Networking Working Group
Internet-Draft
Intended status: Standards Track
Expires: April 16, 2016

L. Ginsberg
P. Psenak
S. Previdi
Cisco Systems
October 14, 2015

Segment Routing Conflict Resolution
draft-ginsberg-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

Table of Contents

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] 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 indexes 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. The question then arises what receiving routers should do if they receive an SRGB which includes overlapping ranges. In such a case the following rule is defined:

Each range is examined in the order it was advertised. If it does not overlap with any advertised range which preceded it the advertised range is used. If the range overlaps with any preceding range it MUST NOT be used and all ranges advertised after the first encountered overlapping range also MUST NOT be used.

Consider the following example:

The originating router advertises the following ranges:

```
Range 1: (100, 199]
Range 2: (1000, 1099)
Range 3: (100, 599)
Range 4: (2000, 2099)
```

Range 3 overlaps with Range 1.
Only Ranges #1 and #2 are usable.
Ranges #3 and #4 are ignored.
Note that Range #4 is not used even though it does not overlap with any of the other ranges.

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).

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

Mapping Entry is then the tuple: (Pi/L, Si, R)
Pe = (Pi + ((R-1) << (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". Consider the following set of advertisements:

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

The prefix 192.0.2.120/32 has been assigned two different SIDs - 200 by the first advertisement - 30 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)
(192.0.2.121/32, 30, 10)

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.

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. The algorithm defined here does NOT attempt to support use of a partial range.

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)
(192.0.2.121/32, 320, 10)

Although there is prefix overlap between the two entries the same SID is assigned to all of the shared prefixes by the two entries. It is possible to utilize both entries but it complicates the implementation of the database required to support this. See

Appendix A for a more complete discussion of this case. An alternative is to ensure at the nodes which originate these advertisements that no such overlap is allowed to be configured. Such overlaps can then be considered as a conflict if they are received. This allows a simpler and more efficient implementation of the database. This is the approach assumed in this document.

Given two mapping entries:

(P1/L1, S1, R1) and (P2/L2, S2, R2)

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

1)The prefixes are in the same address family.
2)L1 == L2
3)((P1 < P2) && (P1e >= P2)) || ((P2 < P1) && (P2e >= P1))

3.1.2. SID Conflict

When the same SID has been assigned to multiple prefixes we have a "SID conflict". Consider the following example:

(192.0.2.1/32, 200, 1)
(192.0.2.222/32, 200,1)

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.

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

(192.0.2.1/32, 200, 200)
(192.1.2.1/32, 300, 10)

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 192.1.2.1/32 - 192.1.2.10/32.

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. The algorithm defined here does NOT attempt to support use of a partial range.

SID conflicts are independent of address-family and independent of prefix len. 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.

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. 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.

Ikplementation requires identifying the conflicting entries and ensuring that they are not used. The occurrence of conflicts is easily diagnosed from the behavior of the network as the forwarding of traffic which would, in the absence of conflicts, utilize segments no longer does so. Which prefixes are impacted is easily seen and therefore the entries which are misconfigured are easily identified. Unintended traffic flow will never occur.

The downside of ignoring conflicting entries is that forwarding of all packets with destinations covered by the conflicting entries will always be negatively impacted.

3.2.2. Preference Algorithm

For entries which are in conflict properties of the advertisement (e.g. prefix value, prefix length, SID value, etc.) are used to determine which of the conflicting entries are used in forwarding and which are ignored.

This approach requires that conflicting entries first be identified and then evaluated based on the 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 considered.

As at least some of the traffic continues to be forwarded after the conflict is detected, the presence of the conflict may be harder to diagnose based on traffic flowthan when using the ignore policy.

The upside of the preference algorithm is that in some cases forwarding of traffic may continue to be correct despite the existence of the conflict. If the preference algorithm happens to prefer the intended configuration traffic will still be successfully delivered. Whether this will occur is a random outcome since the preference algorithm cannot know which of the conflicting entries is the "correct" entry.

3.2.3. Candidate Preference Algorithm

The following algorithm is proposed. Evaluation is made in the order specified.

1. IPv4 entry wins over IPv6 entry
2. Smaller prefix length wins
3. Smaller starting address (considered as an unsigned integer value) wins
4. Smaller starting SID wins
5. Smaller range wins
6. non-attached entries preferred over attached entries (SRMS attached flag)

3.2.4. Example Behavior

Consider the following simple case. The following mapping entries exist:

1. (192.0.2.1/32, 100, 200)
2. (192.0.2.200/32, 150, 300) !Prefix conflict with entry #1

3. (193.3.3.3/32, 400, 100) !SID conflict with entry #2

Using the Ignore conflicts behavior we would not use any of the above entries.

Using a preference rule which favors smaller prefixes, entries #1 and #3 would be used.

- Entry #1 would be used as 192.0.200.1 is less than 192.0.2.200.
- Entry #3 would be used because once Entry #2 has been excluded, entry #3 no longer conflicts with any entry which is being used. (An example of lack of transitivity of conflicts)

If we now add

4. (192.0.1.1/32, 50, 100) ! Prefix conflict with #1

Using ignore policy still none of the entries would be used.

Using a preference rule which favors smaller prefixes , entries #4 and #2 would be used.

- Entry #4 would be used in preference to entry #1 because 192.0.1.1 < 192.0.2.1.
- Entry #2 would be used because once Entry #1 is excluded entry #2 no longer has a prefix conflict with any active entry.
- Entry #3 would NOT be used because once Entry #2 becomes active entry #3 loses due to the SID conflict with Entry #2 since the latter has a smaller prefix.

3.2.5. Other Preference Factors to consider

Prefix to SID mapping is based on a variety of sources.

- SIDs can be configured locally for prefixes assigned to interfaces on the router itself
- 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
- SIDs can be received from SRMS advertisements - these advertisements can originate from routers local to the area or leaked from other areas

SIDs configured locally for prefixes associated with interfaces on the router itself are only used by the originating router in prefix advertisements - they are not installed in the forwarding plane locally. Therefore, they do not need to be considered in conflict resolution.

For other sources, It may seem intuitive to assign priority based on point of origination (e.g. intra-area preferred over inter-area, prefix reachability advertisements preferred over SRMS advertisements, etc.). However, any such policy makes it more likely that inconsistent choices will be made by routers in the network and increase the likelihood of forwarding loops or blackholes. The algorithms defined in this document assume that prefix reachability advertisements are part of the set of entries considered when determining conflicts and conflict resolution and no preference is associated with prefix reachability advertisements over SRMS advertisements.

It is common to use the identity of the advertising source router (e.g. router ID) as a tie breaker. However, in the case of SID advertisements it is possible that the source ID is not known. For example, when leaking SRMS advertisements the source ID may appear to be the Area Border Router (ABR) which performed the leaking. But this means that the relative preference of the SIDs associated with the leaked advertisements will have a different priority in different areas. Therefore router ID is not used in the algorithms discussed above.

4. IANA Considerations

None.

5. Security Considerations

TBD

6. Acknowledgements

The authors would like to thank Martin Pilka for sharing his knowledge of algorithm implementation and complexity.

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>.

7.2. Informational References

[SR-ARCH] "Segment Routing Architecture, draft-ietf-spring-segment-routing-05(work in progress)", September 2015.

[SR-IS-IS] "IS-IS Extensions for Segment Routing, draft-ietf-isis-segment-routing-extensions-05(work in progress)", June 2015.

[SR-OSPF] "OSPF Extensions for Segment Routing, draft-ietf-ospf-segment-routing-extensions-05(work in progress)", June 2015.

Appendix A. Alternate Prefix Conflict Algorithm

It is possible when encountering overlapping prefix ranges to allow use of such entries when there is no actual SID conflict. This case can be avoided if configuration of such entries is blocked at the source - which allows a far simpler implementation when processing received entries. The latter model is what is described in the body of this document. What is described below is the algorithm and consequences of trying to use overlapping ranges when the SID assignments do not conflict.

The algorithm used to determine if two entries are in conflict is as follows.

Given two mapping entries:

(P1/L1, S1, R1) and (P2/L2, S2, R2)

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

1)The prefixes are in the same address family.
2)L1 == L2
3)(P1 < P2) && (P1e >= P2) && (((P2-P1)>>(Lx-L)) + S1) != S2)
   ||
  (P2 < P1) && (P2e >= P1) && (((P1-P2)>>(Lx-L)) + S2) != S1)

From an implementation standpoint, the complexity comes in supporting lookup of the SID for a given prefix in the presence of overlapping ranges which are in use. Consider the following simple example:

Entry #1: (192.0.2.1/32, 100, 250)

Entry #2: (192.0.2.101/32, 200, 10)

Entry #3: (192.0.2.201/32, 300, 100)

Using the conflict detection algorithm described in this section there is no conflict in the three ranges since all prefixes which overlap between the entries are assigned the same SID in all ranges.

If however we want to find the SID for the prefix 192.0.2.150, here are the steps one might go through:

1)Find the entry with the largest value of starting prefix which is less than or equal to 192.0.2.150. In the example this would be Entry #2 above.

2)Determine if the range covers 192.0.2.150. In the example the answer to this would be "no".

If we were not required to support overlapping entries at this point we would be done and conclude that no SID was assigned. But because we know that we may have overlapping entries we have to walk backwards (conceptually) and look at all of the entries with starting prefixes (of prefix length 32) which are less than 192.0.2.150 and check if their range is large enough to include that prefix. In the presence of a large # of entries this would be very slow. To avoid this problem we could build a local entry which contained all of overlapping entries (in this example all three of the entries shown) and use that in our actiuve database to do the lookups. In this example we would generate:

(192.0.2.1/32, 100, 300)

Then we could use steps #1 and #2 above and be confident in the answer we get.

However, since we are no longer using the entries we received in our active database we would need to maintain a link between the generated entry and the set of overlapping entries we received which caused its generation so that if one of those entries was removed or modified we could properly update our active database.

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