

Networking Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: November 13, 2016

L. Ginsberg  
P. Psenak  
S. Previdi  
Cisco Systems  
M. Pilka  
Pantheon Technologies  
May 12, 2016

**Segment Routing Conflict Resolution**  
**draft-ietf-spring-conflict-resolution-00.txt**

**Abstract**

In support of Segment Routing (SR) routing protocols advertise a variety of identifiers used to define the segments which direct forwarding of packets. In cases where the information advertised by a given protocol instance is either internally inconsistent or conflicts with advertisements from another protocol instance a means of achieving consistent forwarding behavior in the network is required. This document defines the policies used to resolve these occurrences.

**Requirements Language**

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC 2119](#) [[RFC2119](#)].

**Status of This Memo**

This Internet-Draft is submitted in full conformance with the provisions of [BCP 78](#) and [BCP 79](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on November 13, 2016.

## Copyright Notice

Copyright (c) 2016 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](#) and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

<a href="#">1.</a>	<a href="#">Introduction</a>	<a href="#">2</a>
<a href="#">2.</a>	<a href="#">SR Global Block Inconsistency</a>	<a href="#">3</a>
<a href="#">3.</a>	<a href="#">Segment Identifier Conflicts</a>	<a href="#">5</a>
<a href="#">3.1.</a>	<a href="#">Conflict Types</a>	<a href="#">5</a>
<a href="#">3.1.1.</a>	<a href="#">Prefix Conflict</a>	<a href="#">6</a>
<a href="#">3.1.2.</a>	<a href="#">SID Conflict</a>	<a href="#">7</a>
<a href="#">3.2.</a>	<a href="#">Processing conflicting entries</a>	<a href="#">8</a>
<a href="#">3.2.1.</a>	<a href="#">Policy: Ignore conflicting entries</a>	<a href="#">8</a>
<a href="#">3.2.2.</a>	<a href="#">Policy: Preference Algorithm/Quarantine</a>	<a href="#">9</a>
<a href="#">3.2.3.</a>	<a href="#">Policy: Preference algorithm/ignore overlap only</a>	<a href="#">9</a>
<a href="#">3.2.4.</a>	<a href="#">Preference Algorithm</a>	<a href="#">9</a>
<a href="#">3.2.5.</a>	<a href="#">Example Behavior</a>	<a href="#">10</a>
<a href="#">3.2.6.</a>	<a href="#">Evaluation of Policy Alternatives</a>	<a href="#">11</a>
<a href="#">3.2.7.</a>	<a href="#">Guaranteeing Database Consistency</a>	<a href="#">11</a>
<a href="#">4.</a>	<a href="#">Security Considerations</a>	<a href="#">12</a>
<a href="#">5.</a>	<a href="#">IANA Consideration</a>	<a href="#">12</a>
<a href="#">6.</a>	<a href="#">Acknowledgements</a>	<a href="#">12</a>
<a href="#">7.</a>	<a href="#">References</a>	<a href="#">12</a>
<a href="#">7.1.</a>	<a href="#">Normative References</a>	<a href="#">12</a>
<a href="#">7.2.</a>	<a href="#">Informational References</a>	<a href="#">13</a>
	<a href="#">Authors' Addresses</a>	<a href="#">13</a>

## [1. Introduction](#)

Segment Routing (SR) as defined in [\[SR-ARCH\]](#) utilizes forwarding instructions called "segments" to direct packets through the network. Depending on the forwarding plane architecture in use, routing protocols advertise various identifiers which define the permissible values which can be used as segments, which values are assigned to specific prefixes, etc. Where segments have global scope it is necessary to have non-conflicting assignments - but given that the



advertisements may originate from multiple nodes the possibility exists that advertisements may be received which are either internally inconsistent or conflicting with advertisements originated by other nodes. In such cases it is necessary to have consistent resolution of conflicts network-wide in order to avoid forwarding loops.

The problem to be addressed is protocol independent i.e., segment related advertisements may be originated by multiple nodes using different protocols and yet the conflict resolution **MUST** be the same on all nodes regardless of the protocol used to transport the advertisements.

The remainder of this document defines conflict resolution policies which meet these requirements. All protocols which support SR **MUST** adhere to the policies defined in this document.

## **2. SR Global Block Inconsistency**

In support of an MPLS dataplane routing protocols advertise an SR Global Block (SRGB) which defines a set of label ranges reserved for use by the advertising node in support of SR. The details of how protocols advertise this information can be found in the protocol specific drafts e.g., [[SR-OSPF](#)], [[SR-OSPFv3](#)], and [[SR-IS-IS](#)]. However the protocol independent semantics are illustrated by the following example:



The originating router advertises the following ranges:

```
Range 1: (100, 199)
Range 2: (1000, 1099)
Range 3: (500, 5990)
```

The receiving routers concatenate the ranges and build the Segment Routing Global Block (SRGB) as follows:

```
SRGB = (100, 199)
       (1000, 1099)
       (500, 599)
```

The indices span multiple ranges:

```
index=0 means label 100
...
index 99 means label 199
index 100 means label 1000
index 199 means label 1099
...
index 200 means label 500
...
```

Note that the ranges are an ordered set - what labels are mapped to a given index depends on the placement of a given label range in the set of ranges advertised.

For the set of ranges to be usable the ranges MUST be disjoint. Sender behavior is defined in various SR protocol drafts such as [SR-IS-IS] which specify that senders MUST NOT advertise overlapping ranges.

Receivers of SRGB ranges MUST validate the SRGB ranges advertised by other nodes. If the advertised ranges do not conform to the restrictions defined in the respective protocol specification receivers MUST ignore all advertised SRGB ranges from that node. Operationally the node is treated as though it did not advertise any SRGB ranges. [[SR-MPLS](#)] defines the procedures for mapping global SIDs to outgoing labels.

Note that utilization of local SIDs (e.g. adjacency SIDs) advertised by a node is not affected by the state of the advertised SRGB.



### 3. Segment Identifier Conflicts

In support of an MPLS dataplane Segment identifiers (SIDs) are advertised and associated with a given prefix. SIDs may be advertised in the prefix reachability advertisements originated by a routing protocol. SIDs may also be advertised by a Segment Routing Mapping Server (SRMS).

Mapping entries have an explicit context which includes the topology and the SR algorithm. A generalized mapping entry can be represented using the following definitions:

Pi - Initial prefix  
Pe - End prefix  
L - Prefix length  
Lx - Maximum prefix length (32 for IPv4, 128 for IPv6)  
Si - Initial SID value  
Se - End SID value  
R - Range value  
T - Topology  
A - Algorithm

A Mapping Entry is then the tuple: (Pi/L, Si, R, T, A)

$Pe = (Pi + ((R-1) \ll (Lx-L)))$

$Se = Si + (R-1)$

Note that the SID advertised in a prefix reachability advertisement can be more generally represented as a mapping entry with a range of 1.

Conflicts in SID advertisements may occur as a result of misconfiguration. Conflicts may occur either in the set of advertisements originated by a single node or between advertisements originated by different nodes. When conflicts occur, it is not possible for routers to know which of the conflicting advertisements is "correct". If a router chooses to use one of the conflicting entries forwarding loops and/or blackholes may result unless it can be guaranteed that all other routers in the network make the same choice. Making the same choice requires that all routers have identical sets of advertisements and that they all use the same selection algorithm.

#### 3.1. Conflict Types

Various types of conflicts may occur.



### **3.1.1. Prefix Conflict**

When different SIDs are assigned to the same prefix we have a "prefix conflict". Prefix conflicts are specific to mapping entries sharing the same topology and algorithm. Consider the following set of advertisements:

```
(192.0.2.120/32, 200, 1, 0, 0)
(192.0.2.120/32, 30, 1, 0, 0)
```

The prefix 192.0.2.120/32 has been assigned two different SIDs:  
200 by the first advertisement  
30 by the second advertisement

```
(2001:DB8::1/128, 400, 1, 2, 0)
(2001:DB8::1/128, 50, 1, 2, 0)
```

The prefix 2001:DB8::1/128 has been assigned two different SIDs:  
400 by the first advertisement  
50 by the second advertisement

Prefix conflicts may also occur as a result of overlapping prefix ranges. Consider the following set of advertisements:

```
(192.0.2.1/32, 200, 200, 0, 0)
(192.0.2.121/32, 30, 10, 0, 0)
```

Prefixes 192.0.2.121/32 - 192.0.2.130/32 are assigned two different SIDs:  
320 through 329 by the first advertisement  
30 through 39 by the second advertisement

```
(2001:DB8::1/128, 400, 200, 2, 0)
(2001:DB8::121/128, 50, 10, 2, 0)
```

Prefixes 2001:DB8::121/128 - 2001:DB8::130/128 are assigned two different SIDs:  
420 through 429 by the first advertisement  
50 through 59 by the second advertisement

The second example illustrates a complication - only part of the range advertised in the first advertisement is in conflict. It is logically possible to isolate the conflicting portion and try to use the non-conflicting portion(s) at the cost of increased implementation complexity.

A variant of the overlapping prefix range is a case where we have overlapping prefix ranges but no actual SID conflict.



```
(192.0.2.1/32, 200, 200, 0, 0)
(192.0.2.121/32, 320, 10, 0, 0)

(2001:DB8::1/128, 400, 200, 2, 0)
(2001:DB8::121/128, 520, 10, 2, 0)
```

Although there is prefix overlap between the two IPv4 entries (and the two IPv6 entries) the same SID is assigned to all of the shared prefixes by the two entries.

Given two mapping entries:

$(P1/L1, S1, R1, T1, A1)$  and  $(P2/L2, S2, R2, T2, A2)$  where  $P1 \leq P2$

a prefix conflict exists if all of the following are true:

- 1)  $(T1 == T2) \ \&\& \ (A1 == A2)$
- 2)  $P1 \leq P2$
- 3) The prefixes are in the same address family.
- 2)  $L1 == L2$
- 3)  $(P1e \geq P2) \ \&\& \ ((S1 + (P2 - P1)) \neq S2)$

### **3.1.2. SID Conflict**

When the same SID has been assigned to multiple prefixes we have a "SID conflict". SID conflicts are independent of address-family, independent of prefix len, independent of topology, and independent of algorithm. A SID conflict occurs when a mapping entry which has previously been checked to have no prefix conflict assigns one or more SIDs that are assigned by another entry which also has no prefix conflicts. Consider the following examples:

```
(192.0.2.1/32, 200, 1, 0, 0)
(192.0.2.222/32, 200, 1, 0, 0)
SID 200 has been assigned to 192.0.2.1/32 by the
first advertisement.
The second advertisement assigns SID 200 to 192.0.2.222/32.
```

```
(2001:DB8::1/128, 400, 1, 2, 0)
(2001:DB8::222/128, 400, 1, 2, 0)
SID 400 has been assigned to 2001:DB8::1/128 by the
first advertisement.
The second advertisement assigns SID 400 to 2001:DB8::222/128
```



SID conflicts may also occur as a result of overlapping SID ranges. Consider the following sets of advertisements:

```
(192.0.2.1/32, 200, 200, 0, 0)
(198.51.100.1/32, 300, 10, 0, 0)
```

SIDs 300 - 309 have been assigned to two different prefixes. The first advertisement assigns these SIDs to 192.0.2.101/32 - 192.0.2.110/32. The second advertisement assigns these SIDs to 198.51.100.1/32 - 198.51.100.10/32.

```
(2001:DB8::1/128, 400, 200, 2, 0)
(2001:DB8:1::1/128, 500, 10, 2, 0)
```

SIDs 500 - 509 have been assigned to two different prefixes. The first advertisement assigns these SIDs to 2001:DB8::101/128 - 2001:DB8::10A/128. The second advertisement assigns these SIDs to 2001:DB8:1::1/128 - 2001:DB8:1::A/128.

The second example illustrates a complication - only part of the range advertised in the first advertisement is in conflict.

### **3.2. Processing conflicting entries**

Two general approaches can be used to process conflicting entries.

1. Conflicting entries can be ignored
2. A standard preference algorithm can be used to choose which of the conflicting entries will be used

The following sections discuss these two approaches in more detail.

Note: This document does not discuss any implementation details i.e. what type of data structure is used to store the entries (trie, radix tree, etc.) nor what type of keys may be used to perform lookups in the database.

#### **3.2.1. Policy: Ignore conflicting entries**

In cases where entries are in conflict none of the conflicting entries are used i.e., the network operates as if the conflicting advertisements were not present.



Implementations are required to identify the conflicting entries and ensure that they are not used.

### **3.2.2. Policy: Preference Algorithm/Quarantine**

For entries which are in conflict properties of the conflicting advertisements are used to determine which of the conflicting entries are used in forwarding and which are "quarantined" and not used. The entire quarantined entry is not used.

This approach requires that conflicting entries first be identified and then evaluated based on a preference rule. Based on which entry is preferred this in turn may impact what other entries are considered in conflict i.e. if A conflicts with B and B conflicts with C - it is possible that A does NOT conflict with C. Hence if as a result of the evaluation of the conflict between A and B, entry B is not used the conflict between B and C will not be detected.

### **3.2.3. Policy: Preference algorithm/ignore overlap only**

A variation of the preference algorithm approach is to quarantine only the portions of the less preferred entry which actually conflicts. The original entry is split into multiple ranges. The ranges which are in conflict are quarantined. The ranges which are not in conflict are used in forwarding. This approach adds complexity as the relationship between the derived sub-ranges of the original mapping entry have to be associated with the original entry - and every time some change to the advertisement database occurs the derived sub-ranges have to be recalculated.

### **3.2.4. Preference Algorithm**

The following algorithm is used to select the preferred mapping entry when a conflict exists. Evaluation is made in the order specified.

1. Smaller range wins
2. IPv6 entry wins over IPv4 entry
3. Smaller algorithm wins
4. Smaller prefix length wins
5. Smaller starting address (considered as an unsigned integer value) wins
6. Smaller starting SID wins



Using smaller range as the highest priority tie breaker makes advertisements with a range of 1 the most preferred. This associates a high priority to SID advertisements associated with protocol prefix advertisements as these always have an implicit range of one. SR mapping server advertisements (SRMS entries) may have any configured range - but in cases where they have a range greater than 1 they will be less preferred as compared to any SIDs in prefix advertisements. This has the nice property that a single misconfiguration of an SRMS entry with a large range will not be preferred over a large number of SIDs advertised in prefix reachability advertisements.

### 3.2.5. Example Behavior

The following mapping entries exist:in the database. For brevity, Topology/Algorithm is omitted and assumed to be (0,0) in all entries.

1. (192.0.2.1/32, 100, 1)
2. (198.51.100.1/32, 200, 1)
3. (203.0.113.1/32, 400, 300) !Prefix conflict with entries 1 and 2
4. (198.51.100.40/32, 200,1) !SID conflict with entry 2

The table below shows what mapping entries will be used in the forwarding plane (Active) and which ones will not be used (Excluded) unde rthe three candidate policies:

Policy	Active Entries	Excluded Entries
Ignore		(192.0.2.1/32,100,1) (198.51.100.1/32,200,1) (203.0.113.1/32,400,300) (199.51.100.40/32,200,1)
Quarantine	(192.0.1.1/32,100,1) (198.51.100.1/32,200,1)	(203.0.113.1/32,400,300) (198.51.100.40/32,100,1)
Overlap-Only	(192.0.2.1/32,100,1) (198.51.100.1/32,200,1) *(203.0.113.2/32,401,255) *(203.0.113.3/32,656,43)	(198.51.100.40/32,200,1) *(203.0.113.1/32,400,1) *(203.0.114.2/32,655,1)

\* Derived from (192.0.1.1/32,400,300)



### **3.2.6. Evaluation of Policy Alternatives**

The previous sections have defined three alternatives for resolving conflicts - ignore, quarantine, and ignore overlap-only.

The ignore policy impacts the greatest amount of traffic as forwarding to all destinations which have a conflict is affected.

Quarantine allows forwarding for some destinations which have a conflict to be supported. The bias is for mapping entries with the smallest range (typically - but not exclusively SIDs advertised in prefix reachability advertisements) to be forwarded while the destinations included in mapping entries with a larger range but NOT covered by entries with a smaller range will not be forwarded.

Ignore overlap-only maximizes the destinations which will be forwarded as all destinations covered by some mapping entry (regardless of range) will be able to use the SID assigned by the winning range. This alternative increases implementation complexity as compared to quarantine. Mapping entries with a range greater than 1 which are in conflict with mapping entries having a smaller range have to internally be split into 2 or more "derived mapping entries". The derived mapping entries then fall into two categories - those that are in conflict with a mapping entry of smaller range - and those which are NOT in conflict with an entry with smaller range. The former are ignored and the latter are used. Each time the underlying mapping database is updated the derived entries have to be recomputed based on the updated database. Internal data structures have to be maintained which maintain the relationship between the advertised mapping entry and the set of derived mapping entries. All nodes in the network have to achieve the same behavior regardless of implementation internals.

There is then a tradeoff between a goal of maximizing traffic delivery and the risks associated with increased implementation complexity.

It is the opinion of the authors that "quarantine" is the best alternative.

### **3.2.7. Guaranteeing Database Consistency**

In order to obtain consistent active entries all nodes in a network MUST have the same mapping entry database. Mapping entries can be obtained from a variety of sources.



- o SIDs can be configured locally for prefixes assigned to interfaces on the router itself. Only SIDs which are advertised to protocol peers can be considered as part of the nmapping entry database.
- o SIDs can be received in prefix reachability advertisements from protocol peers. These advertisements may originate from peers local to the area or be leaked from other areas and/or redistributed from other routing protocols.
- o SIDs can be received from SRMS advertisements - these advertisements can originate from routers local to the area or leaked from other areas
- o In cases where multiple routing protocols are in use mapping entries advertised by all routing protocols MUST be included.

#### **4. Security Considerations**

TBD

#### **5. IANA Consideration**

This document has no actions for IANA.

#### **6. Acknowledgements**

The authors would like to thank Jeff Tantsura, Wim Henderickx, and Bruno Decraene for their careful review and content suggestions..

#### **7. References**

##### **7.1. Normative References**

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), DOI 10.17487/RFC2119, March 1997, <<http://www.rfc-editor.org/info/rfc2119>>.
- [SR-IS-IS] "IS-IS Extensions for Segment Routing, [draft-ietf-isis-segment-routing-extensions-06](#)(work in progress)", December 2015.
- [SR-MPLS] "Segment Routing with MPLS dataplane, [draft-ietf-spring-segment-routing-mpls-04](#)(work in progress)", March 2016.



[SR-OSPF] "OSPF Extensions for Segment Routing, [draft-ietf-ospf-segment-routing-extensions-08](#)(work in progress)", May 2016.

[SR-OSPFv3] "OSPFv3 Extensions for Segment Routing, [draft-ietf-ospf-ospfv3-segment-routing-extensions-05](#)(work in progress)", March 2016.

## **7.2. Informational References**

[SR-ARCH] "Segment Routing Architecture, [draft-ietf-spring-segment-routing-08](#)(work in progress)", May 2016.

### Authors' Addresses

Les Ginsberg  
Cisco Systems  
510 McCarthy Blvd.  
Milpitas, CA 95035  
USA

Email: ginsberg@cisco.com

Peter Psenak  
Cisco Systems  
Apollo Business Center Mlynske nivy 43  
Bratislava 821 09  
Slovakia

Email: ppsenak@cisco.com

Stefano Previdi  
Cisco Systems  
Via Del Serafico 200  
Rome 0144  
Italy

Email: sprevidi@cisco.com

Martin Pilka  
Pantheon Technologies

Email: martin@pantheon.tech

