



Unleashing SR Traffic Engineering capabilities with SR Flexible Algorithm

Jose Liste

January 2019

Agenda

- Overview
- Operation
- Use Cases
- Demonstration

Why are we here?

- Have you ever wished you could tailor IGP computation to fit your own traffic engineering needs? Such as by computing:
 - Paths considering a subset of the routers in your network?
 - Paths that minimize cumulative delay to a destination based on measured per-link delay?
 - Paths traversing only encrypted MACsec links? Both for primary and pre-computed backup paths !!!
 - Paths traversing only high speed interfaces
- All the above use cases are now possible. And furthermore, they are possible at scale and based on the source routing paradigm
- Meet SR Flexible Algorithms !!!

Flexible Algorithm Overview

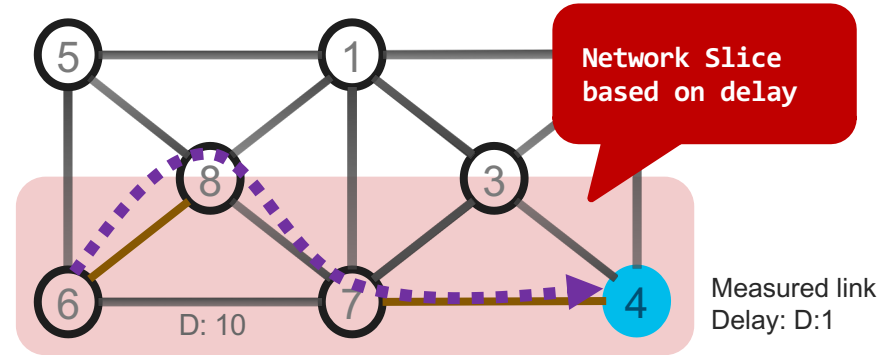
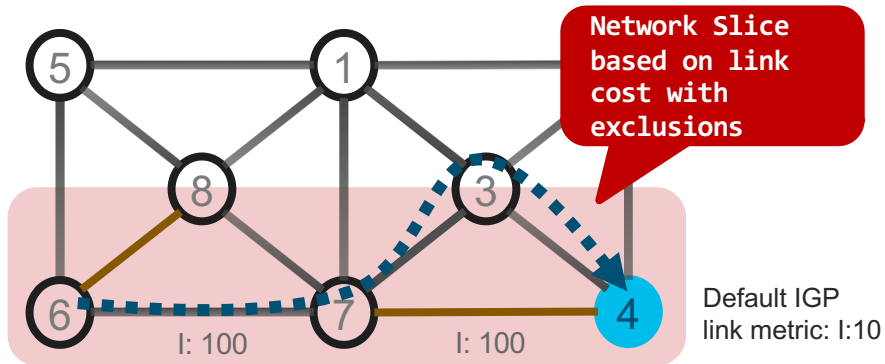
Flexible Algorithm

- We call “Flex-Algo”
 - The algorithm is defined by the operator, on a per-deployment basis
- Flex-Algo K is defined as
 - The minimization of a specified metric: IGP, delay, ...
 - The exclusion of certain link properties: link-affinity, SRLG, ...

SR IGP Flexible Algorithms

- Examples

- Operator defines Flex-Algo 128 as “minimize IGP metric and avoid link-affinity “brown”
- Operator defines Flex-Algo 129 as “minimize delay metric”



Flexible Algorithm Operation

IGP SR Algorithm

- Each Prefix SID is related to an algorithm

0										1										2										3																			
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1																		
+										+										+										+																			
Type										Length										Flags										Algorithm																			
+										+										+										+																			
										SID/Index/Label (variable)																																							
+										+										+										+																			

- Each node advertises its ALGO capability

The SR-Algorithm sub-TLV has following format:

0	1	2	3
0 1 2 3 4 5 6 7 8 9	0 1 2 3 4 5 6 7 8 9	0 1 2 3 4 5 6 7 8 9	0 1 2 3 4 5 6 7 8 9 0 1
+	+	+	+
Type	Length		
+	+	+	+
	Algorithm 1	Algorithm 2	Algorithm ... Algorithm n
+	+	+	+

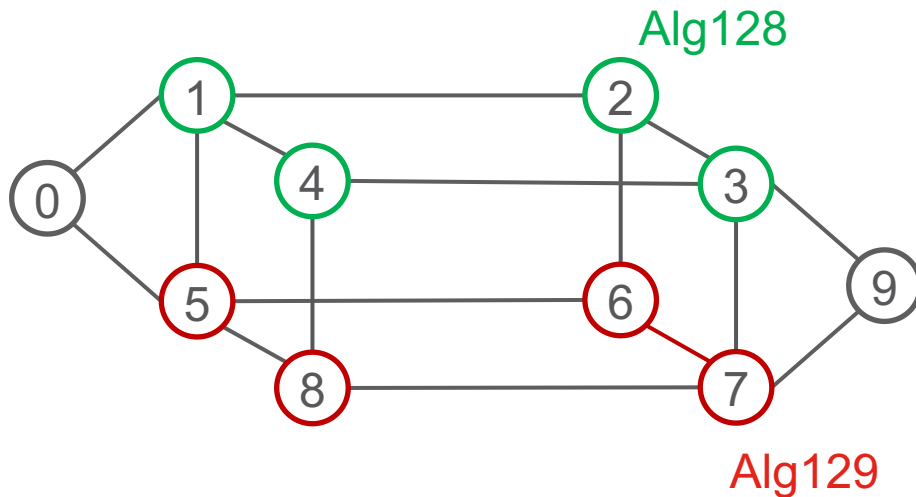
Flex-Algo Participation Advertisement

- Each node MUST advertise Flex-Algo(s) that it is participating in

Nodes 0 and 9 participate to Algo 0 and 128 and 129

Nodes 1/2/3/4 participate to Algo 0 and 128

Nodes 5/6/7/8 participate to Algo 0 and 129



Prefix-SID for each Flex-Algo

- If a node advertises participation in a Flex-Algo likely it also advertises a prefix SID for that Flex-Algo

Node 9 advertises

Prefix SID 16009 for ALGO 0

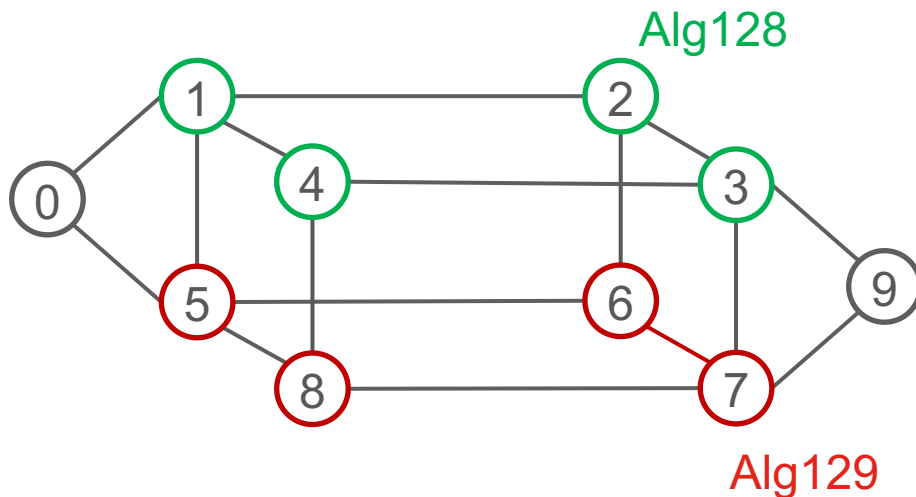
Prefix SID 16809 for ALGO 128

Prefix SID 16909 for ALGO 129

Node 2 advertises

Prefix SID 16002 for ALGO 0

Prefix SID 16802 for ALGO 128



No additional loopback address

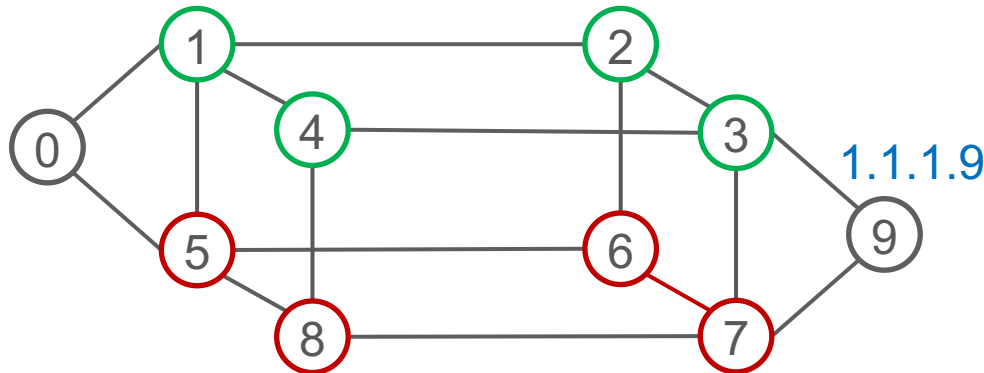
- Flex-Algo Prefix SID's can be advertised as additional prefix-SID's of the existing loopback address

Node 9 advertises loopback0 [1.1.1.9/32](#) with

Prefix SID 16009 for ALGO 0

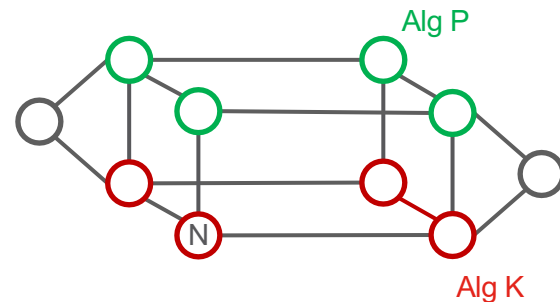
Prefix SID 16809 for ALGO 128

Prefix SID 16909 for ALGO 129



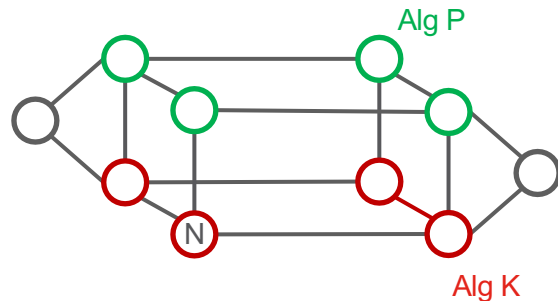
Flex Algo Operation

- Flex-Algo Membership
 - Node N is enabled to participate in Flex-Algo(s) instance K
- Flex-Algo Definition
 - Node N has a consistent definition of K



Flex Algo Operation (cont.)

- Flex-Algo Computation
 - N prunes any node not a member of K
 - N prunes any link that is excluded by K
 - Resulting topology is called Topo(K)
 - N compute shortest-path tree on Topo(K) with metric defined by K
- Flex-Algo Prefix SID installation
 - N installs any reachable Prefix-SID of K in the forwarding table



Flex- Algo Definition

- Each node MUST have the definition of the Flex- Algo(s) that it is participating in
 - e.g. ALGO 128: minimize on IGP metric and avoid TE affinity RED
- Local configuration
 - likely automated via a solution such as NSO
- Learned from a central entity via ISIS flooding
 - new top TLV defined for Flex- Algo definition advertisement

Flex-Algo Computation and Prefix-SID installation

- A node N computes Flex-Algo K if
 - it is enabled for K, and
 - it has a consistent definition for K
- If so, the first step is to define the topology of K
 - N prunes any node that is not advertising participation to K
 - N prunes any link that is excluded by the algorithm of K
 - e.g. if K excludes TE-affinity RED then any link with TE-affinity RED is pruned
 - The resulting topology is called $\text{Topo}(K)$

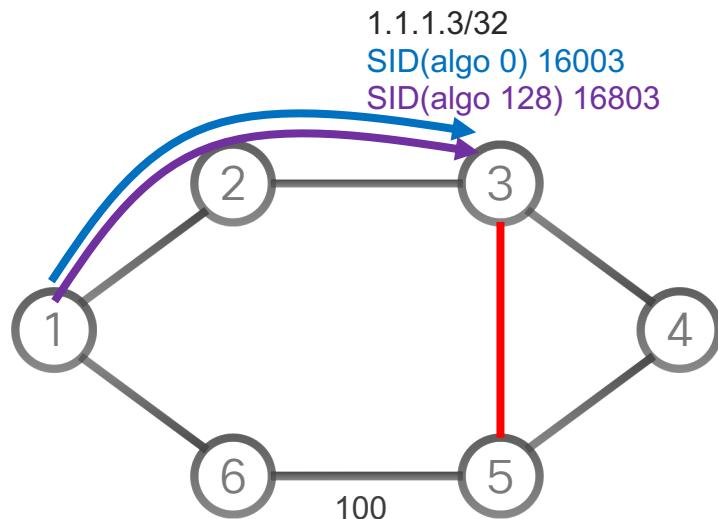
Flex-Algo Computation and Prefix-SID installation

- The second step is to compute shortest-path tree on $\text{Topo}(K)$ with the metric defined by K
 - it could be the IGP metric, the TE metric or the delay
- The third step is to install any reachable Prefix-SID of Flex-Algo K in the forwarding table

TI-LFA

- The TI-LFA algorithm is performed within Topo(K)
- The backup path is expressed with Prefix-SID's of Algo K
- Benefits: the backup path is optimized per Flex-Algo!

Example – Primary paths per Algo



Each node in this topology supports SR alg0, alg128
Default IGP link metric: 1:10

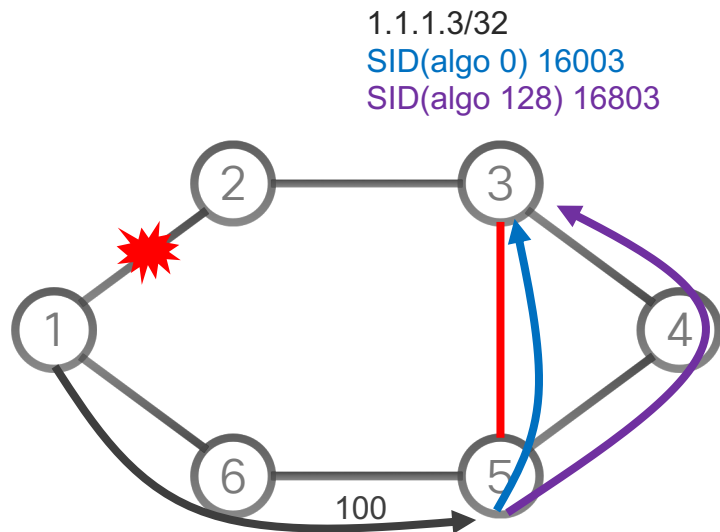
At node 1 for destination 3

16003 => 16003 via 2

16803 => 16803 via 2

All nodes participate to Algo 128
which is defined as min IGP metric
and avoid red affinity

Example – TI-LFA Backup path per Algo



At node 1 for destination 3

16003 => 16003 via 2

Backup: <24065, **16003**> via 6

16803 => 16803 via 2

Backup: <24065, **16803**> via 6

The usage of Algo-128 Prefix-SID 16803 ensures that the Algo 128 backup path also avoids the red link

Reminder: 240XY is the Adj SID from node X to node Y

OSPF and SRv6

- Same applies to OSPF
- Same applies to SRv6

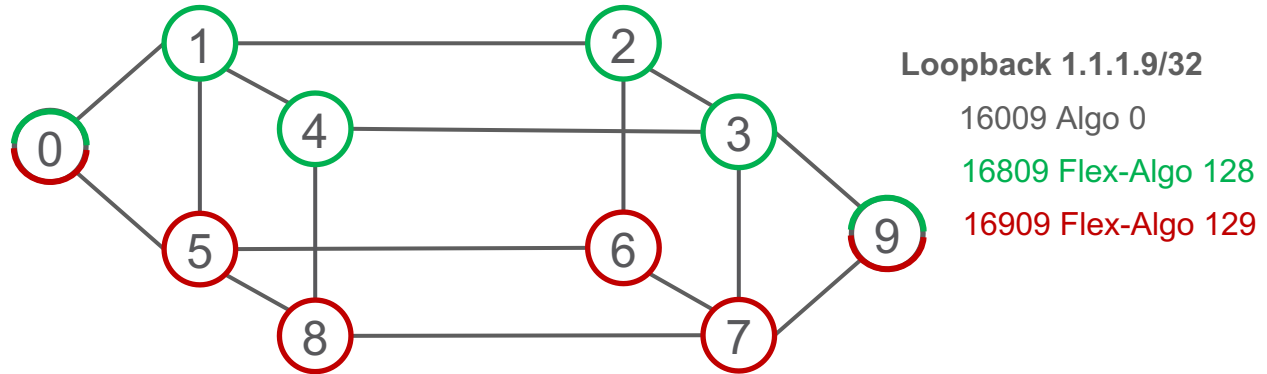
IETF

- draft-ietf-spring-segment-routing
 - Prefix-SID per Algorithm
- draft-ietf-spring-segment-routing-policy
 - SR Policy architecture, ODN, AS
- draft-ietf-lsr-flex-algo
 - Customization of Algo and consistency
- draft-ietf-isis-te-app
 - Used to flood Flex-Algo specific link affinities
- RFC7810 (IS-IS Traffic Engineering (TE) Metric Extensions)
 - Used to advertise extended TE metrics – e.g. link delay

Flexible Algorithm Use-Cases

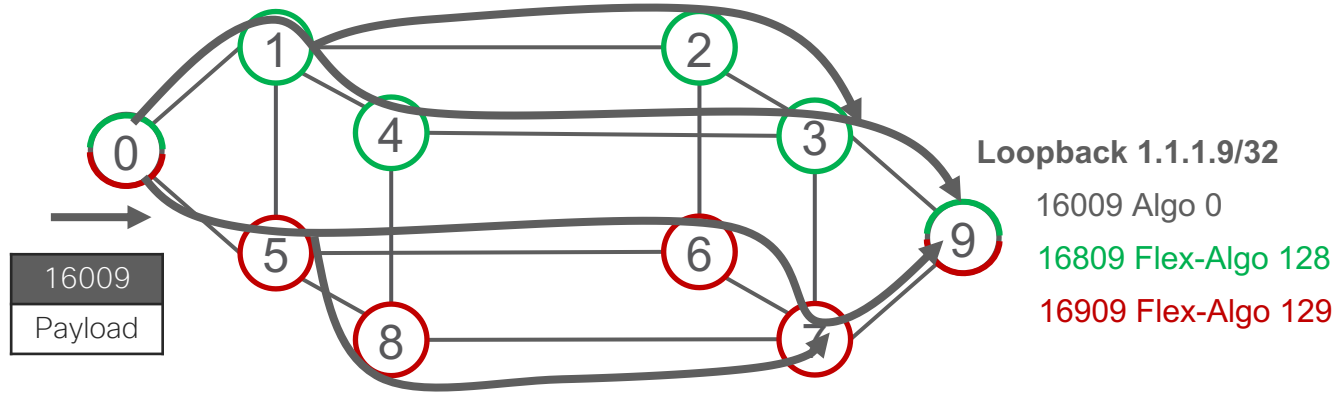
Use-Case – Multi-Plane Networks

- Powered by SR IGP Flex Algo



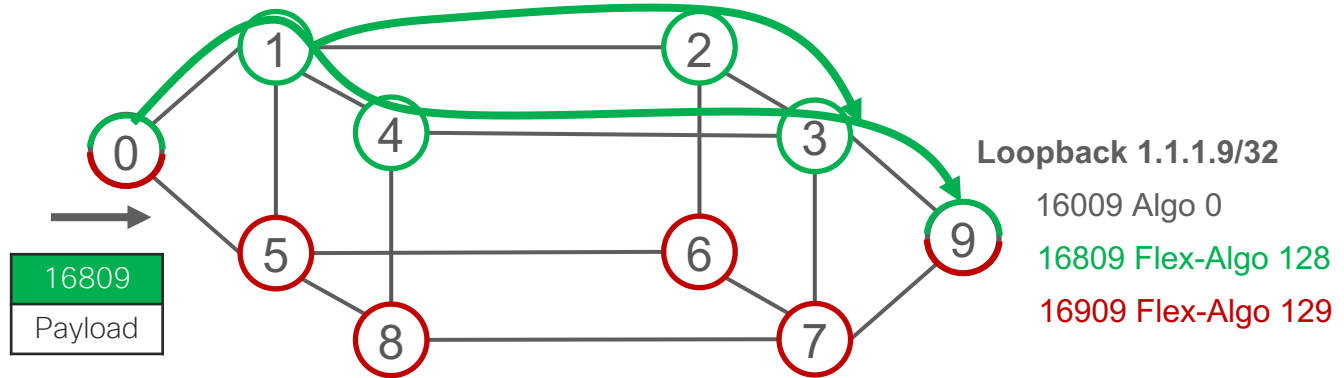
- All the nodes support Algo 0: minimize IGP metric
- Green nodes also support 128: minimize IGP metric
- Red nodes also support 129: minimize Delay

Use-Case – Multi-Plane Networks (cont.)



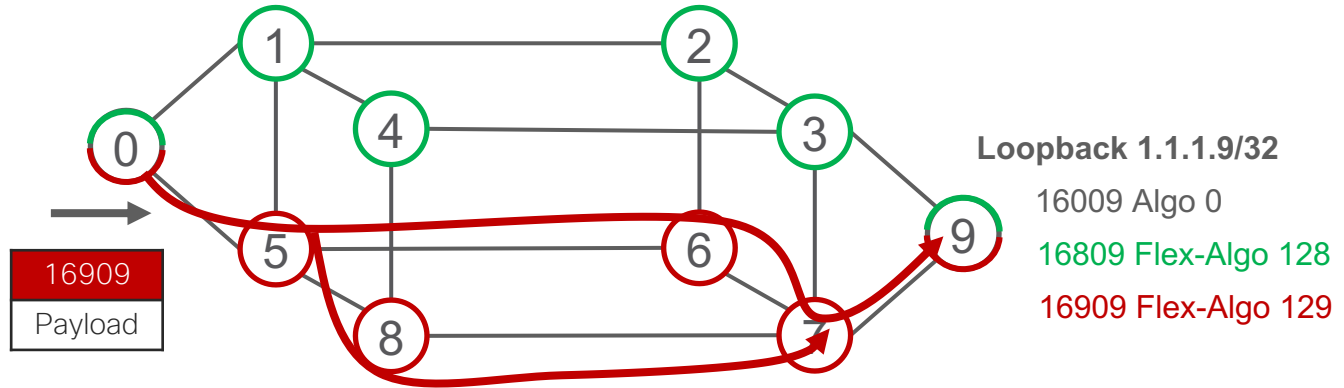
- Path to Node 9 across Algo 0

Use-Case – Multi-Plane Networks (cont.)



- Path to Node 9 across Flex-Algo 128

Use-Case – Multi-Plane Networks (cont.)



- Path to Node 9 across Flex-Algo 129

Automated Steering

- SRTE Automated Steering is leveraged for IGP Flex-Algo

```
segment-routing
traffic-eng
  on-demand color 100
  dynamic mpls
    sid-algorithm 128
```

“Any 100-colored BGP route should be steered
via the prefix-SID(ALGO 128) of the BGP nhop”

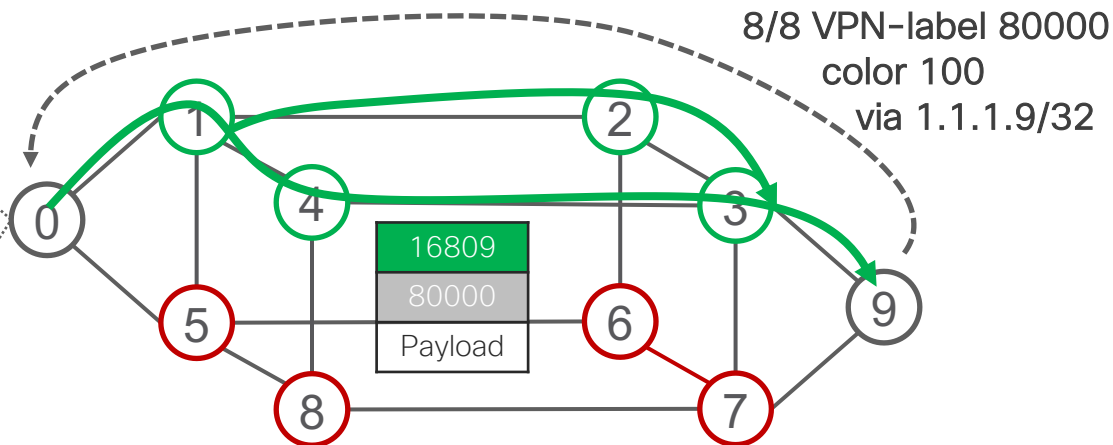
Automated Steering – Multi-Plane

```
segment-routing
traffic-eng
  on-demand color 100
  dynamic mpls
  sid-algorithm 128

router isis 1
  flex-algo 128
```

FIB

8/8: push 80000 16809



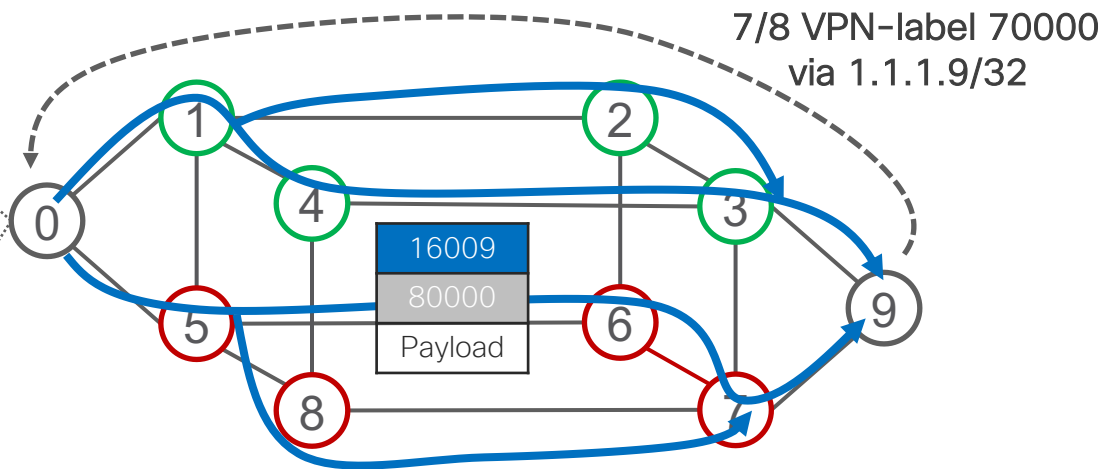
- Node 0 automatically steers any BGP route with color 100 from 9 via 16809 hence via the green plane only
- One single Flex-Algo Prefix-SID expresses the end-to-end SLA path

Automated Steering – Multi-Plane

```
segment-routing
traffic-eng
  on-demand color 100
  dynamic mpls
  sid-algorithm 128

router isis 1
  flex-algo 128
```

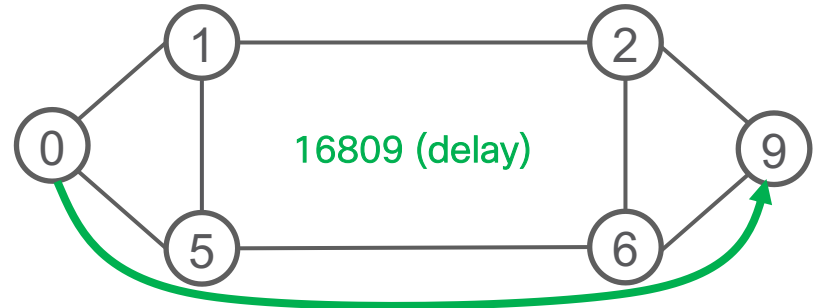
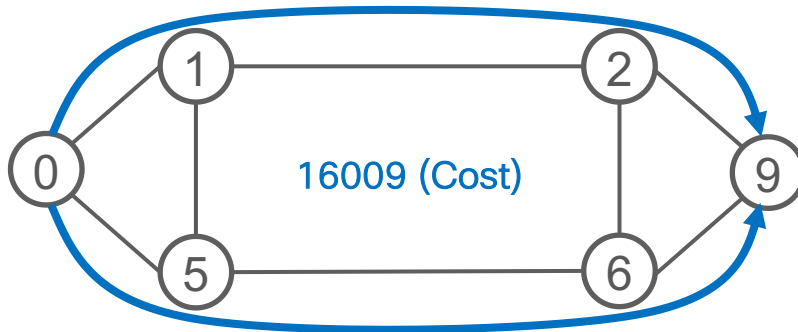
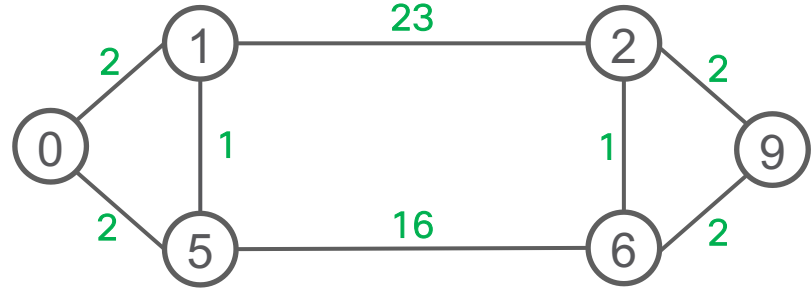
```
FIB
7/8: push 70000 16009
```



- Node 0 automatically steers any BGP route without color from 9 via 16009 (any plane)

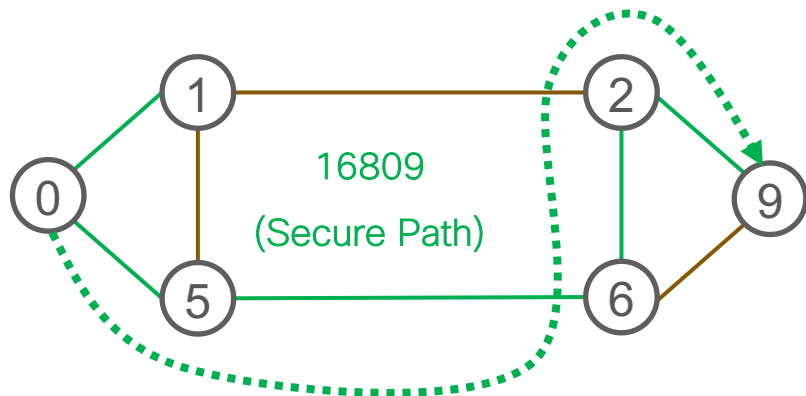
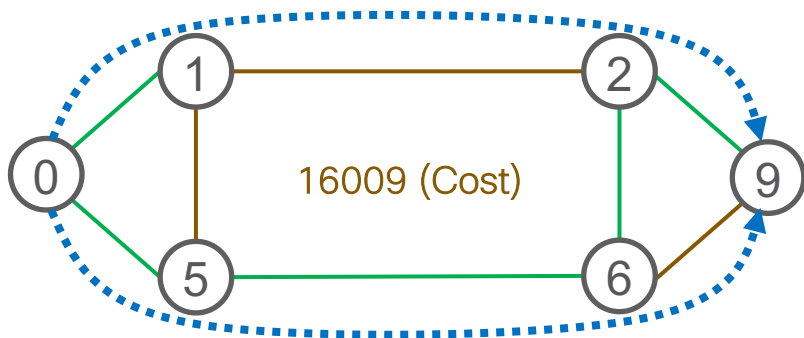
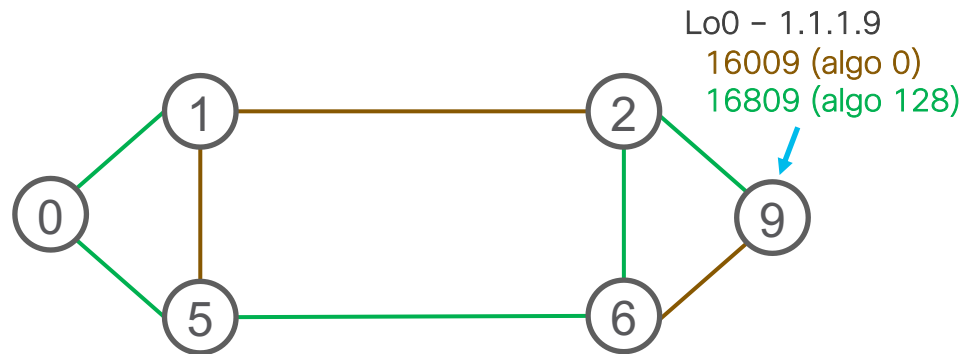
Use-Case – Delay vs Cost of Transport

- All nodes support Algo 0 & 128
- ISIS link metric 10
- Algo 128: minimize delay metric
- Per-link measurement of delay and advertisement as delay metric via ISIS
- Delay metric at that time shown in green



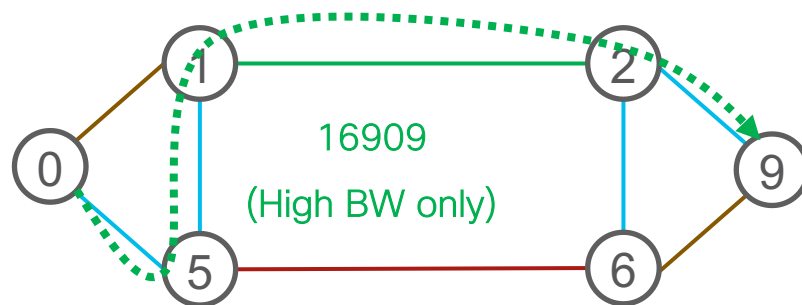
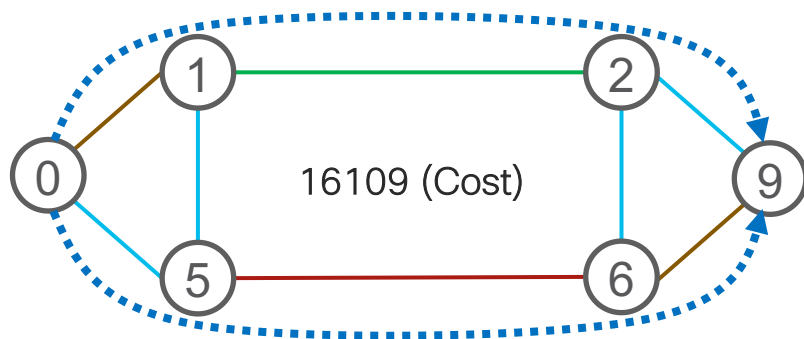
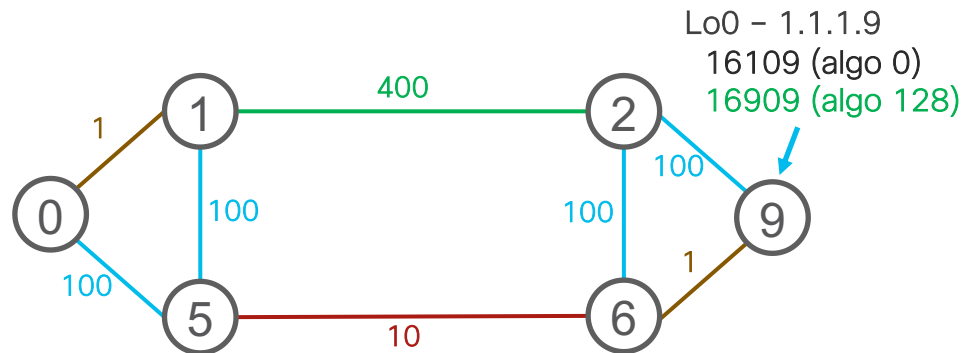
Use-Case - SRTE for Intelligent Secure Paths

- ISIS link metric 10
- Link colors shown **Unencrypted** / **Encrypted**
- All nodes support Algo 0 & 128
- Algo 128: minimize IGP while traversing links with encryption enabled (**exclude brown**)
- Per-link colors flooded in IGP



Use-Case - SRTE for High-BW Links Preference

- ISIS link metric 10
- Link colors shown 1G / 10G / 100G / 400G
- All nodes support Algo 0 & 128
- Algo 128: minimize IGP while traversing links of 100G or more (**exclude brown and red**)
- Per-link colors flooded in IGP



Demo Time

Conclusion

SR IGP Flexible Algorithm

- Complements the SRTE solution by adding new Prefix-Segments with specific optimization objective and constraints
 - minimize igp-metric or delay or te-metric
 - avoid link-affinity or SRLG
- TE path from anywhere to anywhere automatically computed by IGP
- Single SID is used to enforce traffic on the Flex-algo specific path
- Leverages the SRTE benefits of simplicity and automation
 - Automated sub-50msec FRR (TI-LFA)
 - On-Demand Policy (ODN)
 - Automated Steering (AS)

Thank You