

SPRING Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: December 26, 2015

C. Bowers  
P. Sarkar  
H. Gredler  
Juniper Networks  
June 24, 2015

Advertising Per-Algorithm Label Blocks  
draft-bowers-spring-adv-per-algorithm-label-blocks-01

## Abstract

Segment routing uses globally-known labels to accomplish destination-based forwarding along shortest paths computed using Dijkstra's algorithm with IGP metrics. This draft discusses how to use segment routing to accomplish destination-based forwarding along paths computed using other algorithms and metrics. In particular, the draft contrasts two different options for associating labels with different algorithms for computing forwarding next-hops.

## 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 December 26, 2015.

## Copyright Notice

Copyright (c) 2015 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>	Introduction . . . . .	<a href="#">2</a>
<a href="#">2.</a>	Destination-based forwarding in segment routing using other algorithms . . . . .	<a href="#">3</a>
<a href="#">3.</a>	Practical implications of using the per-algorithm node index option . . . . .	<a href="#">4</a>
<a href="#">4.</a>	Advantages of using the per-algorithm label block option . .	<a href="#">5</a>
<a href="#">5.</a>	ISIS extensions to encode per-algorithm label blocks . . . .	<a href="#">5</a>
<a href="#">6.</a>	IANA Considerations . . . . .	<a href="#">6</a>
<a href="#">7.</a>	Management Considerations . . . . .	<a href="#">6</a>
<a href="#">8.</a>	Security Considerations . . . . .	<a href="#">6</a>
<a href="#">9.</a>	Acknowledgements . . . . .	<a href="#">6</a>
<a href="#">10.</a>	Informative References . . . . .	<a href="#">6</a>
	Authors' Addresses . . . . .	<a href="#">7</a>

## [1.](#) Introduction

[I-D.ietf-spring-segment-routing] describes the segment routing architecture. In segment routing, LDP-like forwarding behavior along shortest paths is achieved using globally-known node labels. Globally-known node labels can be distributed in one of two ways. Each router can directly advertise a globally unique node label in the IGP. Or each router can advertise a globally unique node index value as well as a locally assigned label block, allowing any router in the IGP area to determine the mapping of locally-assigned label to globally unique node index for any other router in the area.

This draft focuses on the case where the combination of global node index and local label block is used to distribute locally-significant labels. Figure 1 shows the formula used to determine the locally-significant label values corresponding to shortest path forwarding next-hops computed using Dijkstra's shortest path first algorithm with IGP metrics, which is the default mode of operation for segment routing.

$$\text{SPF\_Label}(X,D) = \text{Label\_Block}(X) + \text{Node\_Index}(D)$$

X is the node for which the label has local significance

D is the destination node

Figure 1: Formula for determining locally-significant shortest path labels

As a simple example, when the computing node (Y) needs to forward a packet ultimately destined for node D, Y first determines the shortest path next-hop node to reach D, which we assume to be X in this example. Y then adds the Node\_Index value advertised by D to the Label\_Block value advertised by X to determine the label value to apply to the packet before sending it to X.

## [2.](#) Destination-based forwarding in segment routing using other algorithms

Figure 2 shows two options for generalizing the above formula, to determine locally-significant labels corresponding to forwarding next-hops computed using other algorithms.

Option 1: per-algorithm node index

$$\text{Label}(X,D,A) = \text{Label\_Block}(X) + \text{Node\_Index}(D,A)$$

Option 2: per-algorithm label block

$$\text{Label}(X,D,A) = \text{Label\_Block}(X,A) + \text{Node\_Index}(D)$$

X is the node for which the label has local significance

D is the destination node

A is the algorithm for computing destination-based forwarding next-hops

Figure 2: Two options for determining locally-significant labels for other algorithms

The computing router (Y) needs to forward a packet destined for node D using next-hops computed by algorithm A. Using either option, Y determines the next-hop computed by algorithm A to reach D, which we assume to be X in this example. Y then needs to figure out the correct label to apply to the packet so that X will also understand that the packet is destined for node D using forwarding next-hops

computed by algorithm A. The two options shown in Figure 2 differ in how Y determines that label value.

In Option 1 each node advertises a unique node index for each algorithm and a single label block. Y determines the label value of local significance to X to reach D using algorithm A by adding the Node\_Index advertised by node D for algorithm A to the Label\_Block advertised by node X. We refer to this as the per-algorithm node index option.

In Option 2 each node advertises a single node index and a unique label block for each algorithm. Y determines the label value of

local significance to X to reach D using algorithm A by adding the Node\_Index advertised by node D to the Label\_Block for algorithm A advertised by node X. We refer to this as the per-algorithm label block option.

The extensions currently defined in [\[I-D.ietf-isis-segment-routing-extensions\]](#) and [\[I-D.ietf-ospf-segment-routing-extensions\]](#) specify encodings for Option 1, the per-algorithm node index option. However, as illustrated in the following section, Option 2 has significant advantages over Option 1.

### [3.](#) Practical implications of using the per-algorithm node index option

The following example illustrates the practical difficulties associated with using the per-algorithm node index option to support other forwarding algorithms.

Suppose an operator has a network with 100 nodes, which we will refer to as R0-R99. The operator assigns the unique node index values 0-99 to those nodes for algorithm=0, in order to accomplish shortest path routing based on IGP metrics with SR labels. Each node will need to advertise a label block of size=100.

Assume that at some future point in time, the IETF defines algorithm=1 to mean shortest path routing based on latency, and vendors implement this. Suppose that the operator wants to use latency-based SPF routes for some traffic and metric-based SPF routes for other traffic. The operator will need to define a new set of

unique node index values for algorithm=1. A reasonable choice would be to assign node index values of 100-199 to R0-R99 for algorithm=1. Each node will now need to advertise a label block of size=200. So far the need for per-algorithm node index values is an annoyance, but not too difficult to deal with.

Now assume that the operator needs to add 10 new nodes to the SR domain, specifically nodes R100-R109. Each node will now need to advertise a label block of size=220. The main issue is deciding how to assign per-algorithm node index for the 10 new nodes? One option is to redo the node index numbering scheme so that R0-R109 have node index values 0-109 for algorithm=0 and node index values 110-229 for algorithm=1. However, this requires renumbering existing nodes. The other option is to avoid renumbering of nodes by assigning nodes R100-R109 node index values 200-209 for algorithm=0 and node index values 210-219 for algorithm=1. Each of these approaches has drawbacks. The first requires renumbering existing nodes, while the second is difficult to maintain since there is no obvious relationship between the node index values for different algorithms.

There are other numbering schemes for per-algorithm node index value that one can consider. However, they run into similar problems as described in the example above when trying to add more nodes or more algorithms. One could also try to mitigate these problems by anticipating growth in each dimension and pre-assigning node index values based on anticipated maximum values in each dimension. However, this can result in needing to advertise label blocks that are potentially much larger than actually needed.

#### [4.](#) Advantages of using the per-algorithm label block option

The use of per-algorithm label blocks avoids the problems associated with assigning and maintaining unique node index values for each forwarding algorithm.

When the SR domain is initially deployed, R0-R99 can be assigned node index values 0-99, as one would expect. When support for algorithm=1 gets added, the operator does not need to assign and configure any new node index values. Instead, the routers automate the process by advertising different label blocks for each forwarding algorithm.

When another 10 nodes are added to the SR domain, R100-R109 get

assigned node index values 100-109 as one would expect. And the router advertises a label block of size=110 for each algorithm, as one would expect. Adding new nodes in the presence of multiple forwarding algorithms is simplified significantly with the use of per-algorithm label blocks.

## 5. ISIS extensions to encode per-algorithm label blocks

Below is a concrete proposal for encoding per-algorithm label blocks in ISIS compatible with the encodings in [\[I-D.ietf-isis-segment-routing-extensions\]](#).

The newly-defined Algorithm-Label-Block sub-TLV is shown in Figure 3. It is carried in the IS-IS Router Capability TLV-242. It contains a 8-bit Algorithm field which associates the SRGB Descriptor entries carried in the sub-TLV with an algorithm for computing destination-based next-hops. Otherwise, the structure and interpretation of the Algorithm-Label-Block sub-TLV is identical to that of the SR-Capabilities sub-TLV defined in section 3.1 of [\[I-D.ietf-isis-segment-routing-extensions\]](#).

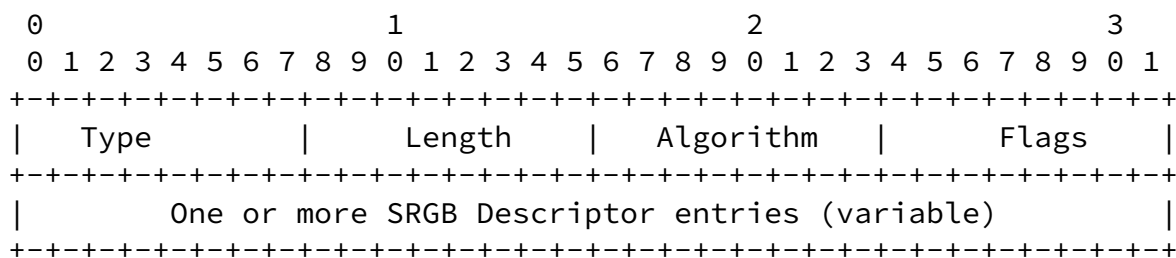


Figure 3: Algorithm-Label-Block sub-TLV

The Algorithm-Label-Block sub-TLV MUST NOT be advertised with Algorithm value=0. In this way, the concatenated label block used to compute the label values for algorithm=0 can only be carried by the SR Capabilities sub-TLV. When the Algorithm-Label-Block sub-TLV is

present with a given non-zero algorithm value, routers MUST determine the label values associated with that algorithm using the per-algorithm label block method and the concatenated label block carried by the Algorithm-Label-Block sub-TLV. The node indices used in this calculation are those carried in Node-SID advertisements with algorithm value=0.

## 6. IANA Considerations

This document introduces no new IANA Considerations.

## 7. Management Considerations

This document proposes the use of per-algorithm label blocks to support destination-based forwarding along next-hops computed using different algorithms. The automated advertisement of per-algorithm label blocks significantly simplifies network management compared to configuration and maintenance of unique per-algorithm node indexes.

## 8. Security Considerations

TBD

## 9. Acknowledgements

The authors thank John Scudder for his review of this document and his suggestions.

## 10. Informative References

[I-D.ietf-isis-segment-routing-extensions]

Previdi, S., Filsfils, C., Bashandy, A., Gredler, H., Litkowski, S., Decraene, B., and J. Tantsura, "IS-IS Extensions for Segment Routing", [draft-ietf-isis-segment-routing-extensions-04](#) (work in progress), May 2015.

[I-D.ietf-ospf-segment-routing-extensions]

Psenak, P., Previdi, S., Filsfils, C., Gredler, H.,

Shakir, R., Henderickx, W., and J. Tantsura, "OSPF Extensions for Segment Routing", [draft-ietf-ospf-segment-routing-extensions-04](#) (work in progress), February 2015.

[I-D.ietf-spring-segment-routing]

Filsfils, C., Previdi, S., Decraene, B., Litkowski, S., and R. Shakir, "Segment Routing Architecture", [draft-ietf-spring-segment-routing-03](#) (work in progress), May 2015.

#### Authors' Addresses

Chris Bowers  
Juniper Networks  
1194 N. Mathilda Ave.  
Sunnyvale, CA 94089  
US

Email: [cbowers@juniper.net](mailto:cbowers@juniper.net)

Pushpasis Sarkar  
Juniper Networks  
Embassy Business Park  
Bangalore, KA 560093  
India

Email: [psarkar@juniper.net](mailto:psarkar@juniper.net)

Hannes Gredler  
Juniper Networks  
1194 N. Mathilda Ave.  
Sunnyvale, CA 94089  
US

Email: [hannes@juniper.net](mailto:hannes@juniper.net)