ISIS extensions for SRv6

draft-bashandy-isis-srv6-extensions-00

Authors:
Ahmed Bashandy, Cisco Systems
Clarence Filsfils, Cisco Systems
Les Ginsberg, Cisco Systems
Bruno Decraene, Orange

IETF98, Mar/2017
Chicago, USA
Separate SR MPLS and SRv6 Drafts

draft-ietf-isis-segment-routing-extensions limited to MPLS dataplane

Latest version has removed:

- IPv6 address option from prefix-SID/ADJ-SID
- References to IPv6 SRH(H flag in SR Capabilities)
- Explicitly state draft is for MPLS dataplane only

Ready for Last Call
MUST READ !!!!!!!

draft-filsfils-spring-srv6-network-programming

Also Read

draft-ietf-6man-segment-routing-header

98th IETF, Chicago, March 2017
What we want to do

Define protocol extensions to support segment routing over IPv6 Dataplane

- Advertise segments IDs (SIDs)
- Support for the following functions
  - fast convergence
  - traffic engineering
Proposed ISIS Extensions

• 1 Capability sub-TLV: SRv6 Capabilities Sub-TLV
• 1 Top Level TLV: SRv6 SID TLV
• 2 sub-TLV of neighbor reachability:
  – P2P SRv6 cross-connect SID
  – LAN SRv6 cross-connect SID
• 1 SRv6-Function Descriptor (sub-TLV or sub-sub-TLV)
SRv6 Capabilities Sub-TLV

Indicates support for SRv6

Existing SR Capability sub-TLV defined in draft-ietf-isis-segment-routing-extensions is SR MPLS only

One flag defined:

E => T.encap is supported
SRv6 Capabilities Sub-TLV(2)

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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   Type        |     Length    |            Flags              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      max-SL   |Max-End-Pop-SRH| Max-T-Ins-SRH |Max-T-Encap-SRH|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| max-End-D-SRH |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

- Max-SL: Maximum Received SL in the SRH
- Max-End-Pop-SRH: Maximum number of SIDs when applying PSP or USP flavors (0 => not supported)
- Max-T-Ins-SRH: Maximum number of SIDs when applying T.insert (0 => not supported)
- Max-T-Encap-SRH: Maximum number of SIDs when applying T.Encap (Valid when E-flag is set)
- Max-End-D-SRH: Maximum number of SIDs when applying End.DX6 or End.DT6

98th IETF, Chicago, March 2017
SRv6 SID TLV

<table>
<thead>
<tr>
<th>0</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5</td>
<td></td>
</tr>
</tbody>
</table>
| +---------------------------------+
| | Type | Length |
| +---------------------------------+

- Top Level TLV
  - Not a prefix to be installed in RIB/FIB
  - Minimum impact on routing functionality
- Advertises SRv6 SIDs and the associated attributes
- Used for SIDs not related to neighbors
- Does **NOT** result in *routing* action on its own
- Shares sub-TLV space with prefix reachability TLV (135/235/236/237)
- Can be leaked between levels
- Advertised SRv6 SID **need not** be covered by IPv6 prefix reachability (TLV 236 and 237)
  - E.g. may be reachable via static route

98th IETF, Chicago, March 2017
SRv6 SID

- D flag – Set when leaked from Level-2 to Level-1
- One sub-TLV defined: SRv6 Function Descriptor
- Multiple SIDs can be in one TLV
P2P X-SID sub-TLV

- Sub-TLV of neighbor reachability
  - P2P adjacency
  - Used to advertise SRv6 SIDs associated with
    - End.X
    - End.DX6

- Same structure as SRv6 SID TLV
- Takes sub-sub-TLVs

  Associated Function is specified by encoding the “**SRv6 Function Descriptor**” sub-TLV

98th IETF, Chicago, March 2017
LAN X-SID sub-TLV

- Similar to P2P X-SID but used for LAN neighbors
  - neighbor System ID is included
- Takes sub-sub-TLVs
  - Associated Function is specified by encoding the “SRv6 Function Descriptor” sub-sub-TLV

98th IETF, Chicago, March 2017
SRv6 Function Descriptor

• Container for advertising SRv6 function associated with a SID

• Behavior
  – Encodes one function with optional flavors

• Multiple Behaviors may be in one sub-TLV

• Sub-TLV in SRv6 SID TLV

• Sub-sub-TLV in P2P/LAN X-SID sub-TLVs
Behavior

- Encodes one function with optional flavors
- Flags
  - P-flag: If set then PSP flavor is enabled
  - U-flag: If set, then USP flavor is enabled
- Function
  - Encodes the function code point of the function
Next Step

WG adoption