



IETF 102 – Montreal  
July 2018  
IDR Working Group

# Advertising Segment Routing Policies in BGP

## draft-ietf-idr-segment-routing-te-policy-04

Dhanendra Jain on behalf of

Stefano Previdi (stefano@previdi.net)

Clarence Filselfil (cfilselfil@cisco.com)

Dhanendra Jain (dhjain@cisco.com)

Paul Mattes (pamattes@microsoft.com)

Eric Rosen (erosen@juniper.net)

Steven Lin (stevenlin@google.com)

Arjun Sreekantia (asreekan@cisco.com)

Acee Lindem (acee@cisco.com)

Siva Sivabalan (msiva@cisco.com)

Imtiaz Mohammed (imtiyaz@arista.com)

Gaurav Dawara (Gdawra.ietf@gmail.com)

# Agenda

- Share updates with the WG
  - Last update was presented at IETF-98
  - Latest update submitted as revision 04 prior to IETF-102
  - <https://tools.ietf.org/html/draft-ietf-idr-segment-routing-te-policy-04>
- Collect feedback/comments from the WG

# Introduction

- This draft defines signaling of Segment Routing Policies via BGP protocol
- BGP Signals a Candidate Path of a given SR Policy
- A new SAFI (SR-Policy, code 73) is defined in this draft
  - Identification of the SR Policy is encoded in NLRI bits
  - Details of the SR Policy Candidate Path are encoded in the SR Policy TLV within Tunnel Encapsulation attribute
- Defines extensions to the Color Extended Community to achieve Automatic Steering

```
SR Policy SAFI NLRI:  
<Distinguisher, Policy-Color,  
Endpoint>  
  
Attributes:  
Tunnel Encaps Attribute (23)  
  Tunnel Type (15): SR Policy  
    Binding SID  
    Preference  
    Priority  
    Policy Name  
    Explicit NULL Label Policy (ENLP)  
    Segment List  
      Weight  
      Segment  
      Segment  
      ...  
      ...
```

# Summary of Updates

- Updates to the Segment Types
  - Correction in SID type 3 and 8 definitions
  - Addition of segment 9, 10, 11 to cover SRv6 segments
- Addition of new sub-TLVs
  - SR Policy Symbolic name sub-TLV
  - SR Policy Priority sub-TLV
  - ENLP sub-TLV
- Addition of SR Flex Algorithm specification in Segment Type sub-TLV
  - Type 3, 8 refer to the Segments with IP Prefix
  - Head-end calculates the SR SID corresponding to the prefix
  - Addition of SR Algorithm ID to indicate Head-End to calculate the Flex-Algo SID
- Addition of new flags in Segment Type sub-TLV
  - V-Flag : Enable Verification of the SID supplied by the controller
  - A-Flag: Enable SR Algorithm
- Addition of new flags to Binding SID
  - S-Flag : “specified-BSID-only” behavior
  - I-Flag: “Drop upon Invalid” behavior
- Other updates

```
SR Policy SAFI NLRI:
<Distinguisher, Policy-Color, Endpoint>

Attributes:
Tunnel Encaps Attribute (23)
Tunnel Type (15): SR Policy
Binding SID
Preference
Priority
Policy Name
Explicit NULL Label Policy (ENLP)
Segment List
Weight
Segment
Segment
...
```

# Other updates

- Next-Hop address length specification to cover IPv4 or IPv6 next-hop in both SAFIs
- Defaults for Policy Preference, Weight parameters
- Updates to align terminology and the section references post WG adoption of SR Policy Architecture doc
  - <https://tools.ietf.org/html/draft-ietf-spring-segment-routing-policy-01>
  - <https://tools.ietf.org/html/draft-filsfils-spring-sr-policy-considerations-01>
- Updates to the Error handling text in few sections
  - Malformed sub-TLVs
  - Duplicate sub-TLVs
  - Clarification on Mandatory and optional sub-TLVs
- Editorial corrections

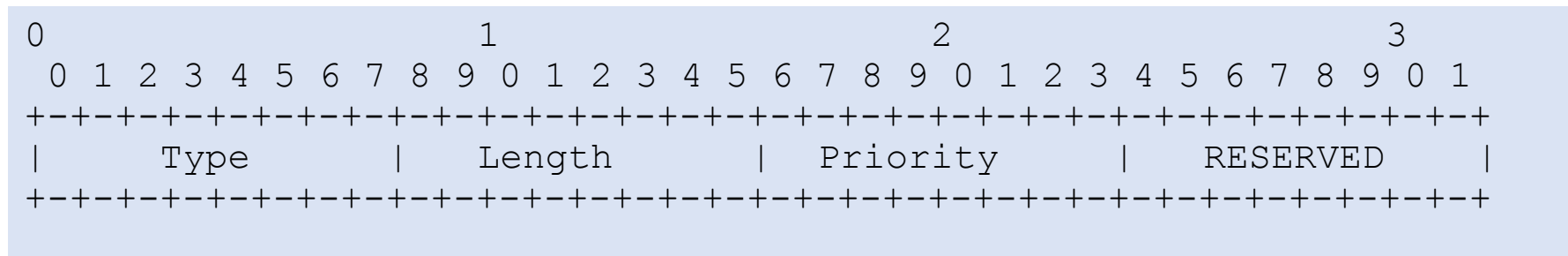
# Next Steps

- Address any comments
- IANA code points assignments for newly defined sub-TLVs and Flags
- Request for WGLC subsequently

# Backup (sub-TLV details)

# New Sub-TLVs (Policy Priority)

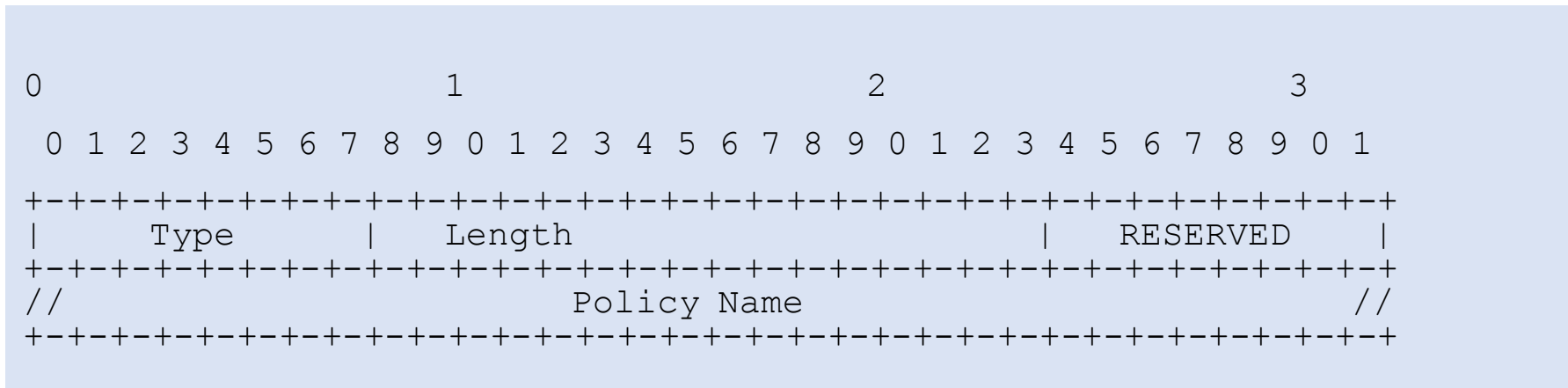
- Policy Priority sub-TLV
- An operator MAY set the Policy Priority sub-TLV to indicate the order in which the SR policies are re-computed upon topological change
- Reference - section 2 in <https://tools.ietf.org/html/draft-ietf-spring-segment-routing-policy-01>





# New Sub-TLVs (Policy Name)

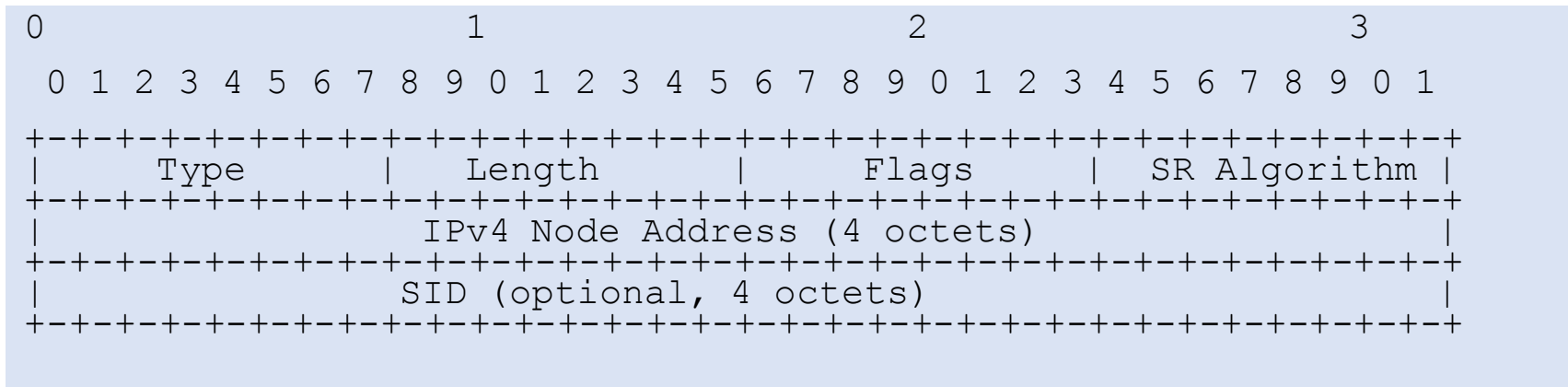
- Policy Name sub-TLV
- An operator MAY set the Policy Name sub-TLV to attach a symbolic name to the SR Policy candidate path
- Reference - section 2 in <https://tools.ietf.org/html/draft-ietf-spring-segment-routing-policy-01>





# SR Algorithm

- SR Flex Algorithm Flag (A-Flag)
- An operator MAY signal this flag with the Segment Type and supply a SR Algorithm ID.
- Reference - section 4 in <https://tools.ietf.org/html/draft-ietf-spring-segment-routing-policy-01>



# New Segments

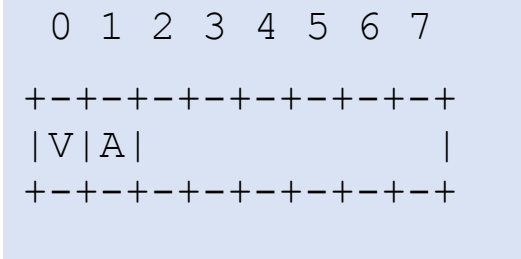
- Type 9: IPv6 Node Address with optional SID for SRv6
- Type 10: IPv6 Address + index for local and remote pair with optional SID for SRv6
- Type 11: IPv6 Local and Remote addresses for SRv6
- Reference - section 4 in <https://tools.ietf.org/html/draft-ietf-spring-segment-routing-policy-01>

## Type 9:

```
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      | SR Algorithm |
+-+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
//                IPv6 Node Address (16 octets)                //
+-+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
//                SID (optional, 16 octets)                //
+-+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--
```

# Segment Flags

- Segment Flags encode the per Segment behavior
- V-Flag : Segment Verification is performed on Head-end
  - Reference - section 5 in <https://tools.ietf.org/html/draft-ietf-spring-segment-routing-policy-01>
- A-Flag: SR Flex Algorithm is used for SID calculation
  - Reference - section 4 in <https://tools.ietf.org/html/draft-ietf-spring-segment-routing-policy-01>



# Binding SID Flags

- S-Flag : Enable “Specified-BSID-only” behavior on Head-End
  - Reference - section 6 in <https://tools.ietf.org/html/draft-ietf-spring-segment-routing-policy-01>
- I-Flag: Enable “Drop Upon Invalid” behavior on Head-End
  - Reference - section 8 in <https://tools.ietf.org/html/draft-ietf-spring-segment-routing-policy-01>

```
 0 1 2 3 4 5 6 7
+--+--+--+--+--+--+
|S|I|          |
+--+--+--+--+--+--+
```