draft-filsfils-spring-segment-routing-policy-00

IETF 98 Chicago

Summary

- Define some specifities of traffic engineering using Segment Routing
- Define the concept of an SR policy
- Define methods to steer traffic into an SR policy
- It is agnostic to the SR dataplane

Learning topology

- SR-TE architecture is multi-domain capable
- Multiple sources of topology:
  - BGP-LS, IGP, NETCONF…

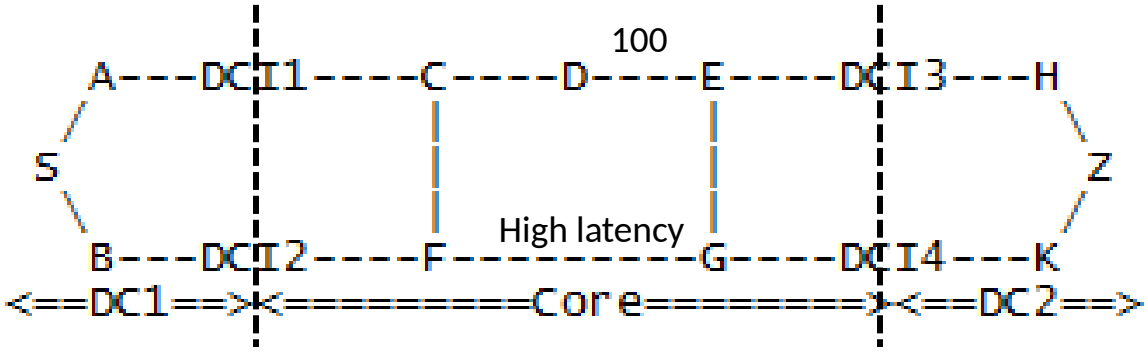
The SR policy

- Identified by the tuple:
  - Head-end (where the policy will be instantiated)
  - Endpoint (what is the destination)
  - Color (that would help steering traffic)
- An SR policy may be created through multiple ways: BGP, PCEP, NETCONF, CLI…
- An SR policy may have multiple candidate paths
  - A single path is selected (preference based)
  - And installed in the FIB
- A path is associated with one or more lists of SIDs and an optional weight (UCMP)

SR policy:
<HEAD,COLOR, ENDPOINT>
Path1 preference X:
SID_list#1, weight W1
Path2 preference Y:
SID_list#2, weight W2
SID_list#3, weight W3

Use of the binding SID

- A binding SID is associated with an SR policy path
- It provides a way to reduce the number of segments pushed by the initial source
- It pr nchor

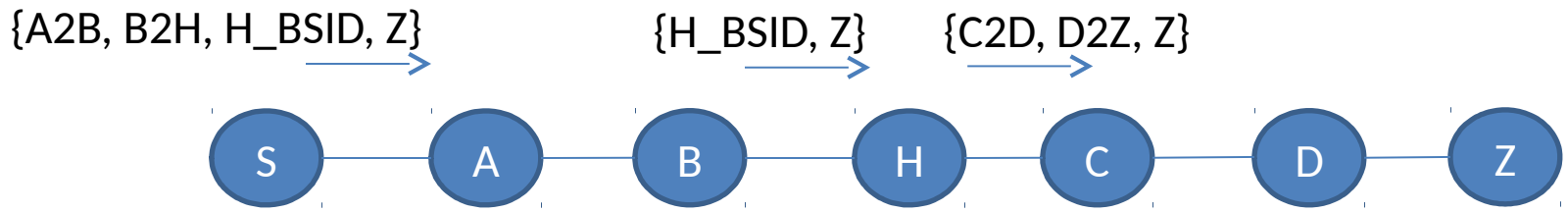


Without binding SID, a low latency path from S to Z is: <DCI1,D,D2E,DCI3,Z>

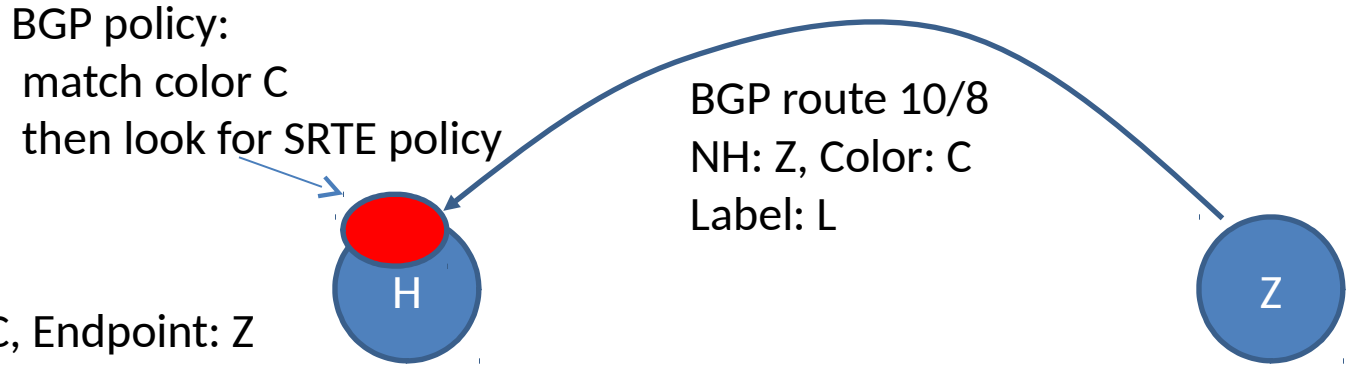
With binding SID allocated by DCI1: <DCI1, BSID, Z> where BSID=<D,D2E,DCI3>

Traffic steering

- An SR path uses BSID corresponding to an SR policy



- Recursion on a BSID

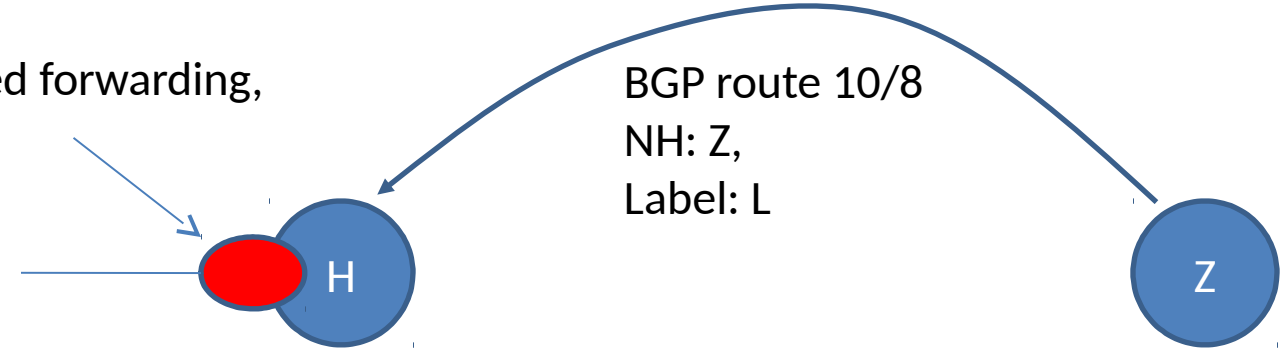


SR policy:
Head: H,Color: C, Endpoint: Z
-> BSID: B
-> SIDlist : {S1,S2,S3}

FIB: (hierarchical)
10/8 push L -> B -> {S1,S2,S3}

Traffic steering

- Class based traffic steering



SR policy:
Head: H,Color: C1, Endpoint: Z
-> BSID: B1
-> SIDlist : {S1,S2,S3}

Head: H,Color: C2, Endpoint: Z
-> BSID: B2
-> SIDlist : {S4,S5,S6}

FIB: (hierarchical)
10/8 push L
FC0 -> IGP path
FC1 -> B1 -> {S1,S2,S3}
FC2 -> B2 -> {S4,S5,S6}

Next steps

- We welcome comments
- This document is a base to understand protocol extension documents