

Inter-Domain Routing
Internet-Draft
Intended status: Standards Track
Expires: July 9, 2020

K. Talaulikar
P. Psenak
Cisco Systems
S. Zandi
G. Dawra
LinkedIn
January 6, 2020

Flexible Algorithm Definition Advertisement with BGP Link-State
draft-ietf-idr-bgp-ls-flex-algo-02

Abstract

Flexible Algorithm is a solution that allows routing protocols (viz. OSPF and IS-IS) to compute paths over a network based on user-defined (and hence, flexible) constraints and metrics. The computation is performed by routers participating in the specific network in a distribute manner using a Flex Algorithm definition. This definition provisioned on one or more routers and propagated (viz. OSPF and IS-IS flooding) through the network.

BGP Link-State (BGP-LS) enables the collection of various topology information from the network. This draft defines extensions to BGP-LS address-family to advertise the Flexible Algorithm Definition as a part of the topology information from the network.

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 <https://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 July 9, 2020.

Copyright Notice

Copyright (c) 2020 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 (<https://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

1.	Introduction	2
1.1.	Requirements Language	4
2.	BGP-LS Extensions for Flex Algo	4
3.	Flexible Algorithm Definition	4
3.1.	Flex Algo Exclude Any Affinity	6
3.2.	Flex Algo Include Any Affinity	6
3.3.	Flex Algo Include All Affinity	7
3.4.	Flex Algo Definition Flags	8
4.	Flex Algorithm Prefix Metric	9
5.	IANA Considerations	10
6.	Manageability Considerations	10
6.1.	Operational Considerations	10
6.2.	Management Considerations	11
7.	Security Considerations	11
8.	Acknowledgements	11
9.	References	11
9.1.	Normative References	11
9.2.	Informative References	12
	Authors' Addresses	13

[1.](#) Introduction

IGP protocols (OSPF and IS-IS) traditionally compute best paths over the network based on the IGP metric assigned to the links. Many network deployments use RSVP-TE [[RFC3209](#)] based or Segment Routing (SR) Policy [[I-D.ietf-spring-segment-routing-policy](#)] based solutions to enforce traffic over a path that is computed using different metrics or constraints than the shortest IGP path. [[I-D.ietf-lsr-flex-algo](#)] defines the Flexible Algorithm solution that allows IGPs themselves to compute constraint based paths over the network.

Flexible Algorithm is called so as it allows a user the flexibility to define

- o the type of calculation to be used (e.g. shortest path)
- o the metric type to be used (e.g. IGP metric or TE metric)
- o the set of constraints to be used (e.g. inclusion or exclusion of certain links using affinities)

The operations of the flexible algorithm solution is described in detail in [[I-D.ietf-lsr-flex-algo](#)] and a high level summary of the same is described here for clarity. The network operator enables the participation of specific nodes in the network for a specific algorithm and then provisions the definition of that flexible algorithm on one or more of these nodes. The nodes where the flexible algorithm definition (FAD) is advertised then flood these definitions via respective IGP (IS-IS and OSPFv2/v3) mechanisms to all other nodes in the network. The nodes select the definition for each algorithm based on the flooded information in a deterministic manner and thus all nodes participating in a flexible algorithm computation arrive at a common understanding of the type of calculation that they need to use.

When using Segment Routing (SR) [[RFC8402](#)] MPLS forwarding plane [[RFC8660](#)], the result of a flex algorithm computation is the provisioning of the Prefix SIDs associated with that algorithm with paths based on the topology computed based on that algorithm. When using SR forwarding plane on IPv6 (SRv6) [[I-D.ietf-spring-srv6-network-programming](#)], the result of a flex algorithm computation is the provisioning of the SRv6 Locators associated with that algorithm with paths based on the topology computed based on that algorithm. This flex algorithm computation is within an IGP area or level similar to the default shortest path tree (SPT) algorithm.

A flex algorithm specific metric MAY be advertised along with the prefix as described in [[I-D.ietf-lsr-flex-algo](#)] to enable end-to-end optimal path computation for prefixes across multiple areas/domains in the flex algorithm computation for the SR-MPLS forwarding plane.

The BGP-LS extensions for SR are defined in [[I-D.ietf-idr-bgp-ls-segment-routing-ext](#)] and [[I-D.ietf-idr-bgp-ls-srv6-ext](#)]. They include the

- o SR Algorithm TLV to indicate the participation of a node in a flex algorithm computation

- o Prefix SID TLV to indicate the association of the Prefix-SIDs to a specific flex algorithm for SR-MPLS forwarding
- o SRv6 Locator TLV to indicate the Locator for specific flex algorithm for SRv6 forwarding

Thus a controller or a Path Computation Engine (PCE) is aware of the IGP topology across multiple domains which includes the above information related to the flexible algorithm. This draft defines extensions to BGP-LS for carrying the Flexible Algorithm Definition information so that it enables the controller/PCE to learn the mapping of the flex algorithm number to its definition in each area/domain of the underlying IGP. The controller/PCE also learns the type of computation used and the constraints for the same. This information can then be leveraged by it for setting up SR Policy paths end to end across domains by leveraging the appropriate Flex Algorithm specific SIDs in the in its Segment List [[I-D.ietf-spring-segment-routing-policy](#)]. e.g. picking the Flex Algorithm Prefix SID (in case of SR-MPLS) or End SID (in case of SRv6) of ABRs/ASBRs corresponding to a definition that optimizes on the delay metric enables the PCE/controller to build an end to end low latency path across IGP domains with minimal SIDs in the SID list.

[1.1.](#) Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [BCP 14](#) [[RFC2119](#)] [[RFC8174](#)] when, and only when, they appear in all capitals, as shown here.

[2.](#) BGP-LS Extensions for Flex Algo

The BGP-LS [[RFC7752](#)] specifies the Node NLRI for advertisement of nodes along with their attributes using the BGP-LS Attribute and the Prefix NLRI for advertisement of prefixes along with their attributes using the BGP-LS Attribute. The Flexible Algorithm Definition (FAD) advertised by a node are considered as its node level attributes and advertised as such. The Flexible Algorithm Prefix Metric (FAPM) are considered as prefix attributes and advertised as such.

[3.](#) Flexible Algorithm Definition

This document defines a new optional BGP-LS Attribute TLV associated with the Node NLRI called the Flexible Algorithm Definition (FAD) TLV and its format is as follows:

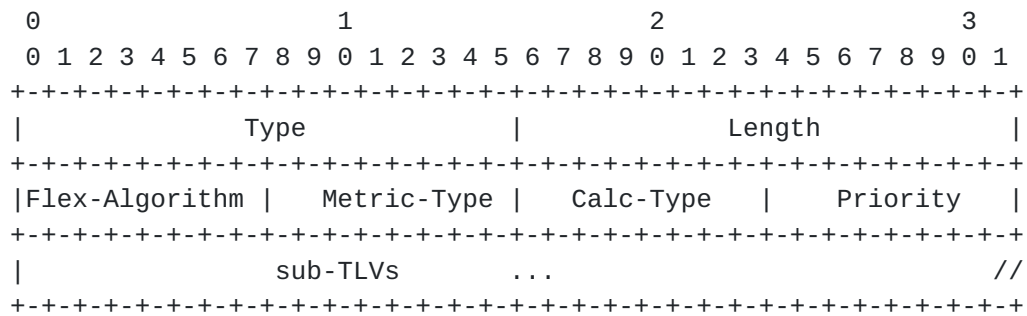


Figure 1: Flex Algorithm Definition TLV

where:

- o Type: 1039
- o Length: variable. Minimum of 4 octets.
- o Flex-Algorithm : 1 octet value in the range between 128 and 255 inclusive which is the range defined for Flexible Algorithms in the IANA "IGP Parameters" registries under the "IGP Algorithm Types" registry [[I-D.ietf-lsr-flex-algo](#)].
- o Metric-Type : 1 octet value indicating the type of the metric used in the computation. Values allowed come from the IANA "IGP Parameters" registries under the "Flexible Algorithm Definition Metric-Type" registry [[I-D.ietf-lsr-flex-algo](#)].
- o Calculation-Type : 1 octet value in the range between 0 and 127 inclusive which is the range defined for the standard algorithms in the IANA "IGP Parameters" registries under the "IGP Algorithm Types" registry [[I-D.ietf-lsr-flex-algo](#)].
- o Priority : 1 octet value between 0 and 255 inclusive that specifies the priority of the FAD.
- o sub-TLVs : zero or more sub-TLVs may be included as described further in this section.

The FAD TLV can only be added to the BGP-LS Attribute of the Node NLRI if the corresponding node originates the underlying IGP TLV/sub-TLV as described below. This information is derived from the protocol specific advertisements as below..

- o IS-IS, as defined by the ISIS Flexible Algorithm Definition sub-TLV in [\[I-D.ietf-lsr-flex-algo\]](#).

- o OSPFv2/OSPFv3, as defined by the OSPF Flexible Algorithm Definition TLV in [[I-D.ietf-lsr-flex-algo](#)].

The BGP-LS Attribute associated with a Node NLRI MAY include one or more FAD TLVs corresponding to the Flexible Algorithm Definition for each algorithm that the particular node is advertising.

The following sub-sections define the sub-TLVs for the FAD TLV.

[3.1.](#) Flex Algo Exclude Any Affinity

The Flex Algo Exclude Any Affinity sub-TLV is an optional sub-TLV that is used to carry the affinity constraints [[RFC2702](#)] associated with the flex algo definition and enable the exclusion of links carrying any of the specified affinities from the computation of the specific algorithm as described in [[I-D.ietf-lsr-flex-algo](#)]. The affinity is expressed in terms of Extended Admin Group (EAG) as defined in [[RFC7308](#)].

The TLV has the following format:

```

      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
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|                               |                               |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|                               |                               |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|                               |                               |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

where:

- o Type: 1040
- o Length: variable, dependent on the size of the Extended Admin Group. MUST be a multiple of 4 octets.
- o Exclude-Any EAG : the bitmask used to represent the affinities to be excluded.

The information in the Flex Algo Exclude Any Affinity sub-TLV is derived from the IS-IS and OSPF protocol specific Flexible Algorithm Exclude Admin Group sub-TLV as defined in [[I-D.ietf-lsr-flex-algo](#)].

[3.2.](#) Flex Algo Include Any Affinity

The Flex Algo Include Any Affinity sub-TLV is an optional sub-TLV that is used to carry the affinity constraints [[RFC2702](#)] associated with the flex algo definition and enable the inclusion of links carrying

any of the specified affinities in the computation of the specific algorithm as described in [[I-D.ietf-lsr-flex-algo](#)]. The affinity is expressed in terms of Extended Admin Group (EAG) as defined in [[RFC7308](#)].

The TLV has the following format:

```

      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               |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|                                     |                                     |
|               Include-Any EAG (variable)                               |//
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

where:

- o Type: 1041
- o Length: variable, dependent on the size of the Extended Admin Group. MUST be a multiple of 4 octets.
- o Include-Any EAG : the bitmask used to represent the affinities to be included.

The information in the Flex Algo Include Any Affinity sub-TLV is derived from the IS-IS and OSPF protocol specific Flexible Algorithm Include-Any Admin Group sub-TLV as defined in [[I-D.ietf-lsr-flex-algo](#)].

3.3. Flex Algo Include All Affinity

The Flex Algo Include All Affinity sub-TLV is an optional sub-TLV that is used to carry the affinity constraints [[RFC2702](#)] associated with the flex algo definition and enable the inclusion of links carrying all of the specified affinities in the computation of the specific algorithm as described in [[I-D.ietf-lsr-flex-algo](#)]. The affinity is expressed in terms of Extended Admin Group (EAG) as defined in [[RFC7308](#)].

The TLV has the following format:


```

      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                               |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|                               Include-All EAG (variable)          |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

where:

- o Type: 1042
- o Length: variable, dependent on the size of the Extended Admin Group. MUST be a multiple of 4 octets.
- o Include-All EAG : the bitmask used to represent the affinities to be included.

The information in the Flex Algo Include All Affinity sub-TLV is derived from the IS-IS and OSPF protocol specific Flexible Algorithm Include-All Admin Group sub-TLV as defined in [\[I-D.ietf-lsr-flex-algo\]](#).

3.4. Flex Algo Definition Flags

The Flex Algo Definition Flags sub-TLV is an optional sub-TLV that is used to carry the flags associated with the flex algo definition that are used in the computation of the specific algorithm as described in [\[I-D.ietf-lsr-flex-algo\]](#).

The TLV has the following format:

```

      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                               |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|                               Flags (variable)                  |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

where:

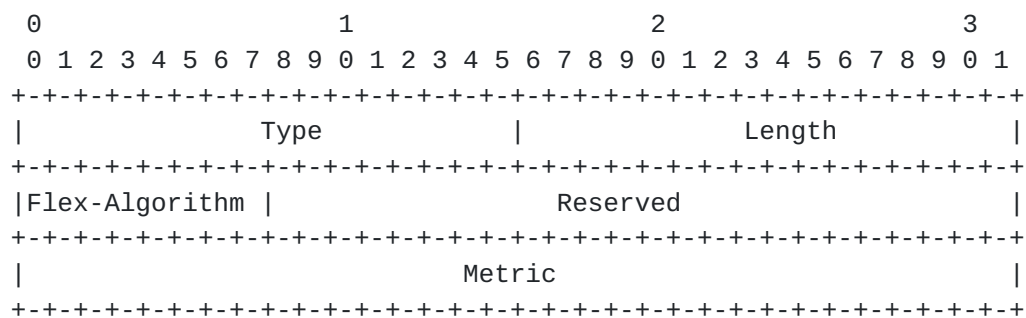
- o Type: 1043
- o Length: variable. MUST be a multiple of 4 octets.
- o Flags : the bitmask used to represent the flags for the flex algorithm definition as introduced by [\[I-D.ietf-lsr-flex-algo\]](#) and

listed in the "Flex-Algorithm Definition Flags" registry under the "Interior Gateway Protocol (IGP) Parameters" IANA registry.

The information in the Flex Algo Definition Flags sub-TLV is derived from the IS-IS and OSPF protocol specific Flexible Algorithm Definition Flags sub-TLV as defined in [[I-D.ietf-lsr-flex-algo](#)].

4. Flex Algorithm Prefix Metric

This document defines a new optional BGP-LS Attribute TLV associated with the Prefix NLRI called the Flexible Algorithm Prefix Metric (FAPM) TLV and its format is as follows:



where:

- o Type: 1044
- o Length: 8 octets.
- o Flex-Algorithm : 1 octet value in the range between 128 and 255 inclusive which is the range defined for Flexible Algorithms in the IANA "IGP Parameters" registries under the "IGP Algorithm Types" registry [[I-D.ietf-lsr-flex-algo](#)].
- o Reserved : 3 octet value that SHOULD be set to 0 by the originator and MUST be ignored by the receiver.
- o Metric : 4 octets field to carry the metric information.

The FAPM TLV can be added to the BGP-LS Attribute of the Prefix NLRI originated by a node, only if the corresponding node originates the Prefix in along with the underlying IGP TLV/sub-TLV as described below. This information is derived from the protocol specific advertisements as below.

- o IS-IS, as defined by the ISIS Flexible Algorithm Prefix Metric sub-TLV in [[I-D.ietf-lsr-flex-algo](#)].

- o OSPFv2/OSPFv3, as defined by the OSPF Flexible Algorithm Prefix Metric sub-TLV in [[I-D.ietf-lsr-flex-algo](#)].

The BGP-LS Attribute associated with a Prefix NLRI MAY include one or more FAPM TLVs corresponding to the Flexible Algorithm Prefix Metric for each algorithm associated with that the particular prefix.

5. IANA Considerations

This document requests assigning code-points from the registry "BGP-LS Node Descriptor, Link Descriptor, Prefix Descriptor, and Attribute TLVs" based on table below which reflects the values assigned via the early allocation process. The column "IS-IS TLV/Sub-TLV" defined in the registry does not require any value and should be left empty.

Code Point	Description	Length
1039	Flex Algorithm Definition TLV	variable
1040	Flex Algo Exclude Any Affinity sub-TLV	variable
1041	Flex Algo Include Any Affinity sub-TLV	variable
1042	Flex Algo Include All Affinity sub-TLV	variable
1043	Flex Algo Definition Flags sub-TLV	variable
1044	Flex Algorithm Prefix Metric TLV	variable

6. Manageability Considerations

This section is structured as recommended in [[RFC5706](#)].

The new protocol extensions introduced in this document augment the existing IGP topology information that was distributed via [[RFC7752](#)]. Procedures and protocol extensions defined in this document do not affect the BGP protocol operations and management other than as discussed in the Manageability Considerations section of [[RFC7752](#)]. Specifically, the malformed NLRIs attribute tests in the Fault Management section of [[RFC7752](#)] now encompass the new TLVs for the BGP-LS NLRI in this document.

6.1. Operational Considerations

No additional operation considerations are defined in this document.

6.2. Management Considerations

No additional management considerations are defined in this document.

7. Security Considerations

The new protocol extensions introduced in this document augment the existing IGP topology information that was distributed via [RFC7752]. Procedures and protocol extensions defined in this document do not affect the BGP security model other than as discussed in the Security Considerations section of [RFC7752].

8. Acknowledgements

The authors would like to thank Les Ginsberg for his reviews and contributions to this work.

9. References

9.1. Normative References

- [I-D.ietf-lsr-flex-algo]
Psenak, P., Hegde, S., Filsfils, C., Talaulikar, K., and A. Gulko, "IGP Flexible Algorithm", [draft-ietf-lsr-flex-algo-05](#) (work in progress), November 2019.
- [I-D.ietf-spring-srv6-network-programming]
Filsfils, C., Camarillo, P., Leddy, J., Voyer, D., Matsushima, S., and Z. Li, "SRv6 Network Programming", [draft-ietf-spring-srv6-network-programming-07](#) (work in progress), December 2019.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC7752] Gredler, H., Ed., Medved, J., Previdi, S., Farrel, A., and S. Ray, "North-Bound Distribution of Link-State and Traffic Engineering (TE) Information Using BGP", [RFC 7752](#), DOI 10.17487/RFC7752, March 2016, <<https://www.rfc-editor.org/info/rfc7752>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in [RFC 2119](#) Key Words", [BCP 14](#), [RFC 8174](#), DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.

- [RFC8660] Bashandy, A., Ed., Filsfils, C., Ed., Previdi, S., Decraene, B., Litkowski, S., and R. Shakir, "Segment Routing with the MPLS Data Plane", [RFC 8660](#), DOI 10.17487/RFC8660, December 2019, <<https://www.rfc-editor.org/info/rfc8660>>.

9.2. Informative References

- [I-D.ietf-idr-bgp-ls-segment-routing-ext] Previdi, S., Talaulikar, K., Filsfils, C., Gredler, H., and M. Chen, "BGP Link-State extensions for Segment Routing", [draft-ietf-idr-bgp-ls-segment-routing-ext-16](#) (work in progress), June 2019.
- [I-D.ietf-idr-bgpls-srv6-ext] Dawra, G., Filsfils, C., Talaulikar, K., Chen, M., daniel.bernier@bell.ca, d., and B. Decraene, "BGP Link State Extensions for SRv6", [draft-ietf-idr-bgpls-srv6-ext-01](#) (work in progress), July 2019.
- [I-D.ietf-spring-segment-routing-policy] Filsfils, C., Sivabalan, S., Voyer, D., Bogdanov, A., and P. Mattes, "Segment Routing Policy Architecture", [draft-ietf-spring-segment-routing-policy-06](#) (work in progress), December 2019.
- [RFC2702] Awduche, D., Malcolm, J., Agogbua, J., O'Dell, M., and J. McManus, "Requirements for Traffic Engineering Over MPLS", [RFC 2702](#), DOI 10.17487/RFC2702, September 1999, <<https://www.rfc-editor.org/info/rfc2702>>.
- [RFC3209] Awduche, D., Berger, L., Gan, D., Li, T., Srinivasan, V., and G. Swallow, "RSVP-TE: Extensions to RSVP for LSP Tunnels", [RFC 3209](#), DOI 10.17487/RFC3209, December 2001, <<https://www.rfc-editor.org/info/rfc3209>>.
- [RFC5706] Harrington, D., "Guidelines for Considering Operations and Management of New Protocols and Protocol Extensions", [RFC 5706](#), DOI 10.17487/RFC5706, November 2009, <<https://www.rfc-editor.org/info/rfc5706>>.
- [RFC7308] Osborne, E., "Extended Administrative Groups in MPLS Traffic Engineering (MPLS-TE)", [RFC 7308](#), DOI 10.17487/RFC7308, July 2014, <<https://www.rfc-editor.org/info/rfc7308>>.

[RFC8402] Filsfils, C., Ed., Previdi, S., Ed., Ginsberg, L., Decraene, B., Litkowski, S., and R. Shakir, "Segment Routing Architecture", [RFC 8402](#), DOI 10.17487/RFC8402, July 2018, <<https://www.rfc-editor.org/info/rfc8402>>.

Authors' Addresses

Ketan Talaulikar
Cisco Systems
India

Email: ketant@cisco.com

Peter Psenak
Cisco Systems
Slovakia

Email: ppsenak@cisco.com

Shawn Zandi
LinkedIn
USA

Email: szandi@linkedin.com

Gaurav Dawra
LinkedIn
USA

Email: gdawra.ietf@gmail.com

