Stream: Internet Engineering Task Force (IETF)
RFC: 9350
Category: Standards Track
Published: February 2023
ISSN: 2070-1721
Authors:
P. Psenak, Ed. — *Cisco Systems, Inc.*
S. Hegde — *Juniper Networks, Inc.*
C. Filsfils — *Cisco Systems, Inc.*
K. Talaulikar — *Cisco Systems, Inc*
A. Gulko — *Edward Jones*

RFC 9350
IGP Flexible Algorithm

Abstract

IGP protocols historically compute the best paths over the network based on the IGP metric assigned to the links. Many network deployments use RSVP-TE or Segment Routing - Traffic Engineering (SR-TE) to steer traffic over a path that is computed using different metrics or constraints than the shortest IGP path. This document specifies a solution that allows IGPs themselves to compute constraint-based paths over the network. This document also specifies a way of using Segment Routing (SR) Prefix-SIDs and SRv6 locators to steer packets along the constraint-based paths.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

An IGP-computed path based on the shortest IGP metric is often replaced by a traffic-engineered path due to requirements that are not reflected by the IGP metric. Some networks engineer the IGP metric assignments in a way that the IGP metric reflects the link bandwidth or delay. If, for example, the IGP metric reflects the bandwidth on the link and user traffic is delay sensitive, the best IGP path may not reflect the best path from such a user's perspective.

To overcome this limitation, various sorts of Traffic Engineering have been deployed, including RSVP-TE and SR-TE, in which case the TE component is responsible for computing paths based on additional metrics and/or constraints. Such paths need to be installed in the forwarding tables in addition to, or as a replacement for, the original paths computed by IGPs. Tunnels are often used to represent the engineered paths and mechanisms, like the one described in [RFC3906], and are used to replace the original IGP paths with such tunnel paths.

This document specifies a set of extensions to IS-IS, OSPFv2, and OSPFv3 that enable a router to advertise TLVs that (a) identify a calculation-type, (b) specify a metric-type, and (c) describe a set of constraints on the topology that are to be used to compute the best paths along the constrained topology. A given combination of calculation-type, metric-type, and constraints is known as a "Flexible Algorithm Definition". A router that sends such a set of TLVs also assigns a Flex-Algorithm value to the specified combination of calculation-type, metric-type, and constraints.

This document also specifies a way for a router to use IGPs to associate one or more Segment Routing with the MPLS Data Plane (SR-MPLS) Prefix-SIDs [RFC8660] or Segment Routing over IPv6 (SRv6) locators [RFC8986] with a particular Flex-Algorithm. Each such Prefix-SID or SRv6 locator then represents a path that is computed according to the identified Flex-Algorithm. In SRv6, it is the locator, not the Segment Identifier (SID), that holds the binding to the algorithm.

2. 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.

3. Terminology

This section defines terms that are often used in this document.

Flexible Algorithm Definition (FAD): the set consisting of (a) a calculation-type, (b) a metric-type, and (c) a set of constraints.

Flex-Algorithm: a numeric identifier in the range 128-255 that is associated via configuration with the Flexible Algorithm Definition.

Flexible Algorithm Participation: per the data plane configuration state that expresses whether the node is participating in a particular Flexible Algorithm. Not all routers in a given network need to participate in a given Flexible Algorithm. The Flexible Algorithm(s) that a given router participates in is determined by configuration.

IGP Algorithm: value from the IANA "IGP Algorithm Types" registry defined under the "Interior Gateway Protocol (IGP) Parameters" registry group. IGP Algorithms represent the triplet (calculation-type, metric-type, and constraints), where the second and third elements of the triplet **MAY** be unspecified.

ABR: Area Border Router. In IS-IS terminology, it is also known as the Level 1 (L1) / Level 2 (L2) router.

ASBR: Autonomous System Border Router.

4. Flexible Algorithm

Many possible constraints may be used to compute a path over a network. Some networks are deployed as multiple planes. A simple form of constraint may be to use a particular plane. A more sophisticated form of constraint can include some extended metric, as described in [RFC8570]. Constraints that restrict paths to links with specific affinities or avoid links with specific affinities are also possible. Combinations of these are also possible.

To provide maximum flexibility, a mechanism is provided that allows a router to identify a particular calculation-type and metric-type, describe a particular set of constraints, and assign a numeric identifier, referred to as Flex-Algorithm, to the combination of that calculation-type, metric-type, and those constraints. The mapping between the Flex-Algorithm and its meaning is flexible and defined by the user. As long as all routers in the domain have a common understanding as to what a particular Flex-Algorithm represents, the resulting routing computation is consistent and traffic is not subject to any looping.

The set consisting of (a) a calculation-type, (b) a metric-type, and (c) a set of constraints is referred to as a Flexible Algorithm Definition.

The Flex-Algorithm is a numeric identifier in the range 128-255 that is associated via configuration with the Flexible Algorithm Definition.

The IANA "IGP Algorithm Types" registry defines the set of values for IGP Algorithms. The following values are allocated by IANA from this registry for Flex-Algorithms:

128-255 - Flex-Algorithms

5. Flexible Algorithm Definition Advertisement

To guarantee loop-free forwarding for paths computed for a particular Flex-Algorithm, all routers that (a) are configured to participate in a particular Flex-Algorithm and (b) are in the same Flex-Algorithm Definition advertisement scope **MUST** agree on the definition of the Flex-Algorithm. The following procedures ensure this condition is fulfilled.

5.1. IS-IS Flexible Algorithm Definition Sub-TLV

The IS-IS Flexible Algorithm Definition (FAD) sub-TLV is used to advertise the definition of the Flex-Algorithm.

The IS-IS FAD sub-TLV is advertised as a sub-TLV of the IS-IS Router CAPABILITY TLV-242, which is defined in [RFC7981].

The IS-IS FAD sub-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     |Flex-Algorithm |  Metric-Type  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   Calc-Type   |    Priority   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          Sub-TLVs                             |
+                                                               +
|                            ...                                |

|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

where:

Type: 26

Length: variable number of octets, dependent on the included sub-TLVs.

Flex-Algorithm: Flexible Algorithm number. Single octet value between 128 and 255 inclusive.

Metric-Type: type of metric from the IANA "IGP Metric-Type" registry (Section 18.1.2) to be used during the calculation. The following values are defined:

0: IGP Metric

1: Min Unidirectional Link Delay, as defined in Section 4.2 of [RFC8570], encoded as an application-specific link attribute, as specified in [RFC8919] and Section 12 of this document.

2: Traffic Engineering Default Metric, as defined in Section 3.7 of [RFC5305], encoded as an application-specific link attribute, as specified in [RFC8919] and Section 12 of this document.

Calc-Type: calculation-type. Value from 0-127 inclusive from the IANA "IGP Algorithm Types" registry defined under the "Interior Gateway Protocol (IGP) Parameters" registry. IGP Algorithms in the range of 0-127 have a defined triplet (calculation-type, metric-type, constraints). When used to specify the calculation-type in the FAD sub-TLV, only the calculation-type defined for the specified IGP Algorithm is used. The Metric/Constraints **MUST NOT** be inherited. If the required calculation-type is Shortest Path First, the value 0 **MUST** appear in this field.

Priority: value between 0 and 255 inclusive that specifies the priority of the advertisement. Numerically greater values are preferred. Usage of the priority is described in Section 5.3.

Sub-TLVs: optional sub-TLVs.

The IS-IS FAD sub-TLV **MAY** be advertised in a Label Switched Path (LSP) of any number. The IS-IS router **MAY** advertise more than one IS-IS FAD sub-TLV for a given Flexible Algorithm (see Section 6).

The IS-IS FAD sub-TLV has an area/level scope. The Router Capability TLV in which the FAD sub-TLV is present **MUST** have the S bit clear.

An IS-IS L1/L2 router **MAY** be configured to regenerate the winning FAD from level 2, without any modification to it, to the level 1 area. The regeneration of the FAD sub-TLV from level 2 to level 1 is determined by the L1/L2 router, not by the originator of the FAD advertisement in level 2. In such a case, the regenerated FAD sub-TLV will be advertised in the level 1 Router Capability TLV originated by the L1/L2 router.

An L1/L2 router **MUST NOT** regenerate any FAD sub-TLV from level 1 to level 2.

5.2. OSPF Flexible Algorithm Definition TLV

The OSPF FAD TLV is advertised as a top-level TLV of the Router Information (RI) Link State Advertisement (LSA), which is defined in [RFC7770].

The OSPF FAD 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            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|Flex-Algorithm |   Metric-Type |   Calc-Type   |    Priority   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                            Sub-TLVs                           |
+                                                               +
|                               ...                             |

|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

where:

Type: 16

Length: variable number of octets, dependent on the included sub-TLVs.

Flex-Algorithm: Flexible Algorithm number. Single octet value between 128 and 255 inclusive.

Metric-Type: type of metric from the IANA "IGP Metric-Type" registry (Section 18.1.2) to be used during the calculation. The following values are defined:

0: IGP Metric

1: Min Unidirectional Link Delay, as defined in Section 4.2 of [RFC7471], encoded as an application-specific link attribute, as specified in [RFC8920] and Section 12 of this document.

2: Traffic Engineering Metric, as defined in Section 2.5.5 of [RFC3630], encoded as an application-specific link attribute, as specified in [RFC8920] and Section 12 of this document.

Calc-Type: as described in Section 5.1.

Priority: as described in Section 5.1.

Sub-TLVs: optional sub-TLVs.

When multiple OSPF FAD TLVs, for the same Flexible Algorithm, are received from a given router, the receiver **MUST** use the first occurrence of the TLV in the RI LSA. If the OSPF FAD TLV, for the same Flex-Algorithm, appears in multiple RI LSAs that have different flooding scopes, the OSPF FAD TLV in the RI LSA with the area-scoped flooding scope **MUST** be used. If the OSPF FAD TLV, for the same algorithm, appears in multiple RI LSAs that have the same flooding scope, the OSPF FAD TLV in the RI LSA with the numerically smallest Instance ID **MUST** be used and subsequent instances of the OSPF FAD TLV **MUST** be ignored.

The RI LSA can be advertised at any of the defined opaque flooding scopes (link, area, or Autonomous System (AS)). For the purpose of OSPF FAD TLV advertisement, area-scoped flooding is **REQUIRED**. The AS flooding scope **SHOULD NOT** be used unless local configuration policy on the originating router indicates domain-wide flooding.

5.3. Common Handling of the Flexible Algorithm Definition TLV

This section describes the protocol-independent handling of the FAD TLV (OSPF) or FAD sub-TLV (IS-IS). We will refer to it as FAD TLV in this section, even though, in the case of IS-IS, it is a sub-TLV.

The value of the Flex-Algorithm **MUST** be between 128 and 255 inclusive. If it is not, the FAD TLV **MUST** be ignored.

Only a subset of the routers participating in the particular Flex-Algorithm need to advertise the definition of the Flex-Algorithm.

Every router that is configured to participate in a particular Flex-Algorithm **MUST** select the Flex-Algorithm Definition based on the following ordered rules. This allows for the consistent Flex-Algorithm Definition selection in cases where different routers advertise different definitions for a given Flex-Algorithm:

1. From the advertisements of the FAD in the area (including both locally generated advertisements and received advertisements), select the one(s) with the numerically greatest priority value.
2. If there are multiple advertisements of the FAD with the same numerically greatest priority, select the one that is originated from the router with the numerically greatest System-ID, in the case of IS-IS, or Router ID, in the case of OSPFv2 and OSPFv3. For IS-IS, the System-ID is described in [ISO10589]. For OSPFv2 and OSPFv3, the standard Router ID is described in [RFC2328] and [RFC5340], respectively.

The FAD selected according to these rules is also known as the "winning FAD".

A router that is not configured to participate in a particular Flex-Algorithm **MUST** ignore FAD sub-TLV advertisements for such Flex-Algorithm.

A router that is not participating in a particular Flex-Algorithm **MAY** advertise the FAD for such Flex-Algorithm. Receiving routers **MUST** consider a received FAD advertisement regardless of the Flex-Algorithm participation of that FAD advertisement's originator.

Any change in the Flex-Algorithm Definition may result in a temporary disruption of traffic that is forwarded based on such Flex-Algorithm paths. The impact is similar to any other event that requires network-wide convergence.

If a node is configured to participate in a particular Flexible Algorithm, but there is no valid Flex-Algorithm Definition available for it or the selected Flex-Algorithm Definition includes calculation-type, metric-type, constraint, flag, or sub-TLV that is not supported by the node, it **MUST** stop participating in such Flexible Algorithm. That implies that it **MUST NOT** announce participation for such Flexible Algorithm, as specified in Section 11, and it **MUST** remove any forwarding state associated with it.

The Flex-Algorithm Definition is topology independent. It applies to all topologies that a router participates in.

6. Sub-TLVs of IS-IS FAD Sub-TLV

One of the limitations of IS-IS [ISO10589] is that the length of a TLV/sub-TLV is limited to a maximum of 255 octets. For the FAD sub-TLV, there are a number of sub-sub-TLVs (defined below) that are supported. For a given Flex-Algorithm, it is possible that the total number of octets required to completely define a FAD exceeds the maximum length supported by a single FAD sub-TLV. In such cases, the FAD **MAY** be split into multiple such sub-TLVs, and the content of the multiple FAD sub-TLVs are combined to provide a complete FAD for the Flex-Algorithm. In such a case, the fixed portion of the FAD (see Section 5.1) **MUST** be identical in all FAD sub-TLVs for a given Flex-Algorithm from a given IS. In case the fixed portion of such FAD sub-TLVs differ, the values in the fixed portion in the FAD sub-TLV in the first occurrence in the lowest-numbered LSP from a given IS **MUST** be used.

Any specification that introduces a new IS-IS FAD sub-sub-TLV **MUST** specify whether the FAD sub-TLV may appear multiple times in the set of FAD sub-TLVs for a given Flex-Algorithm from a given IS and how to handle them if multiple are allowed.

6.1. IS-IS Flexible Algorithm Exclude Admin Group Sub-TLV

The Flexible Algorithm Definition can specify "colors" that are used by the operator to exclude links during the Flex-Algorithm path computation.

The IS-IS Flexible Algorithm Exclude Admin Group (FAEAG) sub-TLV is used to advertise the exclude rule that is used during the Flex-Algorithm path calculation, as specified in Section 13.

The IS-IS FAEAG sub-TLV is a sub-TLV of the IS-IS FAD sub-TLV. It 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     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      Extended Admin Group                     |
+-                                                             -+
|                            ...                                |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

where:

Type: 1

Length: variable, dependent on the size of the Extended Admin Group. **MUST** be a multiple of 4 octets.

Extended Administrative Group: Extended Administrative Group, as defined in [RFC7308].

The IS-IS FAEAG sub-TLV **MUST NOT** appear more than once in a single IS-IS FAD sub-TLV. If it appears more than once, the IS-IS FAD sub-TLV **MUST** be ignored by the receiver.

The IS-IS FAEAG sub-TLV **MUST NOT** appear more than once in the set of FAD sub-TLVs for a given Flex-Algorithm from a given IS. If it appears more than once in such a set, the IS-IS FAEAG sub-TLV in the first occurrence in the lowest-numbered LSP from a given IS **MUST** be used, and any other occurrences **MUST** be ignored.

6.2. IS-IS Flexible Algorithm Include-Any Admin Group Sub-TLV

The Flexible Algorithm Definition can specify "colors" that are used by the operator to include links during the Flex-Algorithm path computation.

The IS-IS Flexible Algorithm Include-Any Admin Group sub-TLV is used to advertise the include-any rule that is used during the Flex-Algorithm path calculation, as specified in Section 13.

The IS-IS Flexible Algorithm Include-Any Admin Group sub-TLV is a sub-TLV of the IS-IS FAD sub-TLV. It 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     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      Extended Admin Group                     |
+-                                                             -+
|                            ...                                |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

where:

Type: 2

Length: variable, dependent on the size of the Extended Admin Group. **MUST** be a multiple of 4 octets.

Extended Administrative Group: Extended Administrative Group, as defined in [RFC7308].

The IS-IS Flexible Algorithm Include-Any Admin Group sub-TLV **MUST NOT** appear more than once in a single IS-IS FAD sub-TLV. If it appears more than once, the IS-IS FAD sub-TLV **MUST** be ignored by the receiver.

The IS-IS Flexible Algorithm Include-Any Admin Group sub-TLV **MUST NOT** appear more than once in the set of FAD sub-TLVs for a given Flex-Algorithm from a given IS. If it appears more than once in such a set, the IS-IS Flexible Algorithm Include-Any Admin Group sub-TLV in the first occurrence in the lowest-numbered LSP from a given IS **MUST** be used, and any other occurrences **MUST** be ignored.

6.3. IS-IS Flexible Algorithm Include-All Admin Group Sub-TLV

The Flexible Algorithm Definition can specify "colors" that are used by the operator to include links during the Flex-Algorithm path computation.

The IS-IS Flexible Algorithm Include-All Admin Group sub-TLV is used to advertise the include-all rule that is used during the Flex-Algorithm path calculation, as specified in Section 13.

The IS-IS Flexible Algorithm Include-All Admin Group sub-TLV is a sub-TLV of the IS-IS FAD sub-TLV. It 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     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      Extended Admin Group                     |
+-                                                             -+
|                              ...                              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

where:

Type: 3

Length: variable, dependent on the size of the Extended Admin Group. **MUST** be a multiple of 4 octets.

Extended Administrative Group: Extended Administrative Group, as defined in [RFC7308].

The IS-IS Flexible Algorithm Include-All Admin Group sub-TLV **MUST NOT** appear more than once in a single IS-IS FAD sub-TLV. If it appears more than once, the IS-IS FAD sub-TLV **MUST** be ignored by the receiver.

The IS-IS Flexible Algorithm Include-All Admin Group sub-TLV **MUST NOT** appear more than once in the set of FAD sub-TLVs for a given Flex-Algorithm from a given IS. If it appears more than once in such a set, the IS-IS Flexible Algorithm Include-All Admin Group sub-TLV in the first occurrence in the lowest-numbered LSP from a given IS **MUST** be used, and any other occurrences **MUST** be ignored.

6.4. IS-IS Flexible Algorithm Definition Flags Sub-TLV

The IS-IS Flexible Algorithm Definition Flags (FADF) sub-TLV is a sub-TLV of the IS-IS FAD sub-TLV. It 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     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                             Flags                             |
+-                                                             -+
|                            ...                                |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

where:

Type: 4

Length: variable, number of octets of the Flags field.

Flags:

```
 0 1 2 3 4 5 6 7...
+-+-+-+-+-+-+-+-+...
|M| | |          ...
+-+-+-+-+-+-+-+-+...
```

M-flag: when set, the Flex-Algorithm-specific prefix metric **MUST** be used for inter-area and external prefix calculation. This flag is not applicable to prefixes advertised as SRv6 locators.

A new IANA "IGP Flexible Algorithm Definition Flags" registry is defined for allocation of bits in the Flags field -- see Section 18.2.

Bits are defined/sent starting with bit 0 defined above. Additional bit definitions that may be defined in the future **SHOULD** be assigned in ascending bit order to minimize the number of bits that will need to be transmitted.

Undefined bits **MUST** be transmitted as 0.

Bits that are not transmitted **MUST** be treated as if they are set to 0 on receipt.

The IS-IS FADF sub-TLV **MUST NOT** appear more than once in a single IS-IS FAD sub-TLV. If it appears more than once, the IS-IS FAD sub-TLV **MUST** be ignored by the receiver.

The IS-IS FADF sub-TLV **MUST NOT** appear more than once in the set of FAD sub-TLVs for a given Flex-Algorithm from a given IS. If it appears more than once in such a set, the IS-IS FADF sub-TLV in the first occurrence in the lowest-numbered LSP from a given IS **MUST** be used, and any other occurrences **MUST** be ignored.

If the IS-IS FADF sub-TLV is not present inside the IS-IS FAD sub-TLV, all the bits are assumed to be set to 0.

If a node is configured to participate in a particular Flexible Algorithm, but the selected Flex-Algorithm Definition includes a bit in the IS-IS FADF sub-TLV that is not supported by the node, it **MUST** stop participating in such Flexible Algorithm.

New flag bits may be defined in the future. Implementations **MUST** check all advertised flag bits in the received IS-IS FADF sub-TLV -- not just the subset currently defined.

The M-flag **MUST** not be used when calculating prefix reachability for the SRv6 Locator prefix.

6.5. IS-IS Flexible Algorithm Exclude SRLG Sub-TLV

The Flexible Algorithm Definition can specify Shared Risk Link Groups (SRLGs) that the operator wants to exclude during the Flex-Algorithm path computation.

The IS-IS Flexible Algorithm Exclude SRLG (FAESRLG) sub-TLV is used to advertise the exclude rule that is used during the Flex-Algorithm path calculation, as specified in Section 13.

The IS-IS FAESRLG sub-TLV is a sub-TLV of the IS-IS FAD sub-TLV. It 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     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                  Shared Risk Link Group Value                 |
+-                                                             -+
|                             ...                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

where:

Type: 5

Length: variable, dependent on number of SRLG values. **MUST** be a multiple of 4 octets.

Shared Risk Link Group Value: SRLG value, as defined in [RFC5307].

The IS-IS FAESRLG sub-TLV **MUST NOT** appear more than once in a single IS-IS FAD sub-TLV. If it appears more than once, the IS-IS FAD sub-TLV **MUST** be ignored by the receiver.

The IS-IS FAESRLG sub-TLV **MAY** appear more than once in the set of FAD sub-TLVs for a given Flex-Algorithm from a given IS. This may be necessary in cases where the total number of SRLG values that are specified cause the FAD sub-TLV to exceed the maximum length of a single FAD sub-TLV. In such a case, the receiver **MUST** use the union of all values across all IS-IS FAESRLG sub-TLVs from such set.

7. Sub-TLVs of the OSPF FAD TLV

7.1. OSPF Flexible Algorithm Exclude Admin Group Sub-TLV

The OSPF Flexible Algorithm Exclude Admin Group (FAEAG) sub-TLV is a sub-TLV of the OSPF FAD TLV. Its usage is described in Section 6.1. It 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            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      Extended Admin Group                     |
+-                                                             -+
|                             ...                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

where:

Type: 1

Length: variable, dependent on the size of the Extended Admin Group. **MUST** be a multiple of 4 octets.

Extended Administrative Group: Extended Administrative Group, as defined in [RFC7308].

The OSPF FAEAG sub-TLV **MUST NOT** appear more than once in an OSPF FAD TLV. If it appears more than once, the OSPF FAD TLV **MUST** be ignored by the receiver.

7.2. OSPF Flexible Algorithm Include-Any Admin Group Sub-TLV

The OSPF Flexible Algorithm Include-Any Admin Group sub-TLV is a sub-TLV of the OSPF FAD TLV. The usage of this sub-TLV is described in Section 6.2. It 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            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      Extended Admin Group                     |
+-                                                             -+
|                             ...                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

where:

Type: 2

Length: variable, dependent on the size of the Extended Admin Group. **MUST** be a multiple of 4 octets.

Extended Administrative Group: Extended Administrative Group, as defined in [RFC7308].

The OSPF Flexible Algorithm Include-Any Admin Group sub-TLV **MUST NOT** appear more than once in an OSPF FAD TLV. If it appears more than once, the OSPF FAD TLV **MUST** be ignored by the receiver.

7.3. OSPF Flexible Algorithm Include-All Admin Group Sub-TLV

The OSPF Flexible Algorithm Include-All Admin Group sub-TLV is a sub-TLV of the OSPF FAD TLV. The usage of this sub-TLV is described in Section 6.3. It 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            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      Extended Admin Group                     |
+-                                                             -+
|                            ...                                |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

where:

Type: 3

Length: variable, dependent on the size of the Extended Admin Group. **MUST** be a multiple of 4 octets.

Extended Administrative Group: Extended Administrative Group, as defined in [RFC7308].

The OSPF Flexible Algorithm Include-All Admin Group sub-TLV **MUST NOT** appear more than once in an OSPF FAD TLV. If it appears more than once, the OSPF FAD TLV **MUST** be ignored by the receiver.

7.4. OSPF Flexible Algorithm Definition Flags Sub-TLV

The OSPF Flexible Algorithm Definition Flags (FADF) sub-TLV is a sub-TLV of the OSPF FAD TLV. It 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            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                             Flags                             |
+-                                                             -+
|                            ...                                |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

where:

Type: 4

Length: variable, dependent on the size of the Flags field. **MUST** be a multiple of 4 octets.

Flags:

```
               0 1 2 3 4 5 6 7...
              +-+-+-+-+-+-+-+-+...
              |M| | |          ...
              +-+-+-+-+-+-+-+-+...
```

M-flag: when set, the Flex-Algorithm-specific prefix and ASBR metric **MUST** be used for inter-area and external prefix calculation. This flag is not applicable to prefixes advertised as SRv6 locators.

A new IANA "IGP Flexible Algorithm Definition Flags" registry is defined for allocation of bits in the Flags field -- see Section 18.2.

Bits are defined/sent starting with bit 0 defined above. Additional bit definitions that may be defined in the future **SHOULD** be assigned in ascending bit order to minimize the number of bits that will need to be transmitted.

Undefined bits **MUST** be transmitted as 0.

Bits that are not transmitted **MUST** be treated as if they are set to 0 on receipt.

The OSPF FADF sub-TLV **MUST NOT** appear more than once in an OSPF FAD TLV. If it appears more than once, the OSPF FAD TLV **MUST** be ignored by the receiver.

If the OSPF FADF sub-TLV is not present inside the OSPF FAD TLV, all the bits are assumed to be set to 0.

If a node is configured to participate in a particular Flexible Algorithm, but the selected Flex-Algorithm Definition includes a bit in the OSPF FADF sub-TLV that is not supported by the node, it **MUST** stop participating in such Flexible Algorithm.

New flag bits may be defined in the future. Implementations **MUST** check all advertised flag bits in the received OSPF FADF sub-TLV -- not just the subset currently defined.

The M-flag **MUST** not be used when calculating prefix reachability for the SRv6 Locator prefix.

7.5. OSPF Flexible Algorithm Exclude SRLG Sub-TLV

The OSPF Flexible Algorithm Exclude SRLG (FAESRLG) sub-TLV is a sub-TLV of the OSPF FAD TLV. Its usage is described in Section 6.5. It 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            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                  Shared Risk Link Group Value                 |
+-                                                             -+
|                            ...                                |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

where:

Type: 5

Length: variable, dependent on the number of SRLGs. **MUST** be a multiple of 4 octets.

Shared Risk Link Group Value: SRLG value, as defined in [RFC4203].

The OSPF FAESRLG sub-TLV **MUST NOT** appear more than once in an OSPF FAD TLV. If it appears more than once, the OSPF FAD TLV **MUST** be ignored by the receiver.

8. IS-IS Flexible Algorithm Prefix Metric Sub-TLV

The IS-IS Flexible Algorithm Prefix Metric (FAPM) sub-TLV supports the advertisement of a Flex-Algorithm-specific prefix metric associated with a given prefix advertisement.

The IS-IS FAPM sub-TLV is a sub-TLV of TLVs 135, 235, 236, and 237 and 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     |Flex-Algorithm |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                             Metric                            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

where:

Type: 6

Length: 5 octets

Flex-Algorithm: single octet value between 128 and 255 inclusive.

Metric: 4 octets of metric information.

The IS-IS FAPM sub-TLV **MAY** appear multiple times in its parent TLV. If it appears more than once with the same Flex-Algorithm value, the first instance **MUST** be used and any subsequent instances **MUST** be ignored.

If a prefix is advertised with a Flex-Algorithm prefix metric larger than MAX_PATH_METRIC, as defined in [RFC5305], this prefix **MUST NOT** be considered during the Flexible Algorithm computation.

The usage of the Flex-Algorithm prefix metric is described in Section 13.

The IS-IS FAPM sub-TLV **MUST NOT** be advertised as a sub-TLV of the IS-IS SRv6 Locator TLV [RFC9352]. The IS-IS SRv6 Locator TLV includes the Algorithm and Metric fields, which **MUST** be used instead. If the FAPM sub-TLV is present as a sub-TLV of the IS-IS SRv6 Locator TLV in the received LSP, such FAPM sub-TLV **MUST** be ignored.

9. OSPF Flexible Algorithm Prefix Metric Sub-TLV

The OSPF Flexible Algorithm Prefix Metric (FAPM) sub-TLV supports the advertisement of a Flex-Algorithm-specific prefix metric associated with a given prefix advertisement.

The OSPF FAPM sub-TLV is a sub-TLV of the:

- OSPFv2 Extended Prefix TLV [RFC7684] and
- following OSPFv3 TLVs, as defined in [RFC8362]:
  - Inter-Area Prefix TLV
  - External-Prefix TLV

The OSPF FAPM sub-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            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|Flex-Algorithm |     Flags     |            Reserved           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                            Metric                             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

where:

Type: 3 for OSPFv2, and 26 for OSPFv3

Length: 8 octets

Flex-Algorithm: single octet value between 128 and 255 inclusive.

Flags: 1-octet value

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

E bit: position 0: The type of external metric. If the bit is set, the metric specified is a Type 2 external metric. This bit is applicable only to OSPF external and Not-So-Stubby Area (NSSA) external prefixes. This is semantically the same as the E bit in Appendix A.4.5 of [RFC2328] and Appendix A.4.7 of [RFC5340] for OSPFv2 and OSPFv3, respectively.

Bits 1 through 7: **MUST** be cleared by the originator and ignored by the receiver.

Reserved: **MUST** be set to 0 and ignored at reception.

Metric: 4 octets of metric information.

The OSPF FAPM sub-TLV **MAY** appear multiple times in its parent TLV. If it appears more than once with the same Flex-Algorithm value, the first instance **MUST** be used and any subsequent instances **MUST** be ignored.

The usage of the Flex-Algorithm prefix metric is described in Section 13.

10. OSPF Flexible Algorithm ASBR Reachability Advertisement

An OSPF ABR advertises the reachability of ASBRs in its attached areas to enable routers within those areas to perform route calculations for external prefixes advertised by the ASBRs. OSPF extensions for advertisement of Flex-Algorithm-specific reachability and the metric for ASBRs is similarly required for Flex-Algorithm external prefix computations, as described further in Section 13.1.

10.1. OSPFv2 Extended Inter-Area ASBR LSA

The OSPFv2 Extended Inter-Area ASBR (EIA-ASBR) LSA is an OSPF Opaque LSA [RFC5250] that is used to advertise additional attributes related to the reachability of the OSPFv2 ASBR that is external to the area yet internal to the OSPF domain. Semantically, the OSPFv2 EIA-ASBR LSA is equivalent to the fixed format Type 4 summary-LSA [RFC2328]. Unlike the Type 4 summary-LSA, the Link State ID (LSID) of the EIA-ASBR LSA does not carry the ASBR Router ID -- the ASBR Router ID is carried in the body of the LSA. The OSPFv2 EIA-ASBR LSA is advertised by an OSPFv2 ABR, and its flooding is defined to be area-scoped only.

An OSPFv2 ABR generates the EIA-ASBR LSA for an ASBR when it is advertising the Type 4 summary-LSA for it and has the need for advertising additional attributes for that ASBR beyond what is conveyed in the fixed-format Type 4 summary-LSA. An OSPFv2 ABR **MUST NOT** advertise the EIA-ASBR LSA for an ASBR for which it is not advertising the Type 4 summary-LSA. This ensures that the ABR does not generate the EIA-ASBR LSA for an ASBR to which it does not have reachability in the base OSPFv2 topology calculation. The OSPFv2 ABR **SHOULD NOT** advertise the EIA-ASBR LSA for an ASBR when it does not have additional attributes to advertise for that ASBR.

The OSPFv2 EIA-ASBR LSA 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|            LS age             |     Options   |    LS Type    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Opaque Type  |                  Opaque ID                    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                     Advertising Router                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                     LS sequence number                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         LS checksum           |             Length            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
+-                            TLVs                             -+
|                             ...                               |
```

The LS age and Options fields are as defined in Appendix A.4.1 of [RFC2328].

The LS Type **MUST** be 10, indicating that the Opaque LSA flooding scope is area-local [RFC5250].

The Opaque Type used by the OSPFv2 EIA-ASBR LSA is 11. The Opaque Type is used to differentiate the various types of OSPFv2 Opaque LSAs and is described in Section 3 of [RFC5250].

The Opaque ID field is an arbitrary value used to maintain multiple OSPFv2 EIA-ASBR LSAs. For OSPFv2 EIA-ASBR LSAs, the Opaque ID has no semantic significance other than to differentiate OSPFv2 EIA-ASBR LSAs originated by the same OSPFv2 ABR. If multiple OSPFv2 EIA-ASBR LSAs specify the same ASBR, the attributes from the Opaque LSA with the lowest Opaque ID **SHOULD** be used.

The Advertising Router, LS sequence number, and LS checksum fields are as defined in Appendix A.4.1 of [RFC2328].

The Length field is as defined in Appendix A.4.1 of [RFC2328]. It represents the total length (in octets) of the Opaque LSA, including the LSA header and all TLVs (including padding).

The format of the TLVs within the body of the OSPFv2 EIA-ASBR LSA is the same as the format used by the Traffic Engineering Extensions to OSPFv2 [RFC3630]. The variable TLV section consists of one or more nested TLV tuples. Nested TLVs are also referred to as sub-TLVs. The TLV Length field defines the length of the value portion in octets (thus, a TLV with no value portion would have a length of 0). The TLV is padded to 4-octet alignment; padding is not included in the Length field (so a 3-octet value would have a length of 3, but the total size of the TLV would be 8 octets). Nested TLVs are also 32-bit aligned. For example, a 1-octet value would have the Length field set to 1, and 3 octets of padding would be added to the end of the value portion of the TLV. The padding is composed of zeros.

10.1.1. OSPFv2 Extended Inter-Area ASBR TLV

The OSPFv2 Extended Inter-Area ASBR (EIA-ASBR) TLV is a top-level TLV of the OSPFv2 EIA-ASBR LSA and is used to advertise additional attributes associated with the reachability of an ASBR.

The OSPFv2 EIA-ASBR 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            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          ASBR Router ID                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
.                                                               .
.                              Sub-TLVs                         .
.                                                               .
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

where:

Type: 1

Length: variable number of octets.

ASBR Router ID: 4 octets carrying the OSPF Router ID of the ASBR whose information is being carried.

Sub-TLVs: variable

Only a single OSPFv2 EIA-ASBR TLV **MUST** be advertised in each OSPFv2 EIA-ASBR LSA, and the receiver **MUST** ignore all instances of this TLV other than the first one in an LSA.

The OSPFv2 EIA-ASBR TLV **MUST** be present inside an OSPFv2 EIA-ASBR LSA and **MUST** include at least a single sub-TLV; otherwise, the OSPFv2 EIA-ASBR LSA **MUST** be ignored by the receiver.

10.2. OSPF Flexible Algorithm ASBR Metric Sub-TLV

The OSPF Flexible Algorithm ASBR Metric (FAAM) sub-TLV supports the advertisement of a Flex-Algorithm-specific metric associated with a given ASBR reachability advertisement by an ABR.

The OSPF FAAM sub-TLV is a sub-TLV of the:

- OSPFv2 Extended Inter-Area ASBR TLV, as defined in Section 10.1.1, and
- OSPFv3 Inter-Area-Router TLV, as defined in [RFC8362].

The OSPF FAAM sub-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            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|Flex-Algorithm |                   Reserved                    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                            Metric                             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

where:

Type: 1 for OSPFv2, and 33 for OSPFv3

Length: 8 octets

Flex-Algorithm: single octet value between 128 and 255 inclusive.

Reserved: 3 octets. **MUST** be set to 0 and ignored at reception.

Metric: 4 octets of metric information.

The OSPF FAAM sub-TLV **MAY** appear multiple times in its parent TLV. If it appears more than once with the same Flex-Algorithm value, the first instance **MUST** be used and any subsequent instances **MUST** be ignored.

The advertisement of the ASBR reachability using the OSPF FAAM sub-TLV inside the OSPFv2 EIA-ASBR LSA follows Section 12.4.3 of [RFC2328] and inside the OSPFv3 E-Inter-Area-Router-LSA follows Section 4.8.5 of [RFC5340]. The reachability of the ASBR is evaluated in the context of the specific Flex-Algorithm.

The FAAM computed by the ABR will be equal to the metric to reach the ASBR for a given Flex-Algorithm in a source area or the cumulative metric via an ABR(s) when the ASBR is in a remote area. This is similar in nature to how the metric is set when the ASBR reachability metric is computed in the default algorithm for the metric in the OSPFv2 Type 4 ASBR summary-LSA and the OSPFv3 Inter-Area-Router-LSA.

An OSPF ABR **MUST NOT** include the OSPF FAAM sub-TLV with a specific Flex-Algorithm in its reachability advertisement for an ASBR between areas unless that ASBR is reachable for it in the context of that specific Flex-Algorithm.

An OSPF ABR **MUST** include the OSPF FAAM sub-TLVs as part of the ASBR reachability advertisement between areas for any Flex-Algorithm for which the winning FAD includes the M-flag and the ASBR is reachable in the context of that specific Flex-Algorithm.

OSPF routers **MUST** use the OSPF FAAM sub-TLV to calculate the reachability of the ASBRs if the winning FAD for the specific Flex- Algorithm includes the M-flag. OSPF routers **MUST NOT** use the OSPF FAAM sub-TLV to calculate the reachability of the ASBRs for the specific Flex-Algorithm if the winning FAD for such Flex-Algorithm does not include the M-flag. Instead, the OSPFv2 Type 4 summary-LSAs or the OSPFv3 Inter-Area-Router-LSAs **MUST** be used, as specified in Section 16.2 of [RFC2328] and Section 4.8.5 of [RFC5340] for OSPFv2 and OSPFv3, respectively.

The processing of a new or changed OSPF FAAM sub-TLV triggers the processing of external routes similar to what is described in Section 16.5 of [RFC2328] for OSPFv2 and Section 4.8.5 of [RFC5340] for OSPFv3 for the specific Flex-Algorithm. The OSPF external and NSSA external route calculation should be limited to a Flex-Algorithm(s) for which the winning FAD(s) includes the M-flag.

Processing of the OSPF FAAM sub-TLV does not require the existence of the equivalent OSPFv2 Type 4 summary-LSA or the OSPFv3 Inter-Area-Router-LSA that is advertised by the same ABR inside the area. The presence of the base LSA is not mandatory for the usage of the extended LSA with the OSPF FAAM sub-TLV.

11. Advertisement of Node Participation in a Flex-Algorithm

When a router is configured to participate in a particular Flex-Algorithm and is advertising such participation, it is participating in that Flex-Algorithm.

Paths for various data planes **MAY** be computed for a specific Flex-Algorithm. Each data plane uses its own specific forwarding over such Flex-Algorithm paths. To guarantee the presence of the data-plane-specific forwarding, associated with a particular Flex-Algorithm, a router **MUST** advertise its participation for a particular Flex-Algorithm for each data plane. Some data planes may share a common participation advertisement (e.g., SR-MPLS and SRv6).

Advertisement of the participation for any particular Flex-Algorithm in any data plane is subject to the condition specified in Section 5.3.

11.1. Advertisement of Node Participation for Segment Routing

[RFC8665], [RFC8666], and [RFC8667] (IGP Segment Routing extensions) describe how the SR-Algorithm is used to compute the IGP best path.

Routers advertise support for the SR-Algorithm as a node capability, as described in the above-mentioned IGP Segment Routing extensions. To advertise participation for a particular Flex-Algorithm for Segment Routing, including both SR-MPLS and SRv6, the Flex-Algorithm value **MUST** be advertised in the SR-Algorithm TLV (OSPF) or sub-TLV (IS-IS).

Segment Routing Flex-Algorithm participation advertisement is topology independent. When a router advertises participation in an SR-Algorithm, the participation applies to all topologies in which the advertising node participates.

11.2. Advertisement of Node Participation for Other Data Planes

This section describes considerations related to how other data planes can advertise their participation in a specific Flex-Algorithm.

Data-plane-specific Flex-Algorithm participation advertisements **MAY** be topology specific or **MAY** be topology independent, depending on the data plane itself.

Data-plane-specific advertisement for Flex-Algorithm participation **MUST** be defined for each data plane and is outside the scope of this document.

12. Advertisement of Link Attributes for Flex-Algorithm

Various link attributes may be used during the Flex-Algorithm path calculation. For example, include or exclude rules based on link affinities can be part of the Flex-Algorithm Definition, as defined in Sections 6 and 7.

Application-specific link attributes, as specified in [RFC8919] or [RFC8920], that are to be used during Flex-Algorithm calculation **MUST** use the Application-Specific Link Attribute (ASLA) advertisements defined in [RFC8919] or [RFC8920] unless, in the case of IS-IS, the L-flag is set in the ASLA advertisement. When the L-flag is set, then legacy advertisements **MUST** be used, subject to the procedures and constraints defined in Section 4.2 of [RFC8919] and Section 6.

The mandatory use of ASLA advertisements applies to link attributes specifically mentioned in this document (Min Unidirectional Link Delay, TE Default Metric, Administrative Group, Extended Administrative Group, and Shared Risk Link Group) and any other link attributes that may be used in support of Flex-Algorithm in the future.

A new Application Identifier Bit is defined to indicate that the ASLA advertisement is associated with the Flex-Algorithm application. This bit is set in the Standard Application Bit Mask (SABM) defined in [RFC8919] or [RFC8920]:

Bit 3: Flexible Algorithm (X-bit)

ASLA Admin Group Advertisements to be used by the Flexible Algorithm application **MAY** use either the Administrative Group or Extended Administrative Group encodings.

A receiver supporting this specification **MUST** accept both ASLA Administrative Group and Extended Administrative Group TLVs, as defined in [RFC8919] or [RFC8920]. In the case of IS-IS, if the L-flag is set in the ASLA advertisement, as defined in Section 4.2 of [RFC8919], then the receiver **MUST** be able to accept both the Administrative Group TLV, as defined in [RFC5305], and the Extended Administrative Group TLV, as defined in [RFC7308].

13. Calculation of Flexible Algorithm Paths

A router **MUST** be configured to participate in a given Flex-Algorithm K and **MUST** select the FAD based on the rules defined in Section 5.3 before it can compute any path for that Flex-Algorithm.

No specific two-way connectivity check is performed during the Flex-Algorithm path computation. The result of the existing Flex-Algorithm-agnostic, two-way connectivity check is used during the Flex-Algorithm path computation.

As described in Section 11, participation for any particular Flex-Algorithm **MUST** be advertised on a per data plane basis. Calculation of the paths for any particular Flex-Algorithm is data plane specific.

Multiple data planes **MAY** use the same Flex-Algorithm value at the same time and, as such, share the FAD for it. Traffic for each data plane will be forwarded based on the data-plane-specific forwarding entries.

The Flex-Algorithm Definition is data plane independent and is used by all Flex-Algorithm data planes.

The way various data planes handle nodes that do not participate in Flexible Algorithm is data plane specific. If the data plane only wants to consider participating nodes during the Flex-Algorithm calculation, then when computing paths for a given Flex-Algorithm, all nodes that do not advertise participation for that Flex-Algorithm in their data-plane-specific advertisements **MUST** be pruned from the topology. Segment Routing, including both SR-MPLS and SRv6, are data planes that **MUST** use such pruning when computing Flex-Algorithm paths.

When computing the path for a given Flex-Algorithm, the metric-type that is part of the Flex-Algorithm Definition (Section 5) **MUST** be used.

When computing the path for a given Flex-Algorithm, the calculation-type that is part of the Flex-Algorithm Definition (Section 5) **MUST** be used.

Various links that include or exclude rules can be part of the Flex-Algorithm Definition. To refer to a particular bit within an Admin Group or Extended Admin Group, we use the term "color".

Rules, in the order as specified below, **MUST** be used to prune links from the topology during the Flex-Algorithm computation.

For all links in the topology:

1. Check if any exclude Administrative Group rule is part of the Flex-Algorithm Definition. If such exclude rule exists, check if any color that is part of the exclude rule is also set on the link. If such a color is set, the link **MUST** be pruned from the computation.
2. Check if any exclude SRLG rule is part of the Flex-Algorithm Definition. If such exclude rule exists, check if the link is part of any SRLG that is also part of the SRLG exclude rule. If the link is part of such SRLG, the link **MUST** be pruned from the computation.
3. Check if any include-any Administrative Group rule is part of the Flex-Algorithm Definition. If such include-any rule exists, check if any color that is part of the include-any rule is also set on the link. If no such color is set, the link **MUST** be pruned from the computation.
4. Check if any include-all Administrative Group rule is part of the Flex-Algorithm Definition. If such include-all rule exists, check if all colors that are part of the include-all rule are also set on the link. If all such colors are not set on the link, the link **MUST** be pruned from the computation.
5. If the Flex-Algorithm Definition uses something other than the IGP metric (Section 5), and such metric is not advertised for the particular link in a topology for which the computation

is done, such link **MUST** be pruned from the computation. A metric of value 0 **MUST NOT** be assumed in such a case.

13.1. Multi-area and Multi-domain Considerations

Any IGP Shortest Path Tree calculation is limited to a single area. This applies to Flex-Algorithm calculations as well. Given that the computing router does not have visibility of the topology of the next areas or domain, the Flex-Algorithm-specific path to an inter-area or inter-domain prefix will be computed for the local area only. The egress L1/L2 router (ABR in OSPF), or ASBR for an inter-domain case, will be selected based on the best path for the given Flex-Algorithm in the local area, and such egress ABR or ASBR router will be responsible to compute the best Flex-Algorithm-specific path over the next area or domain. This may produce an end-to-end path, which is suboptimal based on Flex-Algorithm constraints. In cases where the ABR or ASBR has no reachability to a prefix for a given Flex-Algorithm in the next area or domain, the traffic could be dropped by the ABR/ASBR.

To allow the optimal end-to-end path for an inter-area or inter-domain prefix for any Flex-Algorithm to be computed, the FAPM has been defined in Sections 8 and 9. For external route calculation for prefixes originated by ASBRs in remote areas in OSPF, the FAAM has been defined in Section 10.2 for the ABR to indicate its ASBR reachability along with the metric for the specific Flex-Algorithm.

If the FAD selected based on the rules defined in Section 5.3 includes the M-flag, an ABR or an ASBR **MUST** include the FAPM (see Sections 8 and 9) when advertising the prefix that is reachable in a given Flex-Algorithm between areas or domains. Such metric will be equal to the metric to reach the prefix for that Flex-Algorithm in its source area or domain. This is similar in nature to how the metric is set when prefixes are advertised between areas or domains for the default algorithm. When a prefix is unreachable in its source area or domain in a specific Flex-Algorithm, then an ABR or ASBR **MUST NOT** include the FAPM for that Flex-Algorithm when advertising the prefix between areas or domains.

If the FAD selected based on the rules defined in Section 5.3 includes the M-flag, the FAPM **MUST** be used during the calculation of prefix reachability for the inter-area and external prefixes. If the FAPM for the Flex-Algorithm is not advertised with the inter-area or external prefix reachability advertisement, the prefix **MUST** be considered as unreachable for that Flex-Algorithm. Similarly, in the case of OSPF, for ASBRs in remote areas, if the FAAM is not advertised by the local ABR(s), the ASBR **MUST** be considered as unreachable for that Flex-Algorithm, and the external prefix advertisements from such an ASBR are not considered for that Flex-Algorithm.

The Flex-Algorithm prefix metrics and the OSPF Flex-Algorithm ASBR metrics **MUST NOT** be used during the Flex-Algorithm computation unless the FAD selected based on the rules defined in Section 5.3 includes the M-flag, as described in Sections 6.4 or 7.4.

In the case of OSPF, when calculating external routes in a Flex-Algorithm, if the winning FAD includes the M-flag, and the advertising ASBR is in a remote area, the metric will be the sum of the following:

- the FAPM for that Flex-Algorithm advertised with the external route by the ASBR
- the metric to reach the ASBR for that Flex-Algorithm from the local ABR, i.e., the FAAM for that Flex-Algorithm advertised by the ABR in the local area for that ASBR
- the Flex-Algorithm-specific metric to reach the local ABR

This is similar in nature to how the metric is calculated for routes learned from remote ASBRs in the default algorithm using the OSPFv2 Type 4 ASBR summary-LSA and the OSPFv3 Inter-Area-Router-LSA.

If the FAD selected based on the rules defined in Section 5.3 does not include the M-flag, then the IGP metrics associated with the prefix reachability advertisements used by the base IS-IS and OSPF protocol **MUST** be used for the Flex-Algorithm route computation. Similarly, in the case of external route calculations in OSPF, the ASBR reachability is determined based on the base OSPFv2 Type 4 summary-LSA and the OSFPv3 Inter-Area-Router-LSA.

It is **NOT RECOMMENDED** to use the Flex-Algorithm for inter-area or inter-domain prefix reachability without the M-flag set. The reason is that, without the explicit Flex-Algorithm prefix metric advertisement (and the Flex-Algorithm ASBR metric advertisement in the case of OSPF external route calculation), it is not possible to conclude whether the ABR or ASBR has reachability to the inter-area or inter-domain prefix for a given Flex-Algorithm in the next area or domain. Sending the Flex-Algorithm traffic for such a prefix towards the ABR or ASBR may result in traffic looping or persistent traffic drop.

During the route computation, it is possible for the Flex-Algorithm-specific metric to exceed the maximum value that can be stored in an unsigned 32-bit variable. In such scenarios, the value **MUST** be considered to be of value 0xFFFFFFFF during the computation and advertised as such.

The FAPM **MUST NOT** be advertised with IS-IS L1 or L2 intra-area, OSPFv2 intra-area, or OSPFv3 intra-area routes. If the FAPM is advertised for these route-types, it **MUST** be ignored during the prefix reachability calculation.

The M-flag in the FAD is not applicable to prefixes advertised as SRv6 locators. The IS-IS SRv6 Locator TLV [RFC9352] includes the Algorithm and Metric fields. When the SRv6 Locator is advertised between areas or domains, the Metric field in the Locator TLV of IS-IS **MUST** be used irrespective of the M-flag in the FAD advertisement.

OSPF external and NSSA external prefix advertisements **MAY** include a non-zero forwarding address in the prefix advertisements in the base protocol. In such a scenario, the Flex-Algorithm-specific reachability of the external prefix is determined by Flex-Algorithm-specific reachability of the forwarding address.

In OSPF, the procedures for translation of NSSA external prefix advertisements into external prefix advertisements performed by an NSSA ABR [RFC3101] remain unchanged for Flex-Algorithm. An NSSA translator **MUST** include the OSPF FAPM sub-TLVs for all Flex-Algorithms that are in the original NSSA external prefix advertisement from the NSSA ASBR in the translated external prefix advertisement generated by it, regardless of its participation in those Flex-Algorithms or its having reachability to the NSSA ASBR in those Flex-Algorithms.

An area could become partitioned from the perspective of the Flex-Algorithm due to the constraints and/or metric being used for it while maintaining the continuity in the base algorithm. When that happens, some destinations inside that area could become unreachable in that Flex-Algorithm. These destinations will not be able to use an inter-area path. This is the consequence of the fact that the inter-area prefix reachability advertisement would not be available for these intra-area destinations within the area. It is **RECOMMENDED** to minimize the risk of such partitioning by providing enough redundancy inside the area for each Flex-Algorithm being used.

14. Flex-Algorithm and Forwarding Plane

This section describes how Flex-Algorithm paths are used in forwarding.

14.1. Segment Routing MPLS Forwarding for Flex-Algorithm

This section describes how Flex-Algorithm paths are used with SR MPLS forwarding.

Prefix-SID advertisements include an SR-Algorithm value and, as such, are associated with the specified SR-Algorithm. Prefix-SIDs are also associated with a specific topology that is inherited from the associated prefix reachability advertisement. When the algorithm value advertised is a Flex-Algorithm value, the Prefix-SID is associated with paths calculated using that Flex-Algorithm in the associated topology.

A Flex-Algorithm path **MUST** be installed in the MPLS forwarding plane using the MPLS label that corresponds to the Prefix-SID that was advertised for that Flex-algorithm. If the Prefix-SID for a given Flex-Algorithm is not known, the Flex-Algorithm-specific path cannot be installed in the MPLS forwarding plane.

Traffic that is supposed to be routed via Flex-Algorithm-specific paths **MUST** be dropped when there are no such paths available.

Loop Free Alternate (LFA) paths ([RFC6571] or its variants) for a given Flex-Algorithm **MUST** be computed using the same constraints as the calculation of the primary paths for that Flex-Algorithm. LFA paths **MUST** only use Prefix-SIDs advertised specifically for the given algorithm. LFA paths **MUST NOT** use an Adjacency SID that belongs to a link that has been pruned from the Flex-Algorithm computation.

If LFA protection is being used to protect a given Flex-Algorithm path, all routers in the area participating in the given Flex-Algorithm **SHOULD** advertise at least one Flex-Algorithm-specific Node-SID. These Node-SIDs are used to steer traffic over the LFA-computed backup path.

14.2. SRv6 Forwarding for Flex-Algorithm

This section describes how Flex-Algorithm paths are used with SRv6 forwarding.

In SRv6, a node is provisioned with a (topology, algorithm) specific locator for each of the topology/algorithm pairs supported by that node. Each locator is an aggregate prefix for all SIDs provisioned on that node that have the matching topology/algorithm.

The SRv6 locator advertisement in IS-IS [RFC9352] includes the Multi-Topology Identifier (MTID) value that associates the locator with a specific topology. SRv6 locator advertisements also include an algorithm value that explicitly associates the locator with a specific algorithm. When the algorithm value advertised with a locator represents a Flex-Algorithm, the paths to the locator prefix **MUST** be calculated using the specified Flex-Algorithm in the associated topology.

Forwarding entries for the locator prefixes advertised in IS-IS **MUST** be installed in the forwarding plane of the receiving SRv6-capable routers when the associated topology/algorithm is participating in them. Forwarding entries for locators associated with Flex-Algorithms in which the node is not participating **MUST NOT** be installed in the forwarding plane.

When the locator is associated with a Flex-Algorithm, LFA paths to the locator prefix **MUST** be calculated using such Flex-Algorithm in the associated topology to guarantee that they follow the same constraints as the calculation of the primary paths. LFA paths **MUST** only use SRv6 SIDs advertised specifically for the given Flex-Algorithm.

If LFA protection is being used to protect locators associated with a given Flex-Algorithm, all routers in the area participating in the given Flex-Algorithm **SHOULD** advertise at least one Flex-Algorithm-specific locator and END SID per node and one END.X SID for every link that has not been pruned from such Flex-Algorithm computation. These locators and SIDs are used to steer traffic over the LFA-computed backup path.

14.3. Other Data Planes' Forwarding for Flex-Algorithm

Any data plane that wants to use Flex-Algorithm-specific forwarding needs to install some form of Flex-Algorithm-specific forwarding entries.

Data-plane-specific forwarding for Flex-Algorithms **MUST** be defined for each data plane and is outside the scope of this document.

15. Operational Considerations

15.1. Inter-area Considerations

The scope of the Flex-Algorithm computation and the scope of the FAD is an area. In IS-IS, the Router Capability TLV in which the FAD sub-TLV is advertised **MUST** have the S bit clear, which prevents it from being flooded outside the level in which it was originated. Even though in OSPF the FAD sub-TLV can be flooded in an RI LSA that has an AS flooding scope, the FAD selection is performed for each individual area in which it is being used.

There is no requirement for the FAD for a particular Flex-Algorithm to be identical in all areas in the network. For example, traffic for the same Flex-Algorithm may be optimized for minimal delay (e.g., using delay metric) in one area or level while being optimized for available bandwidth (e.g., using IGP metric) in another area or level.

As described in Section 5.1, IS-IS allows the regeneration of the winning FAD from level 2, without any modification to it, into a level 1 area. This allows the operator to configure the FAD in one or multiple routers in level 2, without the need to repeat the same task in each level 1 area, if the intent is to have the same FAD for the particular Flex-Algorithm across all levels. This can similarly be achieved in OSPF by using the AS flooding scope of the RI LSA in which the FAD sub-TLV for the particular Flex-Algorithm is advertised.

Regeneration of the FAD from a level 1 area to the level 2 area is not supported in IS-IS, so if the intent is to regenerate the FAD between IS-IS levels, the FAD **MUST** be defined on a router(s) that is in level 2. In OSPF, the FAD definition can be done in any area and propagated to all routers in the OSPF routing domain by using the AS flooding scope of the RI LSA.

15.2. Usage of the SRLG Exclude Rule with Flex-Algorithm

There are two different ways in which SRLG information can be used with Flex-Algorithms:

- In a context of a single Flex-Algorithm, it can be used for computation of backup paths, as described in [RTGWG-SEGMENT-ROUTING-TI-LFA]. This usage does not require association of any specific SRLG constraint with the given Flex-Algorithm Definition.
- In the context of multiple Flex-Algorithms, it can be used for creating disjoint sets of paths by pruning the links belonging to a specific SRLG from the topology on which a specific Flex-Algorithm computes its paths. This usage:
  - facilitates the usage of already deployed SRLG configurations for the setup of disjoint paths between two or more Flex-Algorithms and
  - requires explicit association of a given Flex-Algorithm with a specific set of SRLG constraints, as defined in Sections 6.5 and 7.5.

The two usages mentioned above are orthogonal.

15.3. Max-Metric Consideration

Both IS-IS and OSPF have a mechanism to set the IGP metric on a link to a value that would make the link either unreachable or serve as the link of last resort. Similar functionality would be needed for the Min Unidirectional Link Delay and TE metric, as these can be used to compute Flex-Algorithm paths.

The link can be made unreachable for all Flex-Algorithms that use the Min Unidirectional Link Delay as a metric, as described in Section 5.1, by removing the Flex-Algorithm ASLA Min Unidirectional Link Delay advertisement for the link. The link can be made the link of last resort by setting the delay value in the Flex-Algorithm ASLA delay advertisement for the link to the value of 16,777,215 (2^{24} - 1).

The link can be made unreachable for all Flex-Algorithms that use the TE metric, as described in Section 5.1, by removing the Flex-Algorithm ASLA TE metric advertisement for the link. The link can be made the link of last resort by setting the TE metric value in the Flex-Algorithm ASLA delay advertisement for the link to the value of (2^{24} - 1) in IS-IS and (2^{32} - 1) in OSPF.

15.4. Flexible Algorithm Definition and Changes

When configuring a node to participate in a specific Flex-Algorithm, the components of the FAD (calculation-type, metric-type, and constraints) should be considered carefully. The configuration of participation in a particular Flex-Algorithm doesn't guarantee that the node will actively participate in it, because it may not support the calculation-type, the metric-type, or some constraint advertised by the winning FAD (see Section 5.3). Changes in the FAD configuration should also be considered in light of the capabilities of the participating routers in the scope of the FAD advertisement.

As Section 5.3 notes, a change in the Flex-Algorithm Definition may require network-wide Shortest Path First (SPF) recomputation and network reconvergence. This potential for disruption should be taken into consideration when planning and making changes to the FAD.

15.5. Number of Flex-Algorithms

The maximum number of Flex-Algorithms is determined by the algorithm range 128-255, as specified in Section 4. Although possible, it is not expected that all of them will be used simultaneously. Typically, only a limited subset of Flex-Algorithms is expected to be deployed in the network.

16. Backward Compatibility

This extension brings no new backward-compatibility issues. IS-IS, OSPFv2, and OSPFv3 all have well-defined handling of unrecognized TLVs and sub-TLVs that allows the introduction of new extensions, similar to those defined here, without introducing any interoperability issues.

17. Security Considerations

This document adds two new ways to disrupt IGP networks:

- An attacker can hijack a particular Flex-Algorithm by advertising a FAD with a priority of 255 (or any priority higher than that of the legitimate nodes).
- An attacker could make it look like a router supports a particular Flex-Algorithm when it actually doesn't, or vice versa.

Both of these attacks can be addressed by the existing security extensions, as described in [RFC5304] and [RFC5310] for IS-IS, in [RFC2328] and [RFC7474] for OSPFv2, and in [RFC4552] and [RFC5340] for OSPFv3.

If the node that is authenticated is taken over by an attacker, such rogue node can advertise the FAD for any Flex-Algorithm. Doing so may result in traffic for such Flex-Algorithm to be misrouted, or not delivered at all, for example, by using an unsupported metric-type, calculation-type, or constraint. Such attack is not preventable through authentication, and it is not different from advertising any other incorrect information through IS-IS or OSPF.

18. IANA Considerations

18.1. IGP IANA Considerations

18.1.1. IGP Algorithm Types Registry

This document makes the following registration in the "IGP Algorithm Types" registry:

Value	Description	Reference
128-255	Flexible Algorithms	RFC 9350, Section 4

Table 1: IGP Algorithm Types Registry

18.1.2. IGP Metric-Type Registry

IANA has created the "IGP Metric-Type" registry within the "Interior Gateway Protocol (IGP) Parameters" registry group. The registration policy is "Standards Action" [RFC8126] [RFC7120]. Values are assigned from the range 0-255 and have been registered as follows.

Type	Description	Reference
0	IGP Metric	RFC 9350, Section 5.1
1	Min Unidirectional Link Delay as defined in [RFC8570], Section 4.2 and [RFC7471], Section 4.2	RFC 9350, Section 5.1

Type	Description	Reference
2	Traffic Engineering Default Metric as defined in [RFC5305], Section 3.7 and Traffic Engineering Metric as defined in [RFC3630], Section 2.5.5	RFC 9350, Section 5.1

Table 2: IGP Metric-Type Registry

18.2. IGP Flexible Algorithm Definition Flags Registry

IANA has created the "IGP Flexible Algorithm Definition Flags" registry within the "Interior Gateway Protocol (IGP) Parameters" registry group. The registration policy is "Standards Action". New registrations should be assigned in ascending bit order (see Section 6.4); the following single bit has been assigned as follows.

Bit	Name	Reference
0	Prefix Metric Flag (M-flag)	RFC 9350, Sections 6.4 and 7.4

Table 3: IGP Flexible Algorithm Definition Flags Registry

18.3. IS-IS IANA Considerations

18.3.1. IS-IS Sub-TLVs for IS-IS Router CAPABILITY TLV Registry

This document makes the following registration in the "IS-IS Sub-TLVs for IS-IS Router CAPABILITY TLV" registry.

Value	Description	Reference
26	Flexible Algorithm Definition (FAD)	RFC 9350, Section 5.1

Table 4: IS-IS Sub-TLVs for IS-IS Router CAPABILITY TLV Registry

18.3.2. IS-IS Sub-TLVs for TLVs Advertising Prefix Reachability Registry

This document makes the following registration in the "IS-IS Sub-TLVs for TLVs Advertising Prefix Reachability" registry.

Type	Description	27	135	235	236	237	Reference
6	Flexible Algorithm Prefix Metric (FAPM)	n	y	y	y	y	RFC 9350, Section 8

Table 5: IS-IS Sub-TLVs for TLVs Advertising Prefix Reachability Registry

18.3.3. IS-IS Sub-Sub-TLVs for Flexible Algorithm Definition Sub-TLV Registry

IANA has created the "IS-IS Sub-Sub-TLVs for Flexible Algorithm Definition Sub-TLV" registry within the "IS-IS TLV Codepoints" registry group. The registration procedure is "Expert Review" (note that the "IS-IS TLV Codepoints" registry group includes Expert Review guidance that applies to all registries thereunder).

The sub-sub-TLVs defined in this document have been assigned as follows.

Type	Description	Reference
0	Reserved	RFC 9350
1	Flexible Algorithm Exclude Admin Group	RFC 9350, Section 6.1
2	Flexible Algorithm Include-Any Admin Group	RFC 9350, Section 6.2
3	Flexible Algorithm Include-All Admin Group	RFC 9350, Section 6.3
4	Flexible Algorithm Definition Flags	RFC 9350, Section 6.4
5	Flexible Algorithm Exclude SRLG	RFC 9350, Section 6.5
6-255	Unassigned	

Table 6: IS-IS Sub-Sub-TLVs for Flexible Algorithm Definition Sub-TLV Registry

18.4. OSPF IANA Considerations

18.4.1. OSPF Router Information (RI) TLVs Registry

This document makes the following registration in the "OSPF Router Information (RI) TLVs" registry.

Value	Description	Reference
16	Flexible Algorithm Definition (FAD) TLV	RFC 9350, Section 5.2

Table 7: OSPF Router Information (RI) TLVs Registry

18.4.2. OSPFv2 Extended Prefix TLV Sub-TLVs Registry

This document makes the following registration in the "OSPFv2 Extended Prefix TLV Sub-TLVs" registry.

Value	Description	Reference
3	Flexible Algorithm Prefix Metric (FAPM)	RFC 9350, Section 9

Table 8: OSPFv2 Extended Prefix TLV Sub-TLVs Registry

18.4.3. OSPFv3 Extended-LSA Sub-TLVs Registry

This document makes the following registrations in the "OSPFv3 Extended-LSA Sub-TLVs" registry.

Value	Description	Reference
26	Flexible Algorithm Prefix Metric (FAPM)	RFC 9350, Section 9
33	OSPF Flexible Algorithm ASBR Metric	RFC 9350, Section 10.2

Table 9: OSPFv3 Extended-LSA Sub-TLVs Registry

18.4.4. OSPF Flex-Algorithm Prefix Metric Bits Registry

IANA has created the "OSPF Flex-Algorithm Prefix Metric Bits" registry under the "Open Shortest Path First (OSPF) Parameters" registry. The registration procedure is "IETF Review". Bits 1-7 are unassigned, and the initial value has been assigned as follows.

Bit Number	Description	Reference
0	E bit - External Type	RFC 9350, Section 9

Table 10: OSPF Flex-Algorithm Prefix Metric Bits Registry

18.4.5. Opaque Link-State Advertisements (LSA) Option Types Registry

This document makes the following registration in the "Opaque Link-State Advertisements (LSA) Option Types" registry within the "Open Shortest Path First (OSPF) Opaque Link-State Advertisements (LSA) Option Types" registry group.

Value	Opaque Type	Reference
11	OSPFv2 Extended Inter-Area ASBR (EIA-ASBR) LSA	RFC 9350, Section 10.1

Table 11: Opaque Link-State Advertisements (LSA) Option Types Registry

18.4.6. OSPFv2 Extended Inter-Area ASBR TLVs Registry

IANA has created the "OSPFv2 Extended Inter-Area ASBR TLVs" registry within the "Open Shortest Path First v2 (OSPFv2) Parameters" registry group. The registration procedure is "IETF Review" or "IESG Approval". The initial value has been assigned as follows.

Value	Description	Reference
1	Extended Inter-Area ASBR	RFC 9350

Table 12: OSPFv2 Extended Inter-Area ASBR TLVs Registry

The values 2-32767 are unassigned, the values 32768-33023 are reserved for Experimental Use, and the values 0 and 33024-65535 are reserved.

18.4.7. OSPFv2 Extended Inter-Area ASBR Sub-TLVs Registry

IANA has created the "OSPFv2 Extended Inter-Area ASBR Sub-TLVs" registry under the "Open Shortest Path First v2 (OSPFv2) Parameters" registry. The registration procedure is "IETF Review" or "IESG Approval". The initial value has been assigned as follows.

Value	Description	Reference
1	OSPF Flexible Algorithm ASBR Metric	RFC 9350

Table 13: OSPFv2 Extended Inter-Area ASBR Sub-TLVs Registry

The values 2-32767 are unassigned, the values 32768-33023 are reserved for Experimental Use, and the values 0 and 33024-65535 are reserved.

18.4.8. OSPF Flexible Algorithm Definition TLV Sub-TLVs Registry

IANA has created the "OSPF Flexible Algorithm Definition TLV Sub-TLVs" registry within the "Open Shortest Path First (OSPF) Parameters" registry group. The registration procedure is "IETF Review" or "IESG Approval".

The "OSPF Flexible Algorithm Definition TLV Sub-TLVs" registry will define sub-TLVs at any level of nesting for the Flexible Algorithm TLV, and new values can be allocated via the registration procedure.

This document registers the following sub-TLVs.

Bit Number	Description	Reference
0	Reserved	RFC 9350
1	Flexible Algorithm Exclude Admin Group	RFC 9350, Section 7.1
2	Flexible Algorithm Include-Any Admin Group	RFC 9350, Section 7.2
3	Flexible Algorithm Include-All Admin Group	RFC 9350, Section 7.3
4	Flexible Algorithm Definition Flags	RFC 9350, Section 7.4
5	Flexible Algorithm Exclude SRLG	RFC 9350, Section 7.5

Table 14: OSPF Flexible Algorithm Definition TLV Sub-TLVs Registry

The values 6-32767 are unassigned, and values 32768-33023 are for Experimental Use; these will not be registered with IANA.

Types in the range 33024-65535 are not to be assigned at this time. Before any assignments can be made in the 33024-65535 range, there **MUST** be an IETF specification that specifies IANA considerations that cover the range being assigned.

18.4.9. Link Attribute Application Identifiers Registry

This document registers the following bit in the "Link Attribute Application Identifiers" registry.

Bit	Description	Reference
3	Flexible Algorithm (X-bit)	RFC 9350, Section 12

Table 15: Link Attribute Application Identifiers Registry

19. References

19.1. Normative References

[ISO10589] ISO, "Information technology - Telecommunications and information exchange between systems - Intermediate System to Intermediate System intra-domain routeing information exchange protocol for use in conjunction with the protocol for providing the connectionless-mode network service (ISO 8473)", Second Edition, ISO/IEC 10589:2002, November 2002.

[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>.

[RFC4203] Kompella, K., Ed. and Y. Rekhter, Ed., "OSPF Extensions in Support of Generalized Multi-Protocol Label Switching (GMPLS)", RFC 4203, DOI 10.17487/RFC4203, October 2005, <https://www.rfc-editor.org/info/rfc4203>.

[RFC5250] Berger, L., Bryskin, I., Zinin, A., and R. Coltun, "The OSPF Opaque LSA Option", RFC 5250, DOI 10.17487/RFC5250, July 2008, <https://www.rfc-editor.org/info/rfc5250>.

[RFC5307] Kompella, K., Ed. and Y. Rekhter, Ed., "IS-IS Extensions in Support of Generalized Multi-Protocol Label Switching (GMPLS)", RFC 5307, DOI 10.17487/RFC5307, October 2008, <https://www.rfc-editor.org/info/rfc5307>.

[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>.

[RFC7684] Psenak, P., Gredler, H., Shakir, R., Henderickx, W., Tantsura, J., and A. Lindem, "OSPFv2 Prefix/Link Attribute Advertisement", RFC 7684, DOI 10.17487/RFC7684, November 2015, <https://www.rfc-editor.org/info/rfc7684>.

[RFC7770] Lindem, A., Ed., Shen, N., Vasseur, JP., Aggarwal, R., and S. Shaffer, "Extensions to OSPF for Advertising Optional Router Capabilities", RFC 7770, DOI 10.17487/RFC7770, February 2016, <https://www.rfc-editor.org/info/rfc7770>.

[RFC7981] Ginsberg, L., Previdi, S., and M. Chen, "IS-IS Extensions for Advertising Router Information", RFC 7981, DOI 10.17487/RFC7981, October 2016, <https://www.rfc-editor.org/info/rfc7981>.

[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>.

[RFC8362] Lindem, A., Roy, A., Goethals, D., Reddy Vallem, V., and F. Baker, "OSPFv3 Link State Advertisement (LSA) Extensibility", RFC 8362, DOI 10.17487/RFC8362, April 2018, <https://www.rfc-editor.org/info/rfc8362>.

[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>.

[RFC8665] Psenak, P., Ed., Previdi, S., Ed., Filsfils, C., Gredler, H., Shakir, R., Henderickx, W., and J. Tantsura, "OSPF Extensions for Segment Routing", RFC 8665, DOI 10.17487/RFC8665, December 2019, <https://www.rfc-editor.org/info/rfc8665>.

[RFC8666] Psenak, P., Ed. and S. Previdi, Ed., "OSPFv3 Extensions for Segment Routing", RFC 8666, DOI 10.17487/RFC8666, December 2019, <https://www.rfc-editor.org/info/rfc8666>.

[RFC8667] Previdi, S., Ed., Ginsberg, L., Ed., Filsfils, C., Bashandy, A., Gredler, H., and B. Decraene, "IS-IS Extensions for Segment Routing", RFC 8667, DOI 10.17487/RFC8667, December 2019, <https://www.rfc-editor.org/info/rfc8667>.

[RFC8919] Ginsberg, L., Psenak, P., Previdi, S., Henderickx, W., and J. Drake, "IS-IS Application-Specific Link Attributes", RFC 8919, DOI 10.17487/RFC8919, October 2020, <https://www.rfc-editor.org/info/rfc8919>.

[RFC8920] Psenak, P., Ed., Ginsberg, L., Henderickx, W., Tantsura, J., and J. Drake, "OSPF Application-Specific Link Attributes", RFC 8920, DOI 10.17487/RFC8920, October 2020, <https://www.rfc-editor.org/info/rfc8920>.

[RFC9352] Psenak, P., Ed., Filsfils, C., Bashandy, A., Decraene, B., and Z. Hu, "IS-IS Extensions to Support Segment Routing over the IPv6 Data Plane", RFC 9352, DOI 10.17487/RFC9352, February 2023, <https://www.rfc-editor.org/info/rfc9352>.

19.2. Informative References

[RFC2328] Moy, J., "OSPF Version 2", STD 54, RFC 2328, DOI 10.17487/RFC2328, April 1998, <https://www.rfc-editor.org/info/rfc2328>.

[RFC3101] Murphy, P., "The OSPF Not-So-Stubby Area (NSSA) Option", RFC 3101, DOI 10.17487/RFC3101, January 2003, <https://www.rfc-editor.org/info/rfc3101>.

[RFC3630] Katz, D., Kompella, K., and D. Yeung, "Traffic Engineering (TE) Extensions to OSPF Version 2", RFC 3630, DOI 10.17487/RFC3630, September 2003, <https://www.rfc-editor.org/info/rfc3630>.

[RFC3906] Shen, N. and H. Smit, "Calculating Interior Gateway Protocol (IGP) Routes Over Traffic Engineering Tunnels", RFC 3906, DOI 10.17487/RFC3906, October 2004, <https://www.rfc-editor.org/info/rfc3906>.

[RFC4552] Gupta, M. and N. Melam, "Authentication/Confidentiality for OSPFv3", RFC 4552, DOI 10.17487/RFC4552, June 2006, <https://www.rfc-editor.org/info/rfc4552>.

[RFC5304] Li, T. and R. Atkinson, "IS-IS Cryptographic Authentication", RFC 5304, DOI 10.17487/RFC5304, October 2008, <https://www.rfc-editor.org/info/rfc5304>.

[RFC5305] Li, T. and H. Smit, "IS-IS Extensions for Traffic Engineering", RFC 5305, DOI 10.17487/RFC5305, October 2008, <https://www.rfc-editor.org/info/rfc5305>.

[RFC5310] Bhatia, M., Manral, V., Li, T., Atkinson, R., White, R., and M. Fanto, "IS-IS Generic Cryptographic Authentication", RFC 5310, DOI 10.17487/RFC5310, February 2009, <https://www.rfc-editor.org/info/rfc5310>.

[RFC5340] Coltun, R., Ferguson, D., Moy, J., and A. Lindem, "OSPF for IPv6", RFC 5340, DOI 10.17487/RFC5340, July 2008, <https://www.rfc-editor.org/info/rfc5340>.

[RFC6571] Filsfils, C., Ed., Francois, P., Ed., Shand, M., Decraene, B., Uttaro, J., Leymann, N., and M. Horneffer, "Loop-Free Alternate (LFA) Applicability in Service Provider (SP) Networks", RFC 6571, DOI 10.17487/RFC6571, June 2012, <https://www.rfc-editor.org/info/rfc6571>.

[RFC7120] Cotton, M., "Early IANA Allocation of Standards Track Code Points", BCP 100, RFC 7120, DOI 10.17487/RFC7120, January 2014, <https://www.rfc-editor.org/info/rfc7120>.

[RFC7471] Giacalone, S., Ward, D., Drake, J., Atlas, A., and S. Previdi, "OSPF Traffic Engineering (TE) Metric Extensions", RFC 7471, DOI 10.17487/RFC7471, March 2015, <https://www.rfc-editor.org/info/rfc7471>.

[RFC7474] Bhatia, M., Hartman, S., Zhang, D., and A. Lindem, Ed., "Security Extension for OSPFv2 When Using Manual Key Management", RFC 7474, DOI 10.17487/RFC7474, April 2015, <https://www.rfc-editor.org/info/rfc7474>.

[RFC8126] Cotton, M., Leiba, B., and T. Narten, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 8126, DOI 10.17487/RFC8126, June 2017, <https://www.rfc-editor.org/info/rfc8126>.

[RFC8570] Ginsberg, L., Ed., Previdi, S., Ed., Giacalone, S., Ward, D., Drake, J., and Q. Wu, "IS-IS Traffic Engineering (TE) Metric Extensions", RFC 8570, DOI 10.17487/RFC8570, March 2019, <https://www.rfc-editor.org/info/rfc8570>.

[RFC8986] Filsfils, C., Ed., Camarillo, P., Ed., Leddy, J., Voyer, D., Matsushima, S., and Z. Li, "Segment Routing over IPv6 (SRv6) Network Programming", RFC 8986, DOI 10.17487/RFC8986, February 2021, <https://www.rfc-editor.org/info/rfc8986>.

[ROUTING-PLANES-USING-SR] Hegde, S. and A. Gulko, "Separating Routing Planes using Segment Routing", Work in Progress, Internet-Draft, draft-gulkohegde-routing-planes-using-sr-00, 13 March 2017, <https://datatracker.ietf.org/doc/html/draft-gulkohegde-routing-planes-using-sr-00>.

[RTGWG-SEGMENT-ROUTING-TI-LFA] Litkowski, S., Bashandy, A., Filsfils, C., Francois, P., Decraene, B., and D. Voyer, "Topology Independent Fast Reroute using Segment Routing", Work in Progress, Internet-Draft, draft-ietf-rtgwg-segment-routing-ti-lfa-09, 23 December 2022, <https://datatracker.ietf.org/doc/html/draft-ietf-rtgwg-segment-routing-ti-lfa-09>.

Acknowledgements

This document, among other things, addresses the problem that [ROUTING-PLANES-USING-SR] was trying to solve. All authors of that document agreed to join this document.

Thanks to Eric Rosen, Tony Przygienda, William Britto A. J., Gunter Van de Velde, Dirk Goethals, Manju Sivaji, and Baalajee S. for their detailed review and excellent comments.

Thanks to Cengiz Halit for his review and feedback during the initial phase of the solution definition.

Thanks to Kenji Kumaki for his comments.

Thanks to Acee Lindem for editorial comments.

Authors' Addresses

Peter Psenak (EDITOR)
Cisco Systems, Inc.
Apollo Business Center
Mlynske nivy 43
82109 Bratislava
Slovakia
Email: ppsenak@cisco.com

Shraddha Hegde
Juniper Networks, Inc.
Embassy Business Park
Bangalore 560093
KA
India
Email: shraddha@juniper.net

Clarence Filsfils
Cisco Systems, Inc.
Brussels
Belgium
Email: cfilsfil@cisco.com

Ketan Talaulikar
Cisco Systems, Inc
India
Email: ketant.ietf@gmail.com

Arkadiy Gulko
Edward Jones
Email: arkadiy.gulko@edwardjones.com