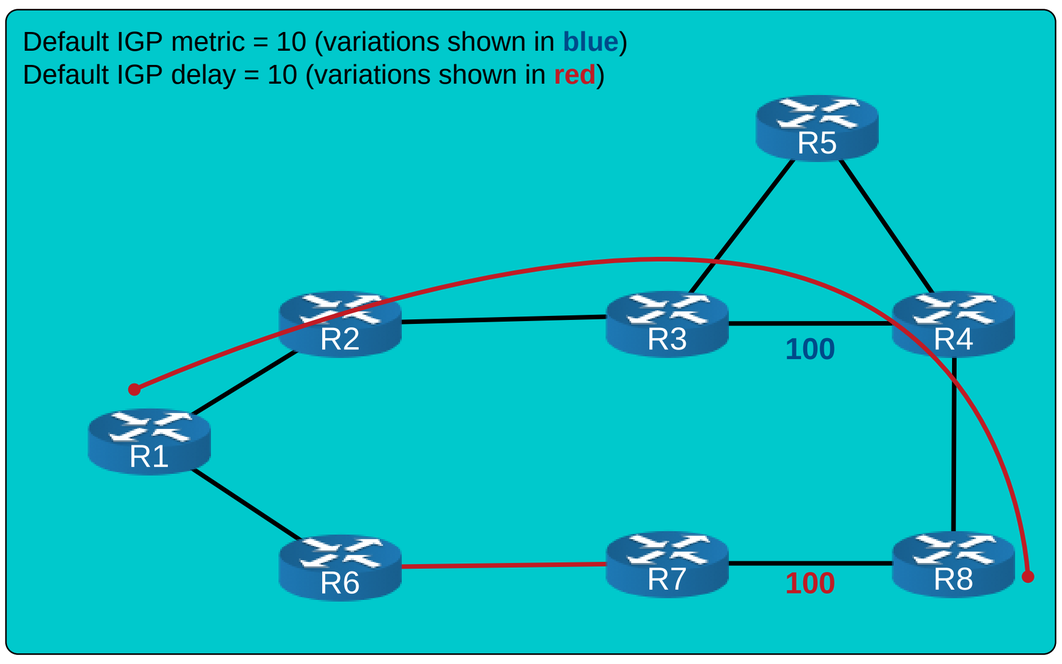


Label Stack:

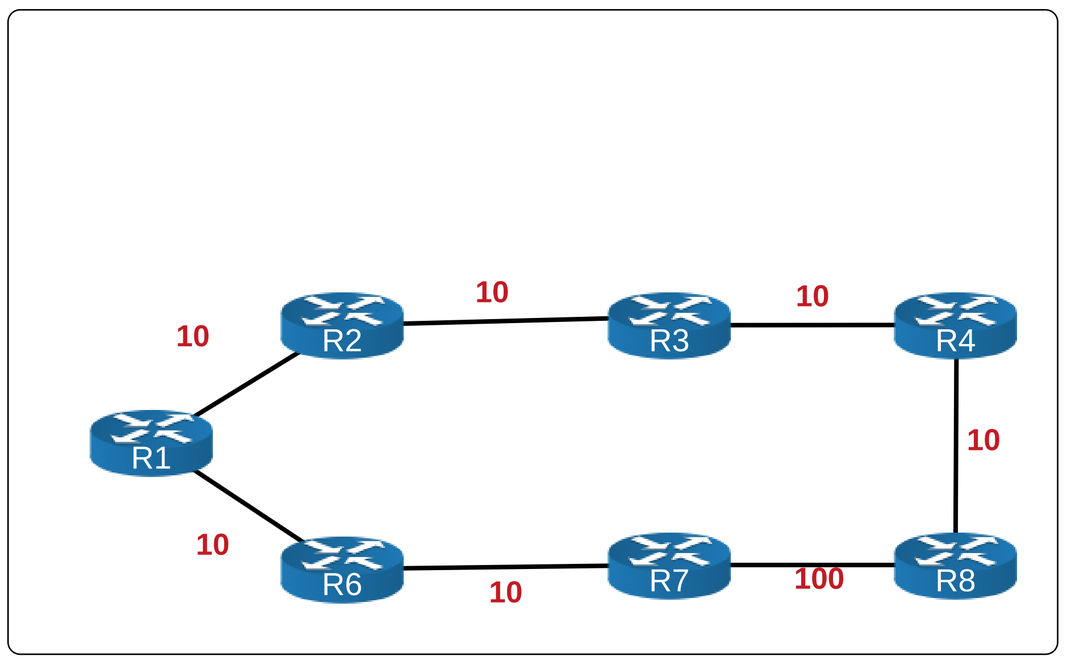
16008	16008	PAYLOAD
PAYLOAD	PAYLOAD	
R1 - R6	**R6 - R7**	**R7-R8 (assuming default PHP)**

Flex-Algo (128)
(Smallest Delay Metric)

Traffic Flow



Topo (128)



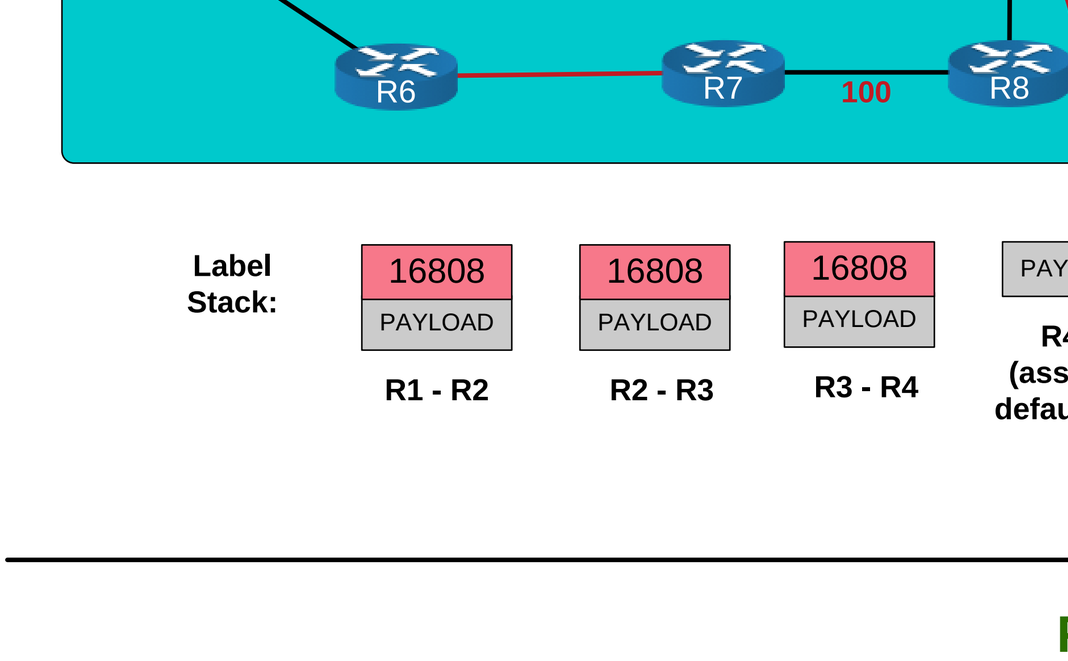
Label Stack:

16808	16808	16808	PAYLOAD
PAYLOAD	PAYLOAD	PAYLOAD	
R1 - R2	**R2 - R3**	**R3 - R4**	**R4-R8 (assuming default PHP)**

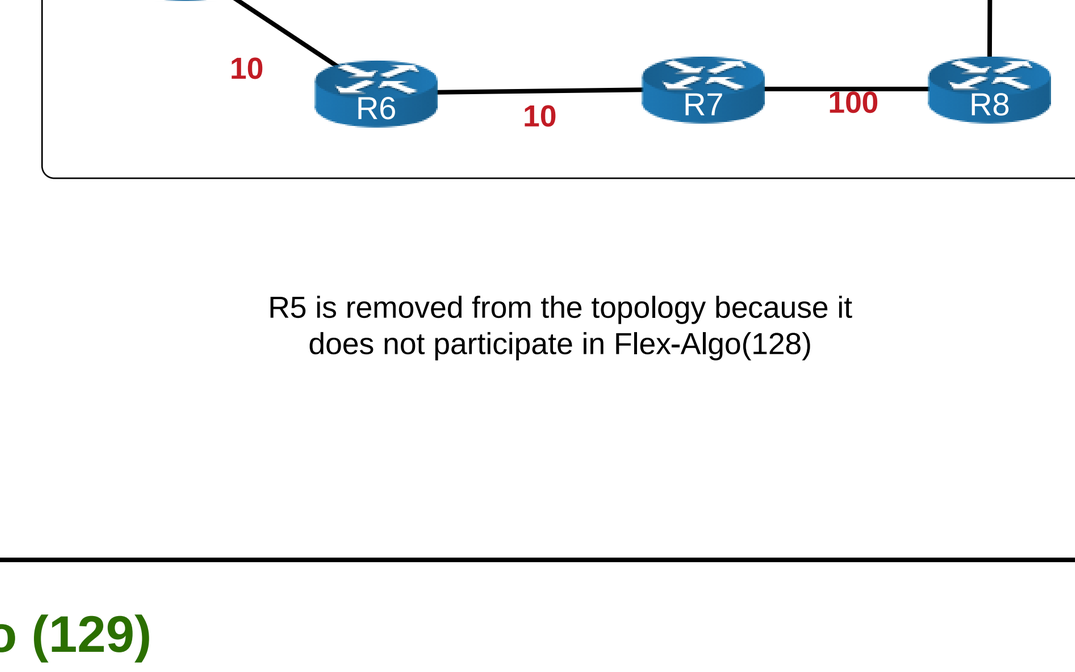
R5 is removed from the topology because it does not participate in Flex-Algo(128)

Flex-Algo (129)
(Smallest IGP Metric - Avoid RED links)

Traffic Flow



Topo (129)



Label Stack:

16908	16908	16908	PAYLOAD
PAYLOAD	PAYLOAD	PAYLOAD	
R1 - R2	**R2 - R3**	**R3 - R4**	**R4-R8 (assuming default PHP)**

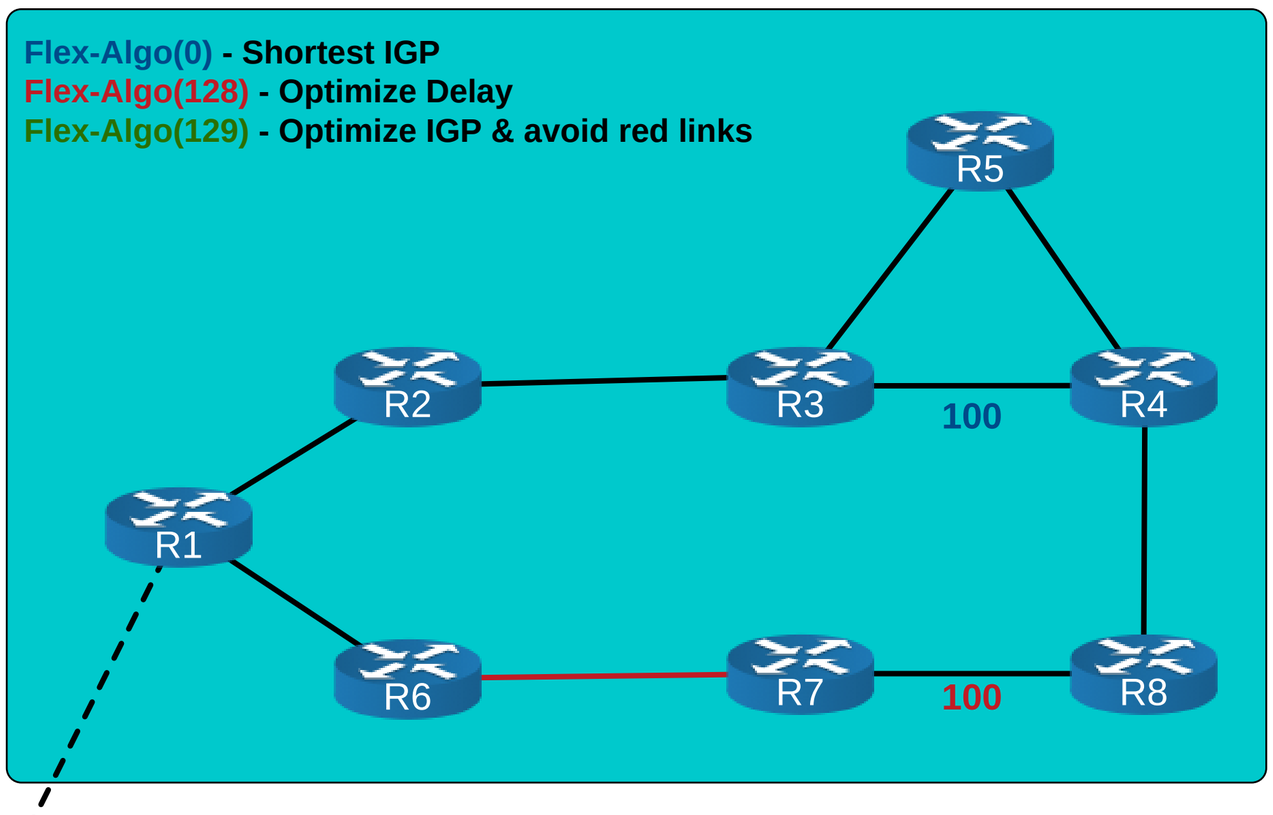
Smaller and more stable SID Lists

If custom Flex-Algos were not used and IGP shortest path (Flex-Algo(0)) was the only option, the SID list to accomplish the same goal would be as follows: <16005, 16008> . 16005 gets the packet to R5 via the shortest IGP path. 16008 get the packet to R8. Also, if the R3 - R5 link failed, this SID list would need updating.

With Flex-Algo(129) in place, only one label is needed and in the event of R3-R5 failing, the algorithm would update itself and converge around the change but the SID list would remain the same.

Segment Routing - Flex-Algo

Output




```
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      ATT/P/OL
R1.00-00            * 0x0000021b   0x21de        1198              0/0/0
<snip>
  Metric: 0 IP-Extended 1.1.1.1/32
    Prefix-SID Index: 1, Algorithm:0 , R:0 N:1 P:0 E:0 V:0 L:0
    Prefix-SID Index: 801, Algorithm:128 , R:0 N:1 P:0 E:0 V:0 L:0
    Prefix-SID Index: 901, Algorithm:129 , R:0 N:1 P:0 E:0 V:0 L:0
    Prefix Attribute Flags: X:0 R:0 N:1
    Source Router ID: 1.1.1.1
<snip>
```

Flex-Algo SIDs are advertised using SR TLVs (or opaque LSA) extensions as normal.

```
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      ATT/P/OL
R1.00-00            * 0x0000013e   0x20ee        1198              0/0/0
<snip>
  Hostname: R1
    Router Cap: 1.1.1.1, D:0, S:0
    Segment Routing: I:1 V:0, SRGB Base: 16000 Range: 8000
    SR Algorithm:
      Algorithm: 0
      Algorithm: 1
      Algorithm: 128
      Algorithm: 129
    Node Maximum SID Depth:
      Label Imposition: 10
    Router ID:  1.1.1.1
    Flex-Algo Definition:
      Algorithm: 128 Metric-Type: 1 Alg-type: 0 Priority: 100
    Flex-Algo Definition:
      Algorithm: 129 Metric-Type: 0 Alg-type: 0 Priority: 100
      Flex-Algo Exclude Ext Admin Group:
        0x00000004
<snip>

RP/0/0/CPU0:R1# show isis database verbose R6

IS-IS 1 (Level-2) Link State Database

LSPID                 LSP Seq Num  LSP Checksum  LSP Holdtime      ATT/P/OL
R6.00-00            * 0x0000539c   0xe81c        736               0/0/0
<snip>
  Metric: 10 IS-Extended R7.00
    Interface IP Address: 10.6.7.6
    Neighbor IP Address: 10.6.7.7
    Link Average Delay: 10 us
    Link Min/Max Delay: 10/10 us
    Link Delay Variation: 0 us
    Application Specific Link Attributes:
      L flag: 0, SA-Length 1, UDA-Length 0
      Standard Applications: FLEX-ALGO
      Ext Admin Group:
        0x00000004
    Link Maximum SID Depth:
      Label Imposition: 10
ADJ-SID: F:0 B:0 V:1 L:1 S:0 P:0 weight:0 Adjacency-sid:24067
<...snip...>
```

Advertisement of support for Flex-Algorithms
IS-IS and OSPF will include the Algorithms it supports in its SR-Algorithm Sub-TLV attached to either an LSP or Opaque Type 4 LSA respectively.

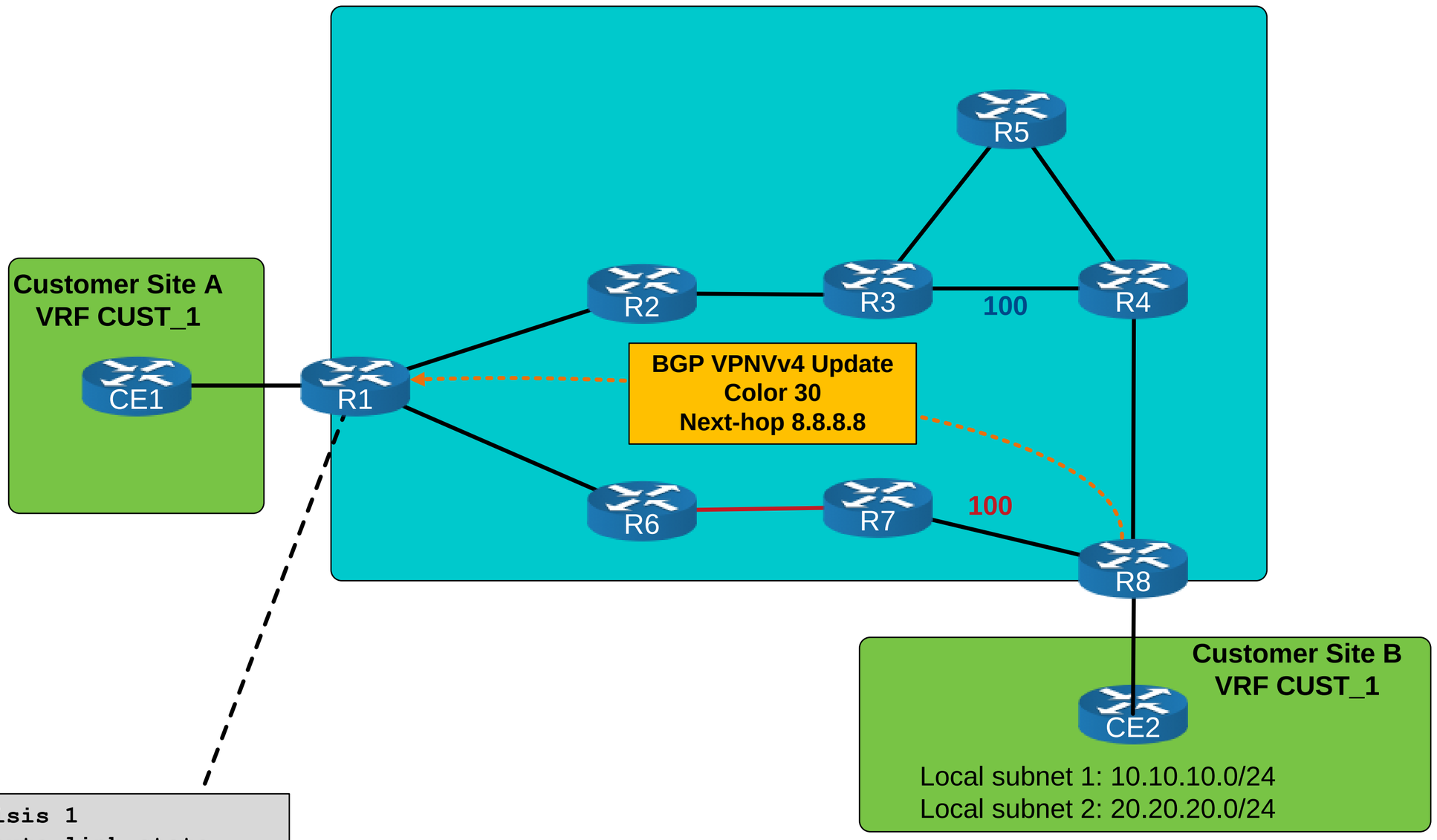
Advertisement of detail of Flex-Algorithms themselves
Metric-Type is the Optimisation Metric
(0 = IGP, 1 = Delay, 2 = TE).
Alg-Type is the Calculation Type
(0 = SFP(0), 1 = SFP(1))
Exclude Ext Admin Group specifies the constraint to avoid links will affinity bit 2 set (0x4 = 100 in binary) which is how RED links have been defined in this example.

This shows how R6 advertises its link to R7 with the affinity bitmap set at index 2 (0x4 = 100 in binary, where the right most binary digit is index 0)

Segment Routing - Flex-Algo

Steering onto a Flex-Algo Path

By default, unlabelled packets will not be steered onto a Flex-Algo path. To do this you will need to use SR-TE steering mechanisms like ODN or AS. The below example shows how you would do it using ODN. The SID Algorithm indentifier is configured as a constraint under the path configuration.



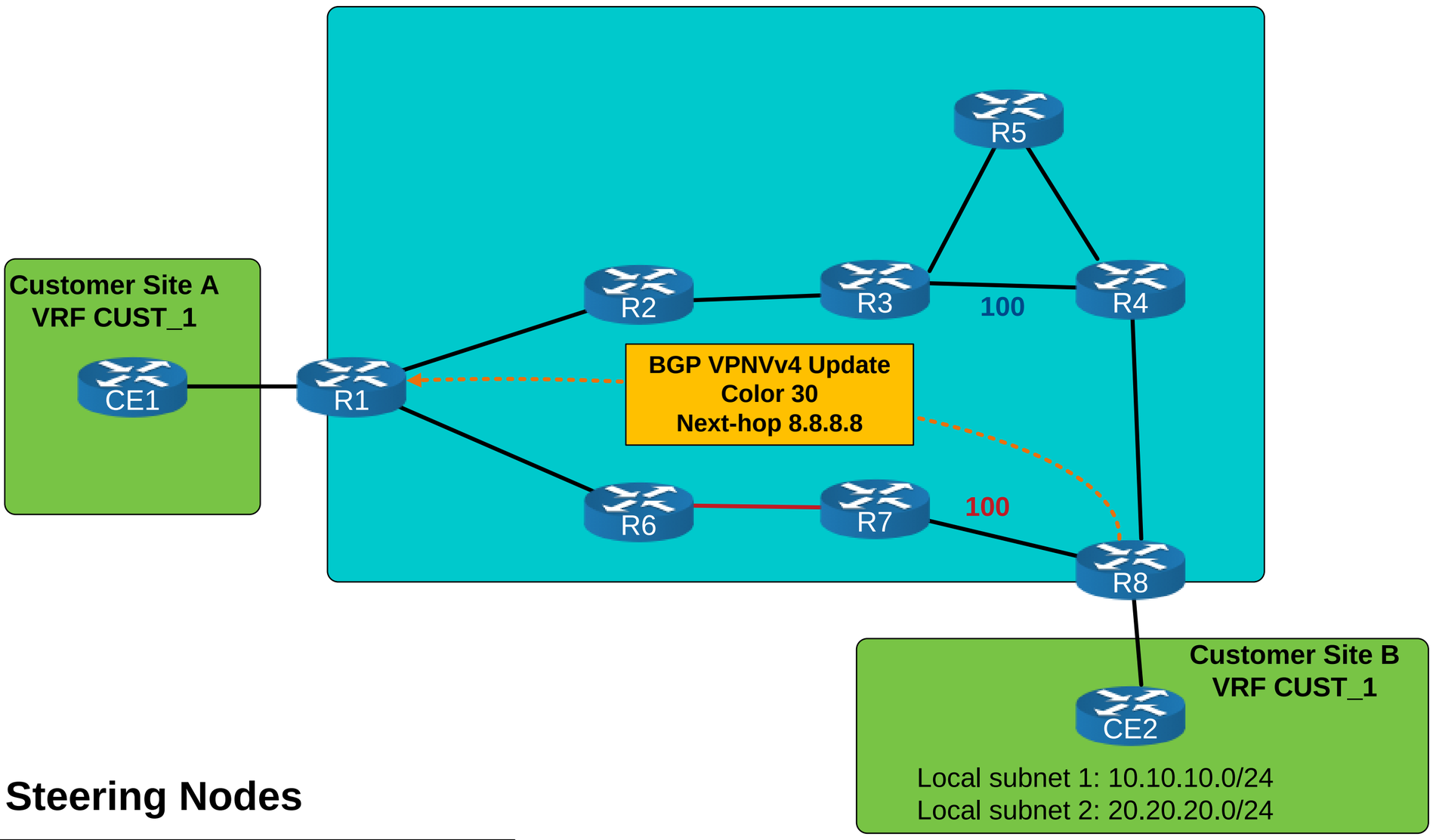

```
router isis 1
 distribute link-state
 <other config omitted>

segment-routing
 traffic-eng
  on-demand color 30
   dynamic
   !
   constraints
    segments
     sid-algorithm 128
```

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

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

Color: 30, End-point: 8.8.8.8
  Name: srte_c_30_ep_8.8.8.8
  Status:
    Admin: up Operational: up for 00:14:02 (since Aug 4 12:21:52.332)
  Candidate-paths:
    Preference: 200 (BGP ODN) (active)
      Requested BSID: dynamic
      Constraints:
        Prefix-SID Algorithm: 128
      Dynamic (valid)
        16808 [Prefix-SID: 8.8.8.8, Algorithm: 128]
    Preference: 100 (BGP ODN)
      Requested BSID: dynamic
      Dynamic (pce) (invalid)
        Metric Type: NONE, Path Accumulated Metric: 0
  Attributes:
    Binding SID: 40001
    Forward Class: 0
    Steering BGP disabled: no
    IPv6 caps enable: yes
```

Segment Routing - Flex-Algo

Requirements

In order for devices to participate in Flex-Algo they must have a minimum configuration. This shows the needed configuration to participate in Flex-Algo for different types of nodes.



Steering Nodes

```
router isis 1
 distribute link-state
 address-family ipv4 unicast
  mpls traffic-eng level-2-only
  mpls traffic-eng router-id Loopback0
 !
 flex-algo 128
 !
 interface Loopback0
  address-family ipv4 unicast
   prefix-sid index 105
   prefix-sid algorithm 128 index 16801
```

These nodes steer the traffic into the Flex-Algo instance - R1 in the above diagram.

Distribute link-state is needed to feed the SR-TE Database and perform the FlexAlgo calculations.

MPLS Traffic Engineering must be enabled (router-id is not strictly needed by recommended)

Referencing flex-algo 128 is needed to allowed R1 to participate in the Flex-Algo instance.

Nodes that define the algorithm

```
router isis 1
affinity-map RED_LINK bit-position 2
 address-family ipv4 unicast
 !
 flex-algo 128
  priority 210
  advertise-definition
  prefix-metric
  affinity exclude-any RED_LINK
  address-family ipv4 unicast
  !
 !
 interface Loopback0
  address-family ipv4 unicast
   prefix-sid index 2
   prefix-sid algorithm 128 index 1002
  !
```

Any node in the network can define the Flex-Algo instance

Other than the flex-algo itself, any affinity maps or srlg group must be defined locally.

Again the SID index isn't strictly needed of it is not used in any SID list for the given Flex-Algo but its configuration is recommended

Error messages

If `distribute link-state` is missing, an error message stating that a *source node cannot be found* will show when looking at the segment routing policy. If traffic-engineering, SID indexes or flex-algo references are missing, the output will show *Path Not Found*.

Other Nodes

```
router isis 1
 address-family ipv4 unicast
 !
 flex-algo 128
 !
 interface Loopback0
  address-family ipv4 unicast
   prefix-sid index 105
   prefix-sid algorithm 128 index 16802
```

Other nodes includes any transit/P or even the egress node that applies the color to the prefixes – R8 for example

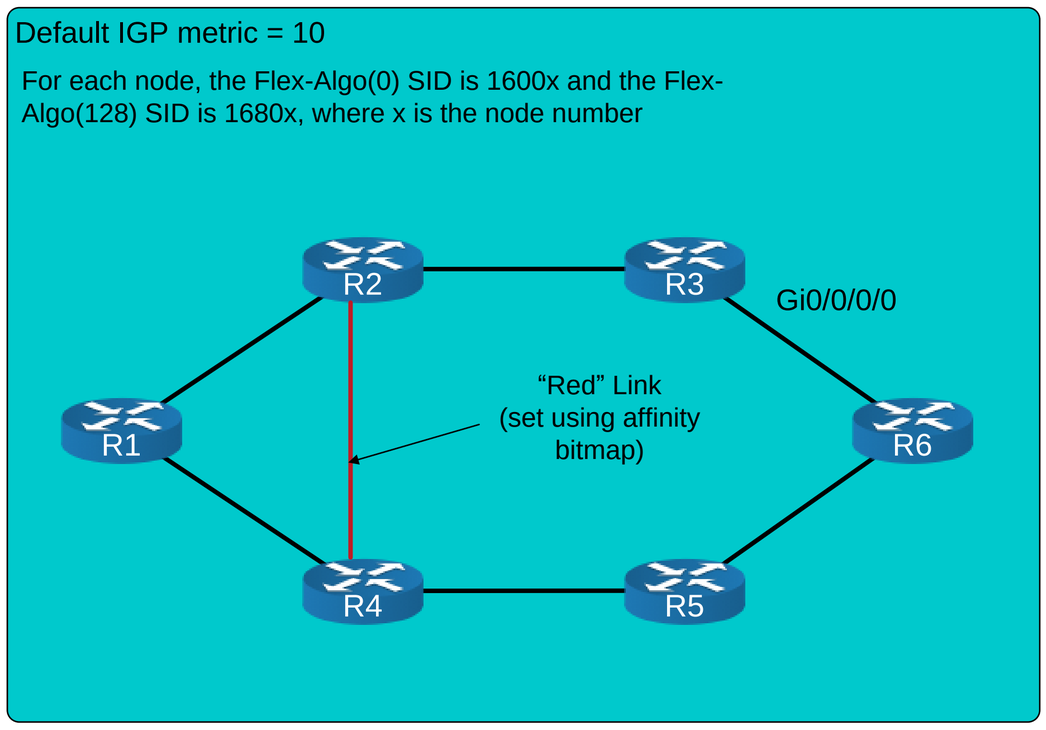
A reference to the flex-algo is needed.

The SID index isn't strictly needed of it is not used in any SID list for the given Flex-Algo – however if it is an egress node, like R8, then it will definitely be required.

Segment Routing - Flex-Algo

TI-LFA

If TI-LFA is configured on a node, then the node will calculate the TI-LFA paths for each of the Flex-Algos that is supports, using a appropriate path optimisations and constraints.

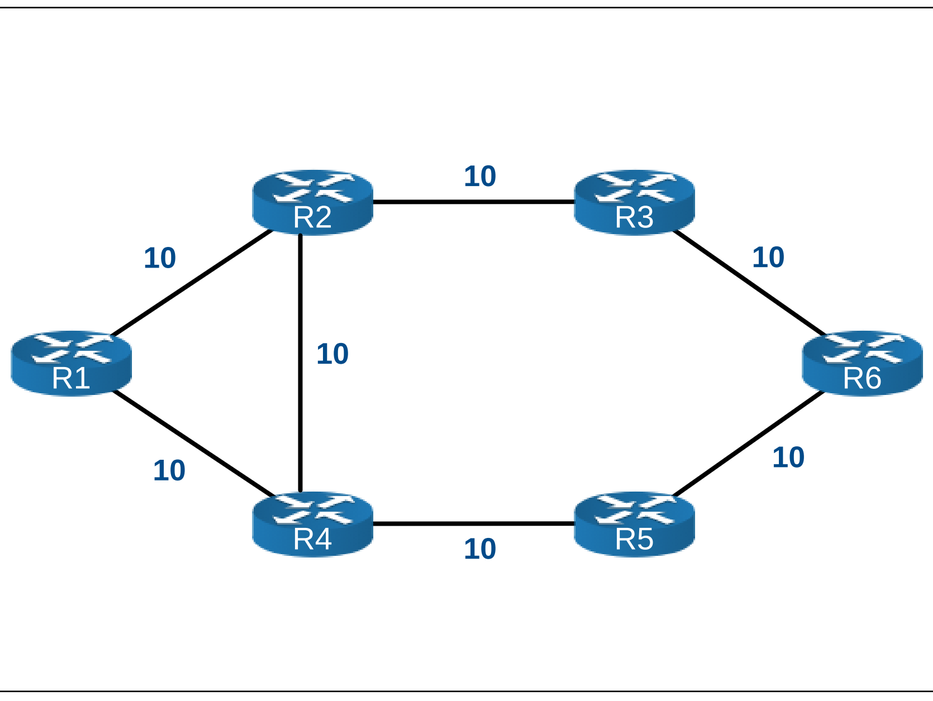


In this example the Flex-Algos are defined as follows:
Flex-Algo(0) - Shortest IGP
Flex-Algo(128) - Optimize IGP & avoid red links

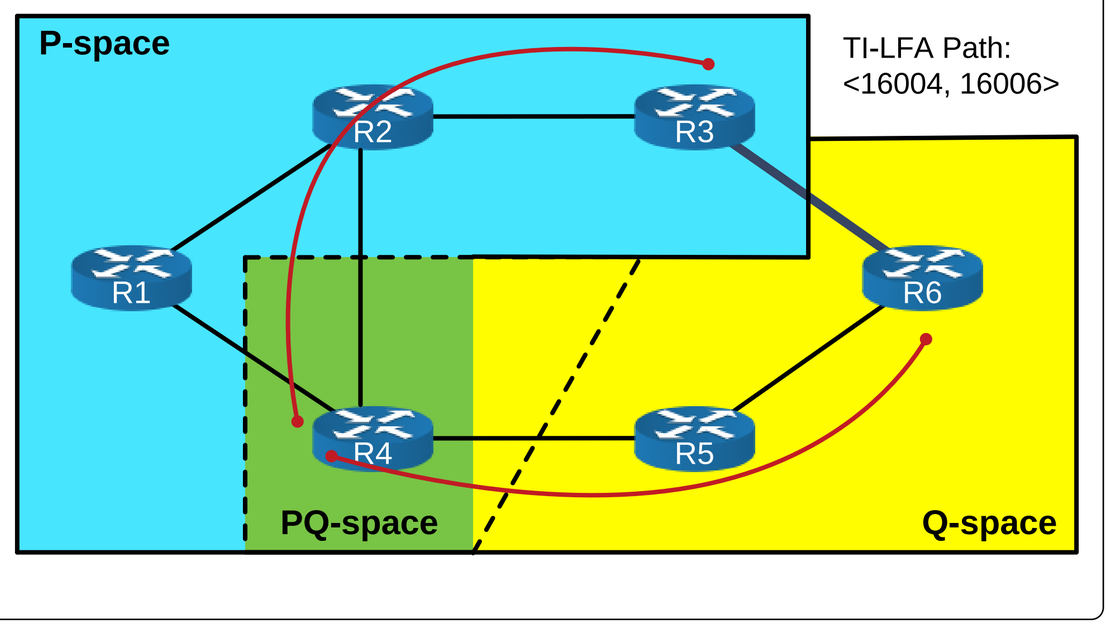
```
interface Loopback0
 ipv4 address 1.1.1.1/32
!
router isis 1
 affinity-map RED bit-position 2
 !
 flex-algo 128
  priority 100
  affinity exclude-any RED
  advertise-definition
!
interface Gi0/0/0/0
 point-to-point
 address-family ipv4 unicast
  fast-reroute per-prefix
  fast-reroute per-prefix ti-lfa
```

Flex-Algo (0)
(Smallest IGP Metric)

Topo (0)

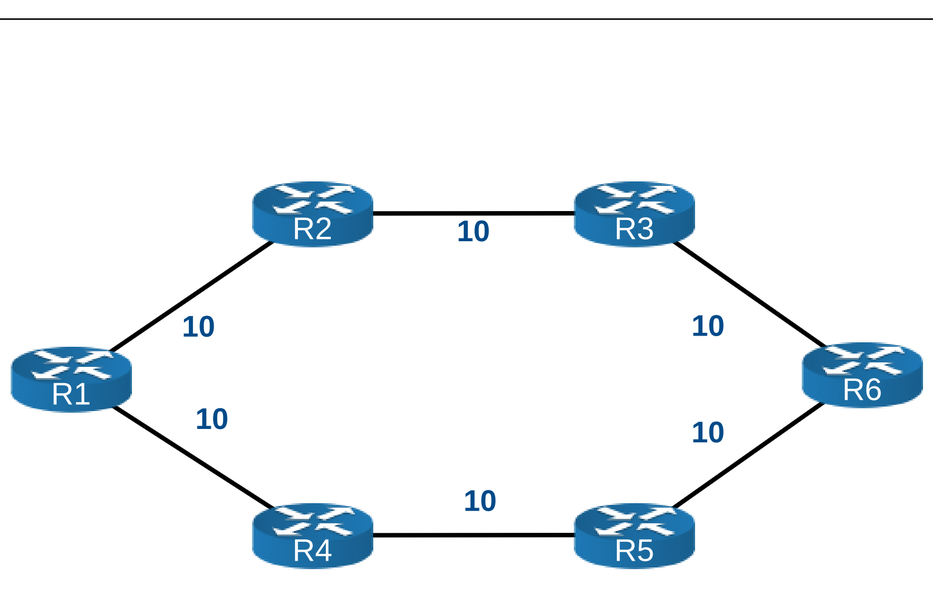


R3 Backup Path Calculation for Gi00/0/0



Flex-Algo (128)
(Smallest IGP Metric avoiding RED links)

Topo (0)



R3 Backup Path Calculation for Gi00/0/0