SR IGP Flex Algo

Clarence Filsfils
Cisco Fellow – cf@cisco.com

Disclaimer

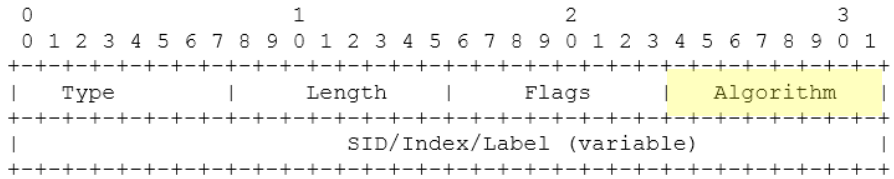
“Many of the products and features described herein remain in varying stages of development and will be offered on a when-and-if-available basis. This roadmap is subject to change at the sole discretion of Cisco, and Cisco will have no liability for delay in the delivery or failure to deliver any of the products or features set forth in this document.”

SR IGP Flex Algo

- Complements the SRTE solution by adding new Prefix-Segments with specific optimization objective and constraints
  - minimize igp-metric or delay or te-metric
  - avoid SRLG or affinity
- Leverages the SRTE benefits of simplicity and automation
  - Automated sub-50msec FRR (TILFA)
  - On-Demand Policy (ODN)
  - Automated Steering (AS)

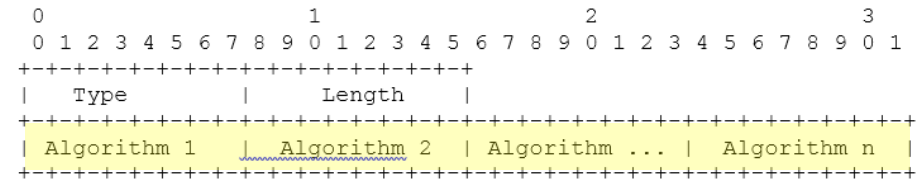
IGP SR Algorithm

- Each Prefix SID is related to an algorithm



- Each node advertises its ALGO capability

The SR-Algorithm sub-TLV has following format:



Currently defined algorithms

- 0: Shortest Path First (SPF) algorithm based on link metric.
  - This is the well-known shortest path algorithm as computed by the IS-IS Decision process. Consistent with the deployed practice for link-state protocols, algorithm 0 permits any node to overwrite the SPF path with a different path based on local policy
- 1: Strict Shortest Path First (SPF) algorithm based on link metric.
  - The algorithm is identical to algorithm 0 but algorithm 1 requires that all nodes along the path will honor the SPF routing decision. Local policy MUST NOT alter the forwarding decision computed by algorithm 1 at the node claiming to support algorithm

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, …
- Example
  - Operator1 defines Flex-Algo 128 as “minimize IGP metric and avoid link-affinity “green”
  - Operator2 defines Flex-Algo 128 as “minimize delay metric and avoid link-affinity “blue”

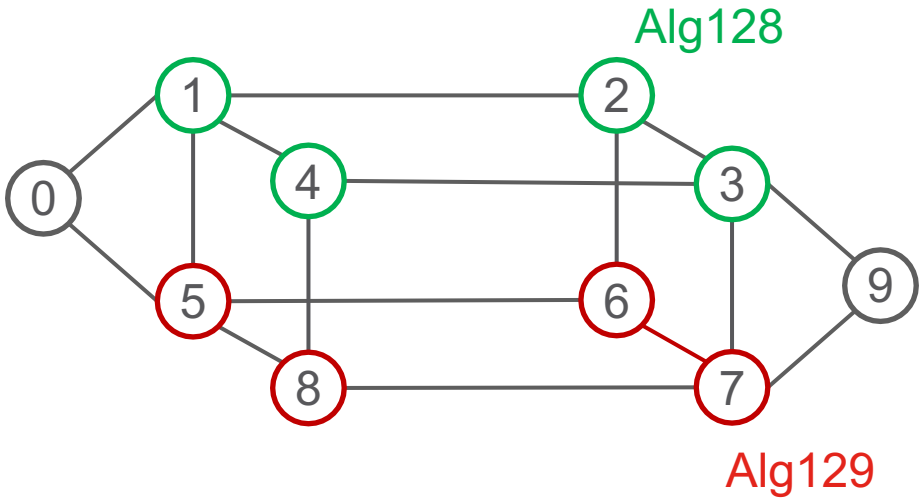
Flex-Algo Participation Advertisement

- 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

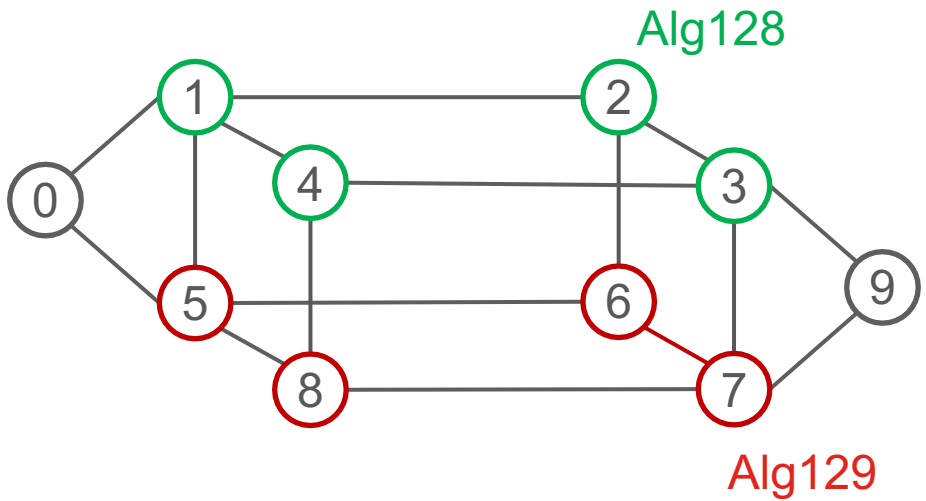
Prefix SID 16802 for ALGO 128

For example, for node N: 16000 + N

+ 0 for Algo 0

+ 800 for Algo 128

+ 900 for Algo 129



No additional loopback address

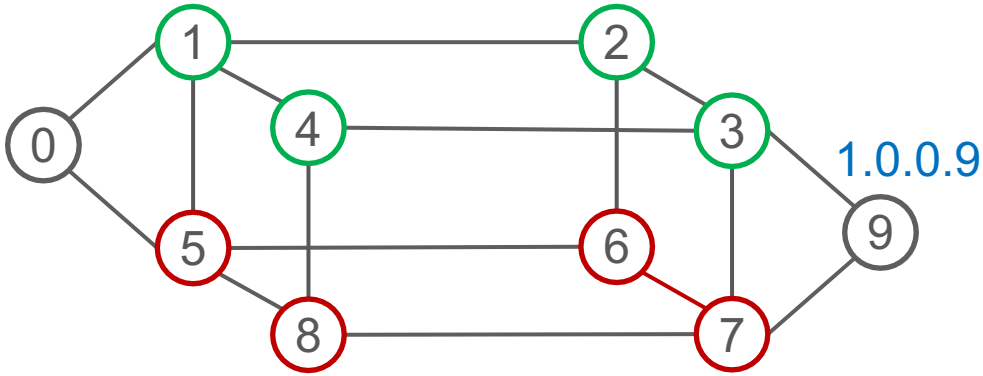
- Flex-Algo Prefix SID's can be advertised as additional prefix-SID's of the existing loopback address

Node 9 advertises loopback0 1.0.0.9/32 with

Prefix SID 16009 for ALGO 0

Prefix SID 16809 for ALGO 128

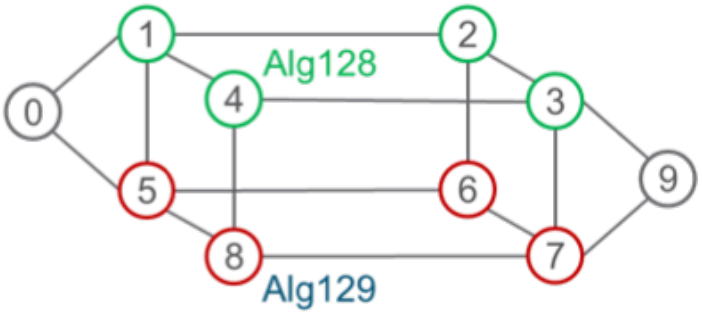
Prefix SID 16909 for ALGO 129



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

Algo 128: minimize IGP metric
Algo 129: minimize IGP metric



Flex-Algo Computation and Prefix-SID installation

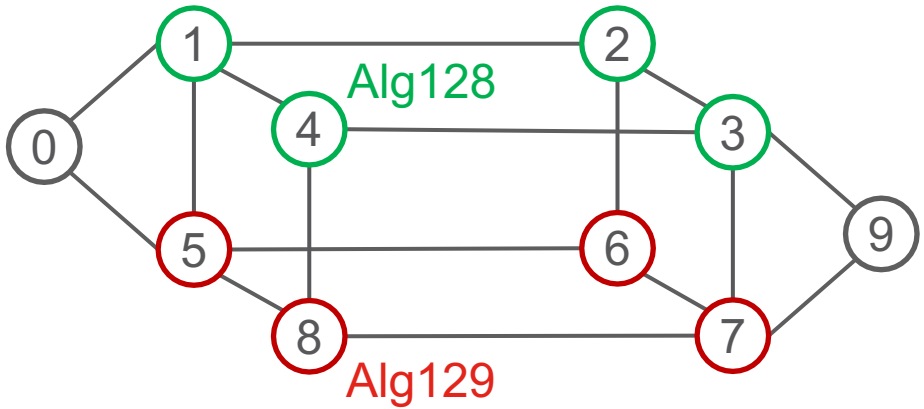
- A node N computes Flex-Algo K if
  - it is enabled for K, and
  - it has a consistent definition for K
- If so, the first step is to define the topology of K
  - N prunes any node that is not advertising participation to K
  - N prunes any link that is excluded by the algorithm of K
    - > e.g. if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

Flex-Algo Computation and Prefix-SID installation

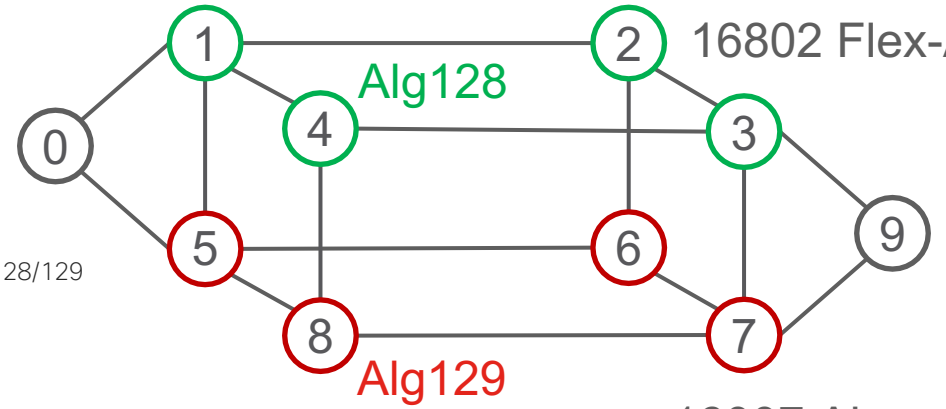
- The second step is to compute shortest-path tree on Topo(K) with the metric defined by K
  - it could be the IGP metric, the TE metric or the delay
- The third step is to install any reachable Prefix-SID of Flex-Algo K in the forwarding table

Example 1

- Grey nodes support Algo 0/128/129
- Green nodes support 0/128
- Red nodes support 0/129
- Algo 128: minimize IGP metric
- Algo 129: minimize IGP metric
- Nodes advertise a Prefix SID for each Algo they support
  - For example, for node N: 16000 + N
    - > + 0 for Algo 0
    - > + 800 for Algo 128
    - > + 900 for Algo 129



Example 1



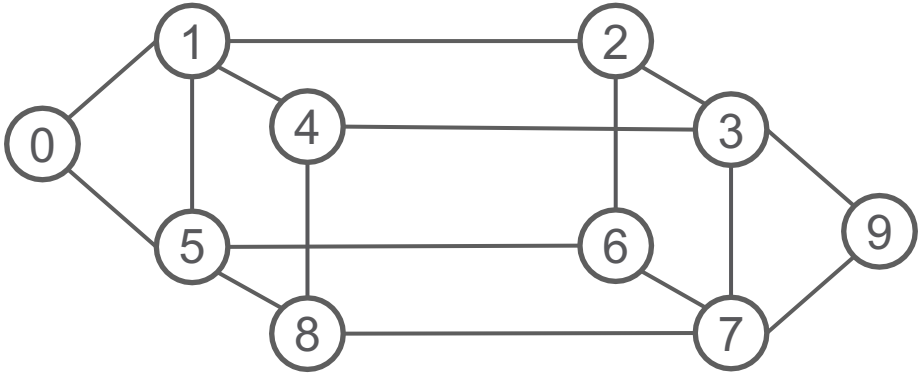
16002 Algo 0
16802 Flex-Algo 128

16009 Algo 0
16809 Flex-Algo 128
16909 Flex-Algo 129

16007 Algo 0
16907 Flex-Algo 129

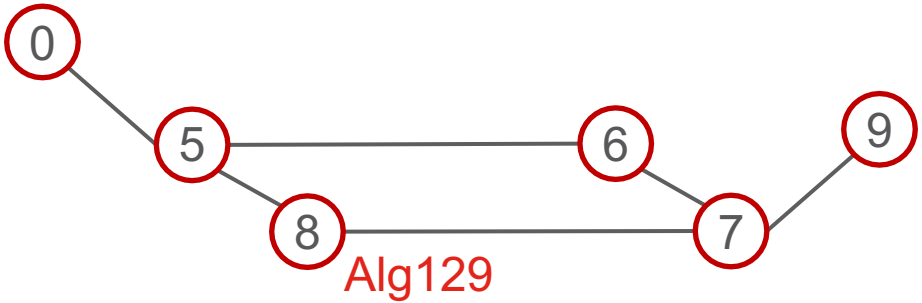
- Grey nodes support Algo 0/128/129
- Green nodes support 0/128
- Red nodes support 0/129
- Algo 128: minimize IGP metric
- Algo 129: minimize IGP metric
- Nodes advertise a Prefix SID for each Algo they support
  - For example, for node N: 16000 + N
    - + 0 for Algo 0
    - + 800 for Algo 128
    - + 900 for Algo 129

Example 1 – Topo(0)

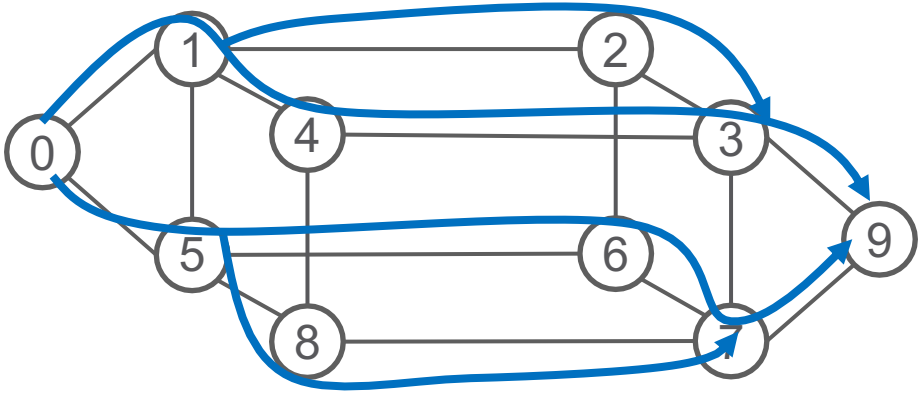


Example 1 – Topo(128)

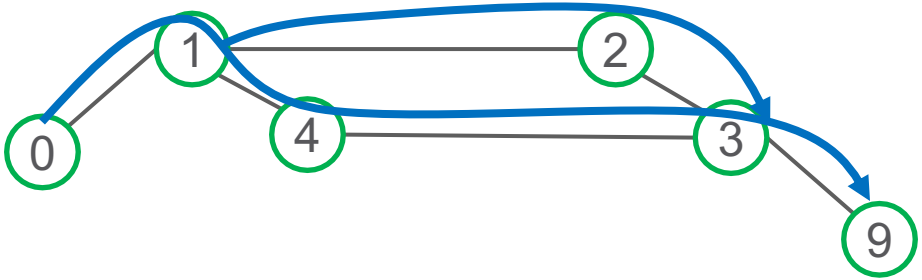
Example 1 – Topo(129)



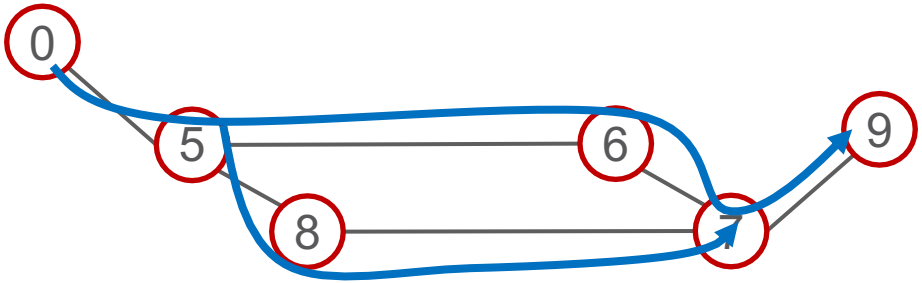
Example 1 – Prefix-SID 16009 of Algo 0



Example 1 – Prefix-SID 16809 of Flex-Algo 128



Example 1 – Prefix-SID 16909 of Flex-Algo 129

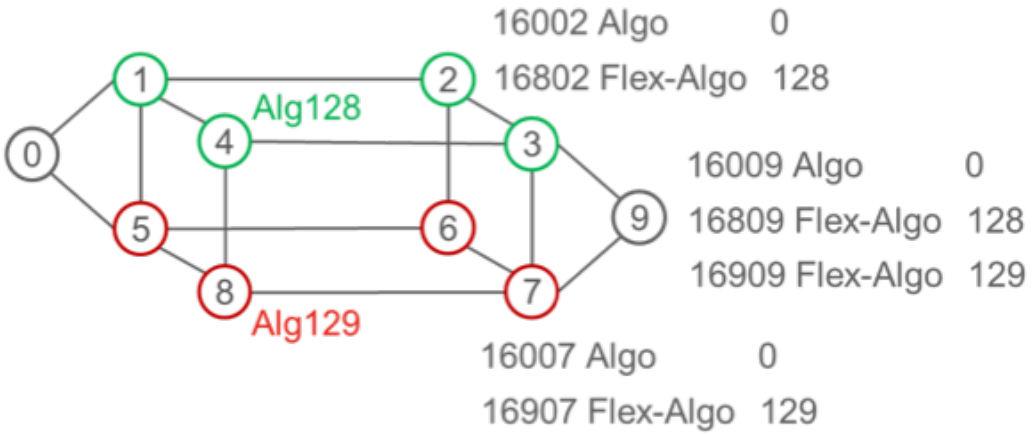


Example 1 – FIB of node 0

- 16009 via 1 or 5
- 16809 via 1
- 16909 via 5

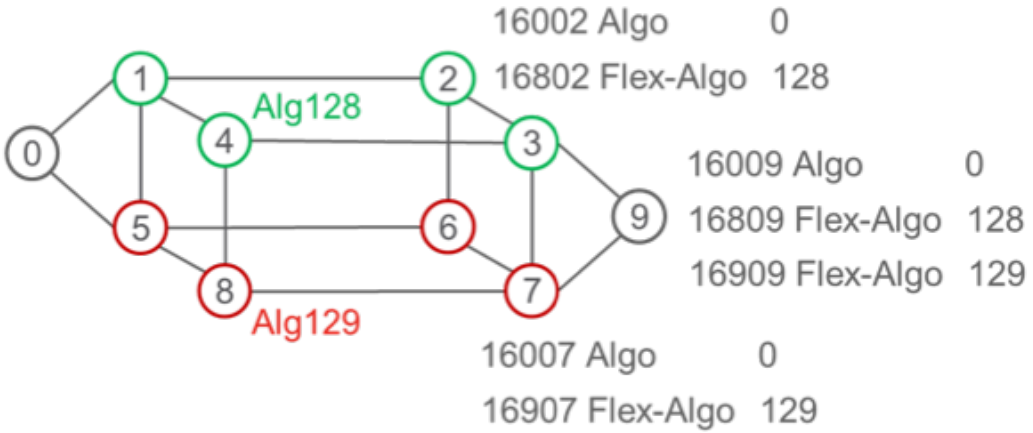
- 16002 via 1
- 16802 via 1

- 16007 via 5
- 16907 via 5



Example 1 – FIB of node 1

- 16009 via 2 or 4
- 16809 via 2 or 4
- 16002 via 2
- 16802 via 2



- Node 1 does not install Prefix-SID for Flex-Algo 129 as Node1 does not participate in 129

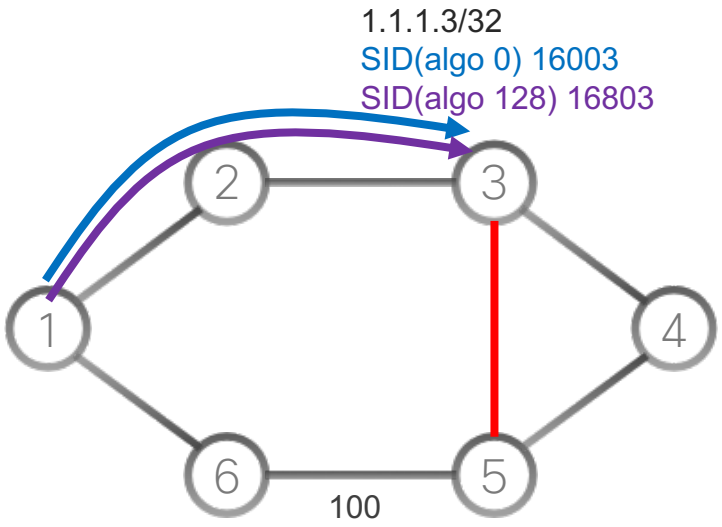
Notes

- This computation is performed by any node part of K
- If a node is part of 2 Flex-Algo's, it performs the described computation independently for each Flex-Algo
- ECMP is obviously supported in each Flex-Algo

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: I:10

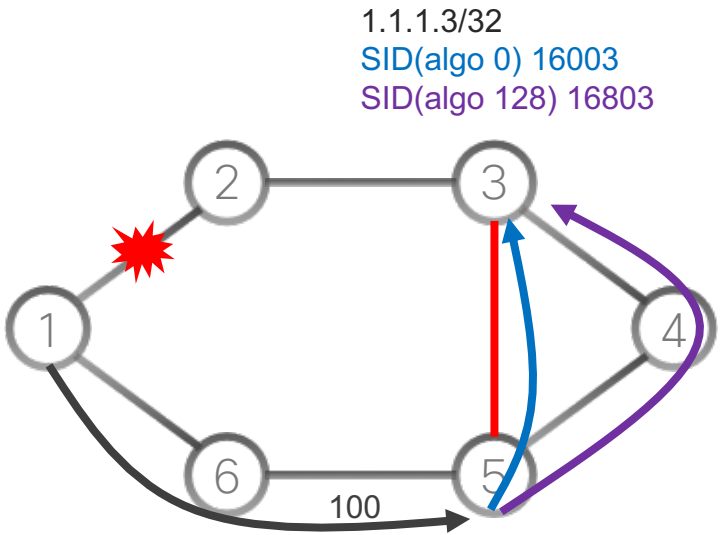
At node 1 for destination 3

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 – TILFA Backup path per Algo



At node 1 for destination 3

16003 => 16003 via 2
Backup: <24065, 16003> via 6

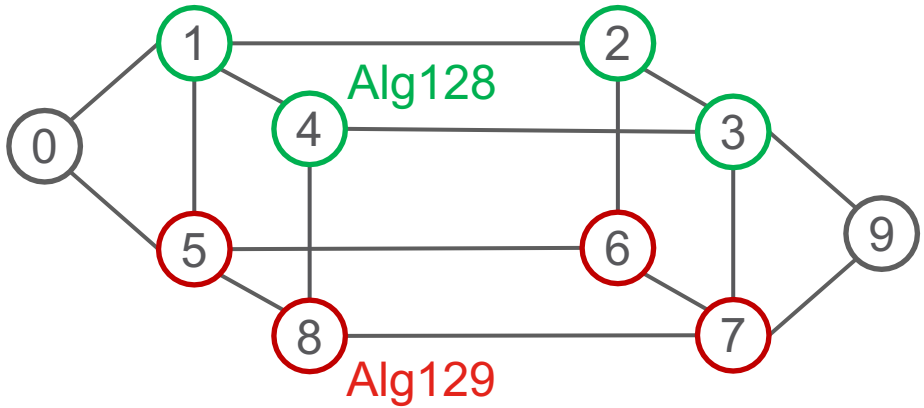
16803 => 16803 via 2
Backup: <24065, 16803> via 6

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

Use-Case – Dual Plane

- Grey nodes support Algo 0/128/129
- Green nodes support 0/128
- Red nodes support 0/129
- Algo 128: minimize IGP metric
- Algo 129: minimize IGP metric
- Nodes advertise a Prefix SID for each Algo they support
  - For example, for node N: 16000 + N
    - > + 0 for Algo 0
    - > + 800 for Algo 128
    - > + 900 for Algo 129

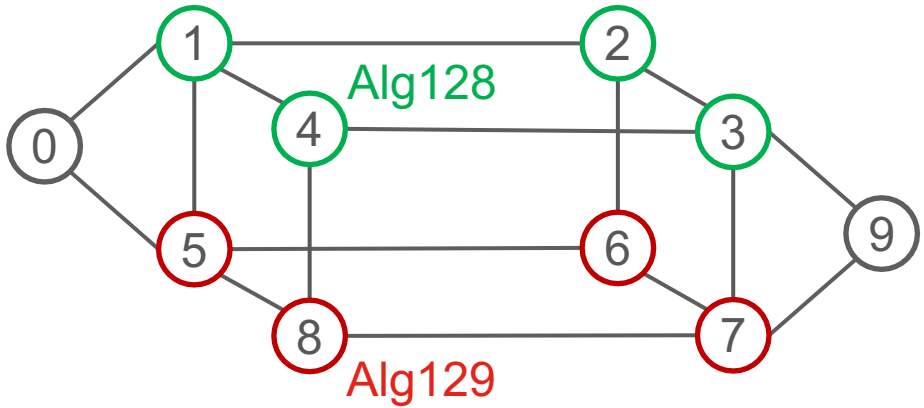


Note: use of TE-affinities is not necessary

CLI – Dual Plane

Config of Node 2

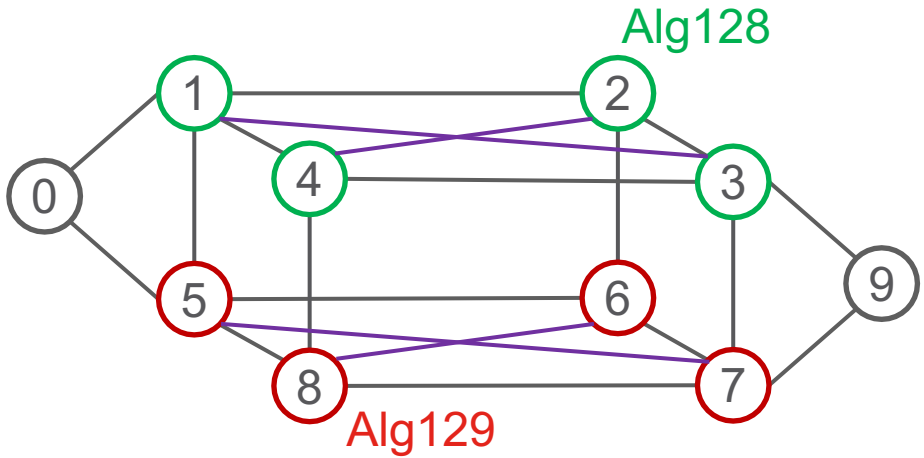
```
router isis 1
 is-type level-2-only
 net 49.0001.0000.0000.0002.00
 flex-algo 128
!
 address-family ipv4 unicast
   router-id 1.1.1.2
   segment-routing mpls
 !
 interface Loopback0
   address-family ipv4 unicast
    prefix-sid absolute 16002
    prefix-sid algorithm 128 absolute 16802
 !
```

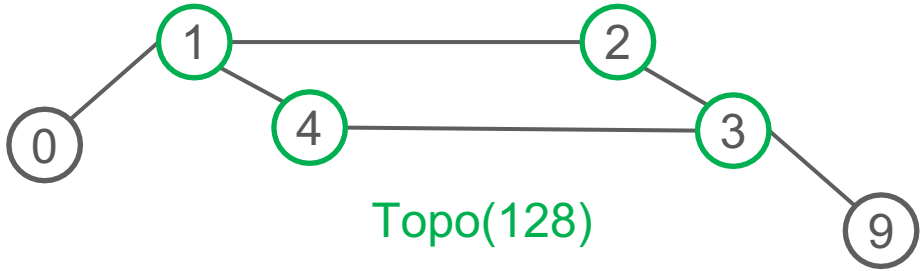


Note: use of TE-affinities is not necessary

Use-Case – Dual Plane with link affinity

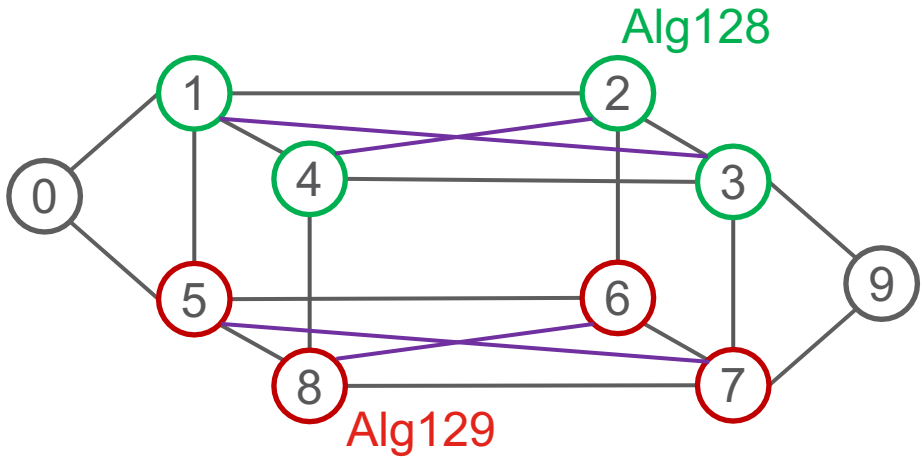
- Grey nodes support Algo 0/128/129
- Green nodes support 0/128
- Red nodes support 0/129
- Algo 128: minimize IGP metric and exclude purple links
- Algo 129: minimize IGP metric and exclude purple links





CLI – Dual Plane with link affinity

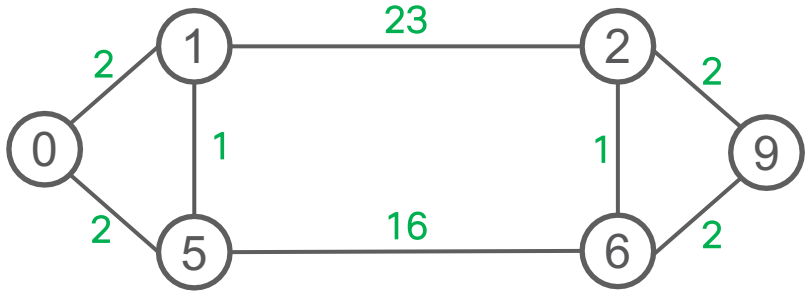
```
router isis 1
 is-type level-2-only
 net 49.0001.0000.0000.0002.00
 affinity-map PURPLE bit-position 1
 flex-algo 128
   exclude affinity PURPLE
 !
 address-family ipv4 unicast
   router-id 1.1.1.2
   segment-routing mpls
 !
 interface Loopback0
   address-family ipv4 unicast
    prefix-sid absolute 16002
    prefix-sid algorithm 128 absolute 16802
 !
 interface GigabitEthernet0/2/0/4
  affinity flex-algo PURPLE
```

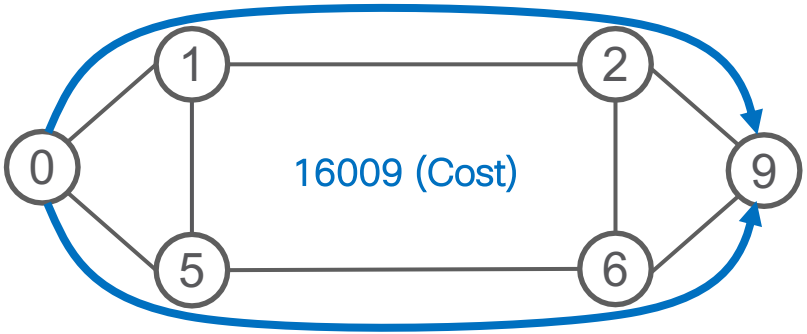


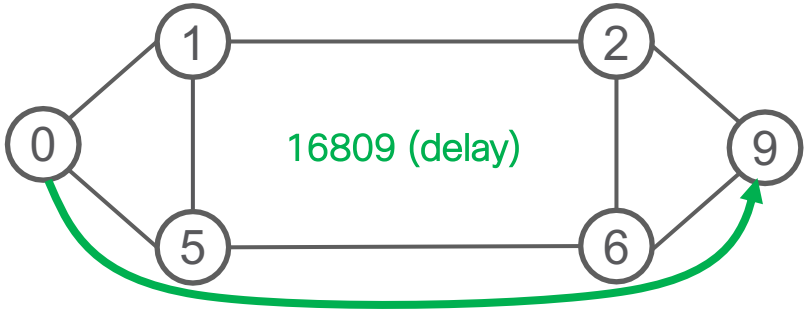
Config of Node 2

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

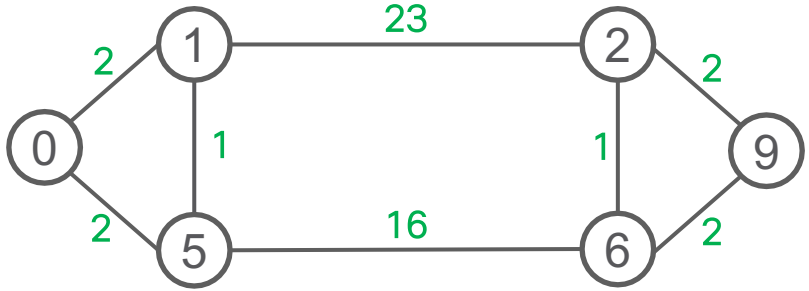






CLI – delay vs Cost of Transport

```
router isis 1
 is-type level-2-only
 net 49.0001.0000.0000.0002.00
 flex-algo 128
   metric-type delay
 !
 address-family ipv4 unicast
   router-id 1.1.1.2
   segment-routing mpls
 !
 interface Loopback0
   address-family ipv4 unicast
    prefix-sid absolute 16002
    prefix-sid algorithm 128 absolute 16802
 !
performance-measurement
   interface GigE0/0/2/6
     delay-measurement
```



Config of Node 2

Automated Steering

- SRTE Automated Steering is leveraged for IGP Flex-Algo

```
segment-routing
  traffic-eng
    on-demand color 100
       dynamic mpls
          flex-algo 128
```

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

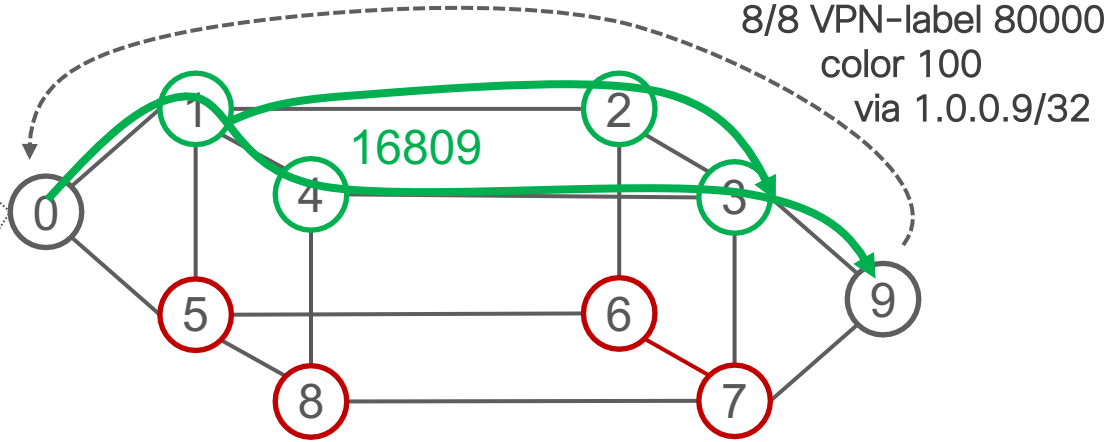
Automated Steering – Dual Plane

```
segment-routing
  traffic-eng
    on-demand color 100
      dynamic mpls
       flex-algo 128

router isis 1
  flex-algo 128
```

```
FIB

8/8: push 80000 16809
```



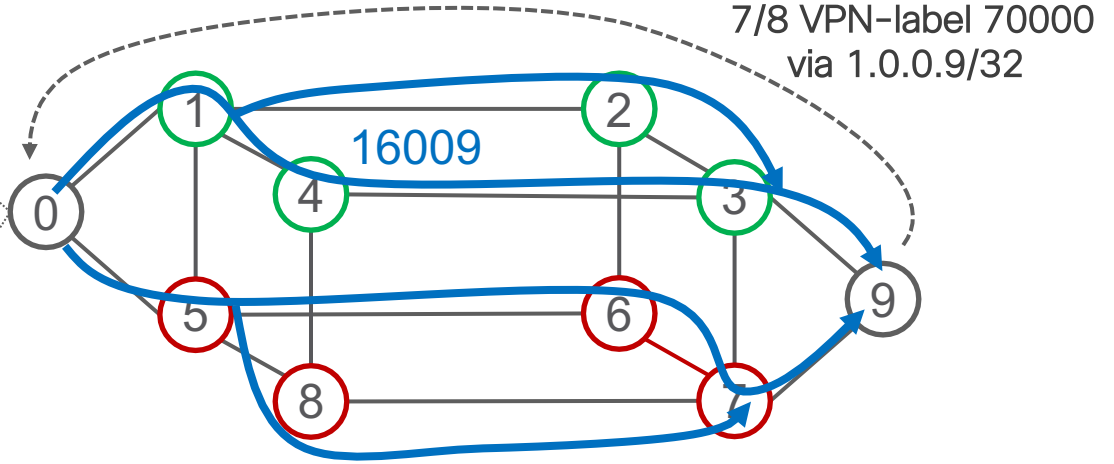
- Node 0 automatically steers any BGP route with color 100 from 9 via 16809 hence via the green plane only
- One single Flex-Algo Prefix-SID expresses the end-to-end SLA path

Automated Steering – Dual Plane

```
segment-routing
  traffic-eng
    on-demand color 100
      dynamic mpls
       flex-algo 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)

Consistency

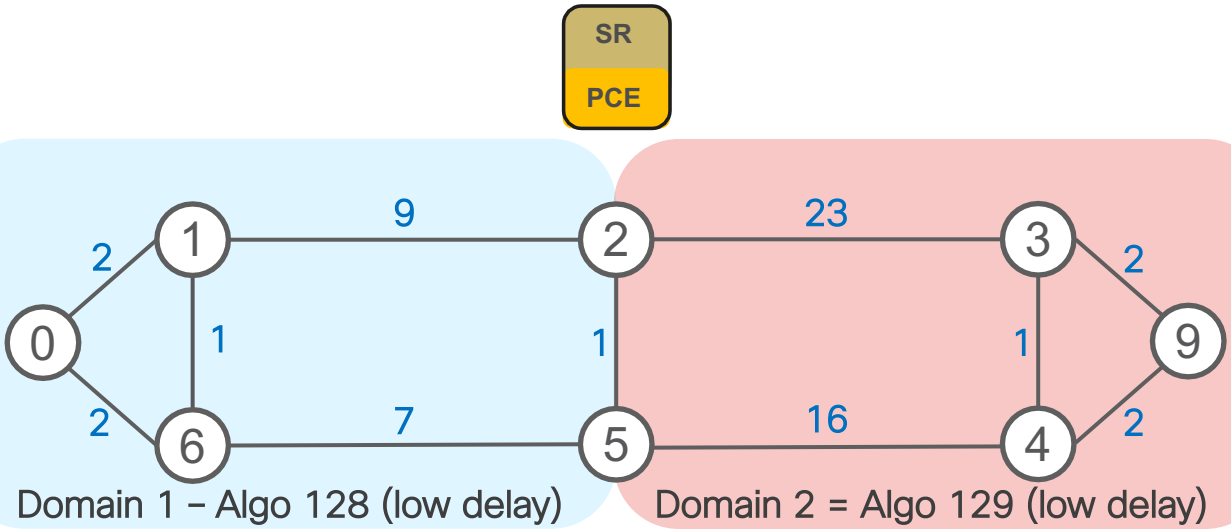
- Any node advertises his definition of Flex-Algo
- As soon as a difference between two definitions of the same Flex-Algo is detected, the Flex-Algo is disabled
  - any Prefix-SID of that Flex-Algo is removed from FIB
  - no path is computed
- Recommendation
  - configure two nodes to advertise the Flex-Algo definition for the domain
    - one with a higher priority
  - do not configure any per-node local definition
  - this way, all the nodes deterministically use the same definition
- Or leverage a solution such as NSO to ensure the domain-wide consistency of the config

OSPF and SRv6

- Same applies to OSPF
- Same applies to SRv6

ODN and AS Inter-Domain delay

```
segment-routing
 traffic-eng
  on-demand color 100
   dynamic mpls
    pce
    metric-type delay
```

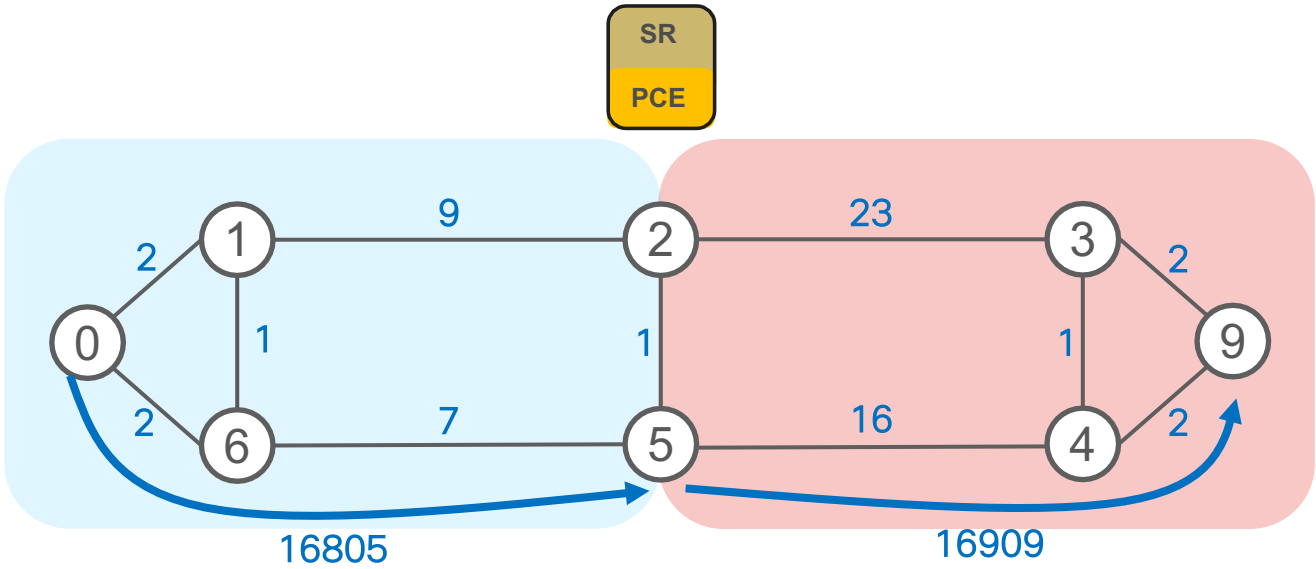


- The delay of each link is reported in drawing
- The ISIS metric per link is 10
- 9 advertises 2/8 with color 100

ODN and AS Inter-Domain delay – Cont.

```
FIB

2/8: push <16805, 16909>
```



- Upon receiving 2/8, node 0 dynamically creates an SRTE policy to 9
- As 9 is beyond its domain, node 0 requests the computation from its PCE and indicates that Flex-Algo128 is needed
- PCE replies with MPLS stack <16805, 16809>

IETF

- draft-ietf-spring-segment-routing
  - Prefix-SID per Algorithm
- draft-filsfils-spring-segment-routing-policy
  - SRTE architecture, ODN, AS
- draft-hegdeppsenak-isis-sr-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

" Flex Algo is very valuable addition to SRTE solution that allow to auto-steer unicast and multicast traffic via any topology/path based on operator defined logic.

Ease of configuration and operational management, ability to provision dynamic constrained paths based on a single SR label with local repair (TI-LFA) respecting the same constraints as the primary path are some of the benefits that we might realize with Flex Algo."

Arkadiy Gulko – Thomson Reuters

Real

- Visit the SR booth and enjoy the numerous demos!

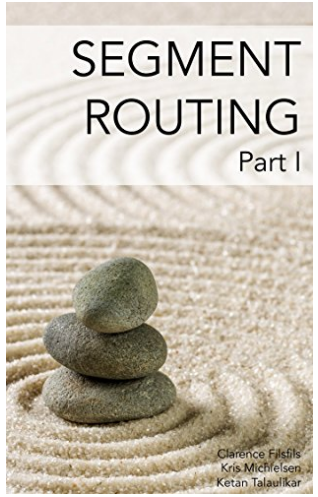
Conclusion

SR IGP Flexible Algorithm

- Scalability
  - Single SID is used to enforce traffic on the Flex-algo specific path
- Flexible
  - Any operator can define its custom algorithm
- Functionality
  - Optimum TILFA backup paths respecting the same constraints as primary path
  - Inter-domain SR Policy based on SR IGP Flex Algo
  - Applicable to multicast
- Automation and Simplicity
  - TE-path from anywhere to anywhere automatically computed by IGP
  - ODN and Automated Steering
  - Network-wide consistency check

Stay uptodate

amzn.com/B01I58LSUO



segment-routing.net

linkedin.com/groups/8266623

twitter.com/SegmentRouting

facebook.com/SegmentRouting/

Contributors

- Jose Liste
- Kris Michielsen
- Peter Psenak
- Tarek Saad


CISCO