Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: 23 August 2026

T. Przygienda
S. Hegde
J. Head
HPE
A. Lindem
Arrcus, Inc

19 February 2026

IGP Flooding Reduction Algorithms Framework
draft-ietf-lsr-flood-reduction-arch-01

Abstract

This document introduces a framework making it possible to deploy multiple flood reduction algorithms within the same IGP domain in an interoperable fashion.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

Scenarios exist where multiple distributed (or centralized) flood reduction algorithms may be deployed simultaneously within an IGP domain. These scenarios necessitate certain agreed on cooperative behaviors between the involved algorithms to ensure the correctness of the overall solution. This is true in both permanent and transient (i.e., migration) deployment cases. Fortunately, existing graph theory concepts allow to provide guidance toward the design of algorithms with the necessary properties to ensure their interoperable coexistence.

This document presents the necessary requirements for the involved algorithms and the details of a framework for their interoperable deployment. Although running multiple algorithms simultaneously may not be a preferred operational choice, it is necessary if the migration from one algorithm to another with minimal network disruption is a priority. A migration itself may be caused by the discovery of defects in the deployed algorithms or the deployment of new algorithms that offer improvements.

Dealing with interoperability or lack thereof between this framework and other published frameworks such as e.g. [RFC9667] is explicitly outside the scope of this document.

2. Flooding Pruner Framework

2.1. Definitions and Axioms

This section outlines a framework that allows the operation of multiple different flood reduction algorithms (called _flooding pruners_ or _pruners_ from here on) in an interoperable fashion.

An important observation upfront, which will become clear later in this section, is that full, non-optimized flooding presents a special case of a pruner itself. Normal flooding includes all adjacencies without any pruning, and hence we name it the _non-pruner_ or _zero_ for short.

2.1.1. Maximum of One Flooding Pruner on a Node

This framework permits the use of at most one pruner on each node. It allows to change a specific pruner at any time on any subset of nodes in the network while limiting the impact to the node itself and possibly the re-convergence of a set of nodes within its connected component.

2.1.2. Connected Component

A _connected component_ (or component for short) is defined as a subset of all nodes in the network running the same pruner, e.g. A (such nodes are denoted by the set notation of A|) where each of the nodes has to be connected to all other nodes by a path that traverses only nodes that run A. Observe that there well may be in the network multiple components that are not connected, but that run the same pruner algorithm. We denote in such case a component for pruner A as A|, and if two _disjoint_ components running the same algorithm A are present in the network, we denote such second set as A|’ and by extension we use A|’’ notation for further such sets.

Non-pruners also build components denoted as Z| and its primes.

Another way to visualize components is to consider a network running multiple pruners as "islands running pruning algorithms" that are connected to each other by components running non-pruners (i.e. using normal flooding).

2.1.3. Flooding Connected Dominating Sets

A pruner may choose within its component a subset of links to flood while making sure that the component remains connected. In other words, after suppressing flooding on some links within the component there must still exist paths consisting of the remaining links that connect each pair of nodes in the component. We use for such remaining links the term _flooding connected dominating set_ or CDS for short (more precisely, a not necessarily loop-free edge dominating set). Such a CDS is colloquially often called _flooding topology_ in context of flood reduction algorithms. A simple spanning tree is an easily visualized special case of a CDS. We denote such a CDS for a component A| as A|*. A|* is often not unique for a component and many different sets of links can be a CDS. Nor is it required that a CDS has to be loop-free since there may be many different paths on the CDS between two nodes in a component. Therefore, it is possible in a most extreme case that each LSP is flooded on a different CDS.

To summarize the section above in simple terms, a pruner must choose at least one set of flooding links that guarantees that all information can reach all the nodes in the component.

2.1.4. Rules for Flooding Pruners

Any flood reduction algorithm expecting to interoperate with other algorithms within this framework but without having to understand their behavior MUST adhere to the following rules. Otherwise, the algorithm cannot be expected to accommodate other algorithms in the network at the same time or is in other words a ship in the night.

1. Each node of a pruner (except the non-pruner) MUST advertise in its flooded node information the currently active pruner. It MUST also understand such information as advertised by other nodes in the network. A node running a pruner MUST NOT assume implicitly that a node is a _non-pruner_ or supports or runs the same algorithm. However, any pruner can safely assume that any node that does not advertise any running pruner in its node information MUST be a non-pruner. Observe that a pruner does not need to understand how the algorithm of another pruner operates (or even whether it is centralized, centrally signalled or fully

distributed). The only requirement is that every pruner uses the same signaling information provided in this framework which indicates the pruner currently running.

2. A pruner MUST NOT prune links in components other than the one it participates in or assume flooding behavior on links in other components (except in the case of a _non-pruner_ where the flooding is well understood). In other words, each pruner is allowed to prune some links from flooding, but only strictly within its own component.

3. A flooding pruner A MUST also include in its flooding CDS all links to adjacent components running a non _non-pruner_ different from A. A node running pruner P that is different from the _non-pruner_ SHOULD include in its flooding CDS all links to non-pruners. It MAY use the known behavior the _non-pruner_ for further optimizations. Nevertheless, such optimizations MUST NOT assume that there is just a single Z| in the network. This is sufficient (but strictly speaking, more than necessary) to guarantee that the overall set of flooding CDSes within each component creates an overall flooding CDS over the whole network. In other words, the resulting set of links that still flood connects all nodes in the network.

This document does not consider other approaches that guarantee a pruner property on e.g. a clique, i.e. a subgraph where every vertex is neighbored to all other vertices in the clique. It assumes that such "ship in the night components" can be considered non-pruners due to their implicit guarantee of correct flooding to nodes that are part of their component where connected to other components.

2.2. Beneficial Properties of the Flooding Pruner Framework

Nodes are free to use any kind of pruner to calculate optimized flooding. Any mode of computation, distributed or centralized, will work fine as long as it adheres to Section 2.1.4. Per Section 2.1.2 a node will become part of one and exactly one component after choosing a pruner.

The framework allows but does not assume any centralized instance or election in a component. Computation and communication within each component is completely independent of other components.

A node is free to choose a different pruner or a _non-pruner_ at any point in time independent of all other nodes. It may end up in another component or become a _non-pruner_ with the maximum impact consisting of re-computation within two components that see such node leave or join. For a distributed algorithm, it is likely that only the adjoining nodes have to adjust their pruning decisions. That is to say, the framework allows for node-by-node deployment or migration of pruners without networkwide recomputation of optimized flooding. This is obviously critical to the stability of large networks that may not converge within reasonable time if the whole network were to revert to non-pruning due to networkwide impact. However, such behavior cannot be excluded, for example, in case of election problems due to misconfiguration or topological separation of nodes if the whole network runs a single pruner relying on centralized election. The network itself cannot ensure correctness of a pruner or prevent a pruner having a blast radius of the whole component depending upon the algorithm and further signaling used.

Although the framework provides extreme operational flexibility when deploying pruners, the most likely scenarios are a node-by-node deployment of a single pruner in addition to a non-pruner or, if the necessity arises, a node-by-node migration to another pruner.

2.3. Example

```
        +==================+         +==================+
        | A|'   +--+       |         | B|    +--+       |
        |    +--|  |---+   |         |    +--|  |---+   |
        |    |  +--+   |   |         |    |  +--+   |   |
        |  +--+       +--+ |         |  +--+       +--+ |
        |  |  |---X---|  |==============|  |---X---|  | ========#
        |  +--+       +--+ |         |  +--+       +--+ |       #
        +====#==========#==+         +====#==========#==+       #
             ########   #                 #       # =#          #
           +========#===#=====+   +=======#====+  #   +=========#===+
           | Z|'   +--+ #     |   | Z|'' +--+  |  #   | Z|''' +--+  |
           |       |  |##     |   |      |  |  |  #   |       |  |  |
           |       +--+       |   |      +--+  |  #   |       +--+  |
           |  +--+  |    +--+ |   |       #    |  #   |         #   |
           |  |  |--+----|  | |   +=======#====+  #   +=========#===+
           |  +--+       +--+ |           #       #             #
           +====#=============+    #======#       #             #
                #                  #              #             #
                #         +========#==========+   #             #
                #         | A|''  +--+        |   #             #
                #         |    +--|  |---+    |   #             #
                #         |    |  +--+   |    |   #             #
                #         |  +--+       +--+======#             #
                #============|  |---X---|  |====================#
                          |  +--+       +--+ |
                          +===================+

           X : removed link due to reduction
           # : link included on component boundary
           = : link included on component boundary
```

Figure 1: Network of Mixed Pruners

Figure 1 illustrates a network with three pruners running. Two components run pruner A and are denoted as A|' and A|'' and one component runs pruner B. Remaining three components run the _non-pruner_ algorithm (annotated as Z|', Z|'', and Z|'''). CDSes within components are shown by indicating the links that were pruned from flooding as crossed out. Additionally, the links that are included to connect the CDS of the component following the rules listed in Section 2.1.4 have been made thicker. Despite multiple algorithms and components being present in the network, the complete graph is obviously still covered by the involved CDSes.

Figure 1 also illustrates why the overall CDS can easily be more than just a spanning tree of the overall network. A node seeing its neighbor running another algorithm cannot always decide based on local knowledge whether the link should be included in flooding or not. Such a decision could be based on the overall view of the network using some global tie-breaking algorithm. However, due to the potential long flooding paths and one-link minimal cuts, such an algorithm is not considered here but could be proposed in the future.

2.4. Signaling

Nodes are REQUIRED to signal when they are actively running any pruner (except a non-pruner). The absence of such an advertisement indicates that a node is either running a non-pruner or some other algorithm where within its component it behaves in an equivalent fashion while also guaranteeing flooding on links connected to other components.

2.4.1. IS-IS - Flood Reduction Algorithm Sub-TLV

The Flood Reduction Algorithm Sub-TLV is advertised as a sub-TLV within a Router Capability TLV (242) defined in [RFC7981]. It uses the following format:

```
 0                   1                   2
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Type      |     Length    |   Algorithm   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 2

* *Type:*TBD1

* *Length:*1

* *Algorithm:*An 8-bit numerical identifier allocated from the "IGP Algorithm Type For Computing Flooding Topology" registry.

This sub-TLV MUST only contain the currently active running pruner (other than a non-pruner). It MUST be flooded within a Router Capability TLV that is strictly area-scoped and MUST NOT be leaked between levels.

2.4.2. OSPFv2/OSPFv3 - Flood Reduction Algorithm TLV

The Flood Reduction Algorithm TLV is advertised within a OSPFv2 Router Information (RI) Opaque LSA or OSPFv3 Router Information LSA as defined in [RFC7770]. It uses 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            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   Algorithm   |
+-+-+-+-+-+-+-+-+
```

Figure 3

* *Type:*TBD1

* *Length:*1

* *Algorithm:*An 8-bit numerical identifier allocated from the "IGP Algorithm Type For Computing Flooding Topology" registry.

Nodes MUST advertise this TLV when they actively run any pruner (other than a non-pruner).

Pruner signaling MUST be flooded in a way that is strictly area-scoped that will not leak between areas. For OSPFv2, the TLV MUST be flooded within a Router Information (RI) Opaque LSA as Type-10. For OSPFv3, the TLV MUST be flooded within a Router Information (RI) LSA with the S2-bit set to 1.

2.4.3. Error Handling

This section defines error handling regardless of the deployed IGP or active algorithm.

If a node receives an advertisement with more than one algorithm, it SHOULD be logged at least once.

If a node receives an advertisement with an unrecognized/invalid algorithm, it SHOULD be logged at least once.

If a node receives an advertisement with a mismatched algorithm, it SHOULD be logged at least once.

3. YANG Data Model

YANG [RFC7950] is a data definition language used to define the contents of a conceptual data store that allows networked devices to be managed using NETCONF [RFC6241] or RESTCONF [RFC8040].

This section defines three YANG modules. Module iana-igp-flooding-topo-algo defines the identities the 'IGP AlgorithmType for Computing Flooding Topology' in the 'Interior Gateway Protocol (IGP) Parameters' registry group. Section 4.1. Module ietf-isis-flooding-reduction-algo, which augments the IS-IS YANG data model [RFC9130] and the YANG Data Model for Routing Management [RFC8349], can be used to configure and manage a flooding reduction algorithm in an IS-IS level 1 or level 2 area. Module ietf-ospf-flooding-reduction-algo, which augments the OSPF YANG data model [RFC9129] and the YANG Data Model for Routing Management [RFC8349], can be used to configure and manage a flooding reduction algorithm in an OSPF area.

This document uses the graphical representation of data model per [RFC8340].

3.1. Tree for IS-IS Flooding Reduction Algorithm

The following shows the tree diagram of the module for an IS-IS Flooding Reduction Algorithm:

```
  augment /rt:routing/rt:control-plane-protocols
            /rt:control-plane-protocol/isis:isis:
    +--rw level-1
    |  +--rw flooding-reduction-algo
    |     +--rw flooding-reduction-algo?   identityref
    +--rw level-2
       +--rw flooding-reduction-algo
          +--rw flooding-reduction-algo?   identityref
  augment /rt:routing/rt:control-plane-protocols
            /rt:control-plane-protocol/isis:isis/isis:database
            /isis:levels/isis:lsp/isis:router-capabilities:
    +--ro flooding-reduction-algo-tlv
       +--ro flooding-reduction-algo?   identityref
```

3.2. Tree for OSPF Flooding Reduction Algorithm

The following shows the tree diagram of the module for an OSPF Flooding Reduction Algorithm:

```
     augment /rt:routing/rt:control-plane-protocols
               /rt:control-plane-protocol/ospf:ospf/ospf:areas
               /ospf:area:
       +--rw flooding-reduction-algo
          +--rw flooding-reduction-algo?   identityref
     augment /rt:routing/rt:control-plane-protocols
               /rt:control-plane-protocol/ospf:ospf/ospf:areas
               /ospf:area/ospf:database/ospf:area-scope-lsa-type
               /ospf:area-scope-lsas/ospf:area-scope-lsa/ospf:version
               /ospf:ospfv2/ospf:ospfv2/ospf:body/ospf:opaque
               /ospf:ri-opaque:
       +--ro flooding-reduction-algo-tlv
          +--ro flooding-reduction-algo?   identityref
     augment /rt:routing/rt:control-plane-protocols
               /rt:control-plane-protocol/ospf:ospf/ospf:areas
               /ospf:area/ospf:database/ospf:area-scope-lsa-type
               /ospf:area-scope-lsas/ospf:area-scope-lsa/ospf:version
               /ospf:ospfv3/ospf:ospfv3/ospf:body
               /ospf:router-information:
       +--ro flooding-reduction-algo-tlv
          +--ro flooding-reduction-algo?   identityref
```

3.3. IANA Module for IGP Flooding Algorithms

IANA has created a registry titled ’IGP Algorithm Type for Computing Flooding Topology’ in the ’Interior Gateway Protocol (IGP) Parameters’ registry group."; Module iana-igp-flooding-topo-algo is an IANA-maintained module, which defines the identities for the IGP algorithms for computing flooding topologies as in the IANA "IGP Algorithm Type for Computing Flooding Topology" registry.

This module is maintained by IANA and will be updated if and when there is any change to the registry.

This document defines the initial version of the IANA-maintained YANG module for the "IGP Algorithm Type for Computing Flooding Topology" registry. registry Section 4.1.

```
<CODE BEGINS> file "iana-igp-flooding-topo-algo@2026-02-10.yang"
module iana-igp-flooding-topo-algo {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:"
          + "iana-igp-flooding-topo-algo";
  prefix iana-igp-flood-algo;

  organization
    "Internet Assigned Numbers Authority (IANA)";
  contact
```

```
       "Internet Assigned Numbers Authority

        ICANN
        12025 Waterfront Drive, Suite 300
        Los Angeles, CA 90094-2536
        United States of America

        Tel:    +1 310 301 5800
        <mailto:iana@iana.org>";
     description
       "This YANG module defines the identities for IGP Algorithms
        for Computing Flooding Topologies.

        This YANG module is maintained by IANA and reflects the
        'IGP Flooding Reduction Algorithm' registry.

        Copyright (c) 2026 IETF Trust and the persons identified as
        authors of the code.  All rights reserved.

        Redistribution and use in source and binary forms, with or
        without modification, is permitted pursuant to, and subject
        to the license terms contained in, the Revised BSD License
        set forth in Section 4.c of the IETF Trust's Legal Provisions
        Relating to IETF Documents
        (https://trustee.ietf.org/license-info).

        All revisions of IETF and IANA published modules can be found
        at the YANG Parameters registry group
        (https://www.iana.org/assignments/yang-parameters).

        This initial version of this YANG module is part of RFC XXXX
        (https://www.rfc-editor.org/info/rfcXXXX); see the RFC itself
        for full legal notices.

        The latest version of this YANG module is available at
        https://www.iana.org/assignments/yang-parameters.";

     revision 2026-02-10 {
       description
         "Initial version";
       reference
         "RFC 9667: Dynamic Flooding on Dense Graphs";
     }

     identity igp-flooding-topo-algo {
       description
         "Base identity for IGP algorithms for computing flooding
          topologies. Algorithms are defined in the 'IGP Algorithm
```

```
          Type for Computing Flooding Topology' in the 'Interior
          Gateway Protocol (IGP) Parameters' registry group.";
     }
   }
   <CODE ENDS>
```

3.4. YANG Module for IS-IS Flooding Reduction Algorithm

The following is the YANG module for IS-IS Flooding Reduction Algorithm:

```
   <CODE BEGINS> file "ietf-isis-flooding-reduction-algo@2026-02-10.yang"
   module ietf-isis-flooding-reduction-algo {
     yang-version 1.1;
     namespace "urn:ietf:params:xml:ns:yang:"
             + "ietf-isis-flooding-reduction-algo";
     prefix isis-flood-reduce-algo;

     import ietf-routing {
       prefix rt;
       reference
         "RFC 8349: A YANG Data Model for Routing Management
                    (NMDA Version)";
     }
     import ietf-isis {
       prefix isis;
       reference
         "RFC 9130: YANG Data Model for the IS-IS Protocol";
     }
     import iana-igp-flooding-topo-algo {
       prefix iana-igp-flood-algo;
       reference
         "RFC 9667: Dynamic Flooding on Dense Graphs";
     }

     organization
       "IETF Link State Routing (LSR) Working Group";
     contact
       "WG Web:   <https://datatracker.ietf.org/wg/lsr/>
        WG List:  <mailto:lsr@ietf.org>

        Author:   Acee Lindem
                  <mailto:acee.ietf@gmail.com>";
     description
       "This YANG module defines the operational state for
        the flooding reduction algorithm in IS-IS as defined
        in RFC XXXX.
```

```
        Copyright (c) 2026 IETF Trust and the persons identified as
        authors of the code.  All rights reserved.
        Redistribution and use in source and binary forms, with or
        without modification, is permitted pursuant to, and subject
        to the license terms contained in, the Revised BSD License
        set forth in Section 4.c of the IETF Trust's Legal Provisions
        Relating to IETF Documents
        (https://trustee.ietf.org/license-info).

        All revisions of IETF and IANA published modules can be found
        at the YANG Parameters registry group
        (https://www.iana.org/assignments/yang-parameters).

        This version of this YANG module is part of RFC XXXX; see
        the RFC itself for full legal notices.";

     revision 2026-02-10 {
       description
         "Initial version";
       reference
         "RFC XXXX: IGP Flooding Reduction Algorithms Framework";
     }

     /*
      * IS-IS Flooding Reduction Algorithm configuration
      */

     grouping flooding-reduction-algo-config {
       description
         "Grouping for IS-IS flooding algorithm router
          configuration.";
       container flooding-reduction-algo {
         description
           "IS-IS Level-1 flooding algorithm routing configuration.
            This containers includes configuration related to the
            active flooding reduction algorithm.";
         leaf flooding-reduction-algo {
           type identityref {
             base iana-igp-flood-algo:igp-flooding-topo-algo;
           }
           description
             "Specifies the active flooding reduction algorithm
              in for IS-IS Level 1.
              RFC XXXX: IGP Flooding Reduction Algorithms Framework";
         }
       }
     }
```

```
     augment "/rt:routing/"
           + "rt:control-plane-protocols/rt:control-plane-protocol"
           + "/isis:isis" {
       when "/rt:routing/rt:control-plane-protocols/"
          + "rt:control-plane-protocol/rt:type = 'isis:isis'" {
         description
           "This augment IS-IS routing protocol when used";
       }
       description
         "This augments IS-IS with flooding reduction algorithm
          configuration.";
       container level-1 {
         uses flooding-reduction-algo-config;
         description
           "IS-IS Level-1 Flooding Reduction Algorithm
            configuration.";
       }
       container level-2 {
         uses flooding-reduction-algo-config;
         description
           "IS-IS Level-2 Flooding Reduction Algorithm
            configuration.";
       }
     }

     /*
      * IS-IS Flooding Reduction Algorithm Router Capabilities
      */

     grouping flooding-reduction-algorithm-tlv {
       description
         "Grouping for IS-IS flooding reduction algorithm
          TLV types.";
       reference
         "RFC 7981: IS-IS Extensions for Advertising Router
                    Information
          RFC XXXX: Flooding Reduction Algorithms Framework";
       container flooding-reduction-algo-tlv {
         description
           "IS-IS Router Flooding Algorithm TLV.";
         leaf flooding-reduction-algo {
           type identityref {
             base iana-igp-flood-algo:igp-flooding-topo-algo;
           }
           description
             "Flooding reduction algorithm active in the
              IS-IS routing domain";
           reference
```

```
             "RFC 7981: IS-IS Extensions for Advertising Router
                        Information
              RFC XXXX: Flooding Reduction Algorithms Framework";
          }
        }
      }

      augment "/rt:routing/"
            + "rt:control-plane-protocols/rt:control-plane-protocol"
            + "/isis:isis/isis:database/isis:levels/isis:lsp"
            + "/isis:router-capabilities" {
        when "/rt:routing/rt:control-plane-protocols/"
           + "rt:control-plane-protocol/"
           + "rt:type = 'isis:isis'" {
          description
            "This augment IS-IS routing protocol when used";
        }
        description
          "This augments IS-IS protocol LSDB router capability.";
        uses flooding-reduction-algorithm-tlv;
      }
    }
    <CODE ENDS>
```

3.5. YANG Module for OSPF Flooding Reduction Algorithm

The following is the YANG module for OSPF Flooding Reduction Algorithm:

```
    <CODE BEGINS> file "ietf-ospf-flooding-reduction-algo@2026-02-10.yang"
    module ietf-ospf-flooding-reduction-algo {
      yang-version 1.1;
      namespace "urn:ietf:params:xml:ns:yang:"
              + "ietf-ospf-flooding-reduction-algo";
      prefix ospf-fc;

      import ietf-routing {
        prefix rt;
        reference
          "RFC 8349: A YANG Data Model for Routing Management
                     (NMDA Version)";
      }
      import ietf-ospf {
        prefix ospf;
        reference
          "RFC 9129: YANG Data Model for the OSPF Protocol";
      }
      import iana-igp-flooding-topo-algo {
```

```
     prefix iana-igp-flood-algo;
     reference
       "RFC 9667: Dynamic Flooding on Dense Graphs";
   }

   organization
     "IETF Link State Routing (LSR) Working Group";
   contact
     "WG Web:   <https://datatracker.ietf.org/wg/lsr/>
      WG List:  <mailto:lsr@ietf.org>

      Author:   Acee Lindem
                <mailto:acee.ietf@gmail.com>";
   description
     "This YANG module defines the configuration and operational
      state for the flooding reduction algorithms in OSPF as
      defined in RFC XXXX.

      Copyright (c) 2026 IETF Trust and the persons identified as
      authors of the code.  All rights reserved.
      Redistribution and use in source and binary forms, with or
      without modification, is permitted pursuant to, and subject
      to the license terms contained in, the Revised BSD License
      set forth in Section 4.c of the IETF Trust's Legal Provisions
      Relating to IETF Documents
      (https://trustee.ietf.org/license-info).

      All revisions of IETF and IANA published modules can be found
      at the YANG Parameters registry group
      (https://www.iana.org/assignments/yang-parameters).

      This version of this YANG module is part of RFC XXXX; see
      the RFC itself for full legal notices.";

   revision 2026-02-10 {
     description
       "Initial version";
     reference
       "RFC XXXX: IGP Flooding Reduction Algorithms Framework";
   }

   grouping flooding-reduction-algorithm-tlv {
     description
       "Grouping for OSPF flooding algorithm capabilities
        TLV types.";
     reference
       "RFC 7770: Extensions to OSPF for Advertising Optional
                  Router Capabilities
```

```
          RFC XXXX: IGP Flooding Reduction Algorithms Framework";
       container flooding-reduction-algo-tlv {
         description
           "OSPF Flooding Reduction Algorithm TLV.";
         leaf flooding-reduction-algo {
           type identityref {
             base iana-igp-flood-algo:igp-flooding-topo-algo;
           }
           description
             "Flooding reduction algorithm active in the
              OSPF routing domain area
              RFC XXXX: Flooding Reduction Algorithms Framework";
         }
       }
     }

     /*
      * OSPF Flooding Reduciton Algorithm Configuration
      */

     augment "/rt:routing/rt:control-plane-protocols/"
           + "rt:control-plane-protocol/ospf:ospf/"
           + "ospf:areas/ospf:area" {
       when "derived-from-or-self(/rt:routing/"
          + "rt:control-plane-protocols/"
          + "rt:control-plane-protocol/rt:type, 'ospf:ospf')" {
         description
           "This augmentation is only valid for both
            OSPFv2 and OSPFv3";
       }
       container flooding-reduction-algo {
         description
           "OSPF area-level flooding reduction algorithm routing
            configuration. This container includes configuration related
            to the active flooding reduction algorithm.";
         leaf flooding-reduction-algo {
           type identityref {
             base iana-igp-flood-algo:igp-flooding-topo-algo;
           }
           description
             "Specifies the active flooding reduction algorithm
              in the OSPF area.
              RFC XXXX: IGP Flooding Reduction Algorithms Framework";
         }
       }
       description
         "This augments the OSPF protocol area configuration with
          flood reduction algorithm specification.";
```

```
     }

     /*
      * OSPFv2 Router-Information Opaque LSA Flooding Reduction TLV
      */

     augment "/rt:routing/"
           + "rt:control-plane-protocols/rt:control-plane-protocol/"
           + "ospf:ospf/ospf:areas/"
           + "ospf:area/ospf:database/"
           + "ospf:area-scope-lsa-type/ospf:area-scope-lsas/"
           + "ospf:area-scope-lsa/ospf:version/ospf:ospfv2/"
           + "ospf:ospfv2/ospf:body/ospf:opaque/"
           + "ospf:ri-opaque" {
       when "derived-from-or-self(/rt:routing/"
          + "rt:control-plane-protocols/"
          + "rt:control-plane-protocol/rt:type, 'ospf:ospfv2')" {
         description
           "This augmentation is only valid for OSPFv2.";
       }
       description
         "OSPFv2 Opaque Area-Scoped Router-Information LSA Router
          Functional capabilities.";
       uses flooding-reduction-algorithm-tlv;
       reference
         "RFC 7770: Extensions to OSPF for Advertising Optional
                    Router Capabilities
          RFC XXXX: IGP Flooding Reduction Algorithms Framework";
     }

     /*
      * OSPFv3 Router-Information LSA Flooding Reduction TLV
      */

     augment "/rt:routing/"
           + "rt:control-plane-protocols/rt:control-plane-protocol/"
           + "ospf:ospf/ospf:areas/"
           + "ospf:area/ospf:database/"
           + "ospf:area-scope-lsa-type/ospf:area-scope-lsas/"
           + "ospf:area-scope-lsa/ospf:version/ospf:ospfv3/"
           + "ospf:ospfv3/ospf:body/ospf:router-information" {
       when "derived-from-or-self(/rt:routing/"
          + "rt:control-plane-protocols/"
          + "rt:control-plane-protocol/rt:type, 'ospf:ospfv3')" {
         description
           "This augmentation is only valid for OSPFv3.";
       }
       description
```

```
         "OSPFv3 Area-Scoped Router-Information LSA Router
          Functional capabilities.";
       uses flooding-reduction-algorithm-tlv;
       reference
         "RFC 7770: Extensions to OSPF for Advertising Optional
                    Router Capabilities
          RFC XXXX: IGP Flooding Reduction Algorithms Framework";
     }
   }
   <CODE ENDS>
```

4. IANA Considerations

4.1. IANA IGP Flooding Algorithm Registry

The existing IANA registry "IGP Algorithm Type for Computing Flooding Topology" in the "Interior Gateway Protocol (IGP) Parameters" registry group [RFC9667] is reused for algorithm definitions in this document. No further IANA action is required.

4.2. IGP Flooding Reduction Algorithm Capabiity

This section requests that IANA reserve a single value from each of the following registries. Both should be named: "Flood Reduction Algorithm".

* IS-IS Sub-TLVs for IS-IS Router CAPABILITY TLV Registry (from within the 1-32767 range)

* OSPF Router Information (RI) TLVs Registry

5. Security Considerations

This document outlines a framework for extensions to an IGP protocol for operation on high-density network topologies. Implementations SHOULD implement cryptographic authentication compliant to e.g. [RFC5304], and should enable other security measures in accordance with the best common practices for the relevant IGP protocol.

6. Contributors

The following people have contributed to this draft and are mentioned without any particular order: Raj Chetan, Les Ginsberg, Peter Psenak and Tony Li.

7. Normative References

[RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011, <https://www.rfc-editor.org/info/rfc6241>.

[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>.

[RFC7950] Bjorklund, M., Ed., "The YANG 1.1 Data Modeling Language", RFC 7950, DOI 10.17487/RFC7950, August 2016, <https://www.rfc-editor.org/info/rfc7950>.

[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>.

[RFC8040] Bierman, A., Bjorklund, M., and K. Watsen, "RESTCONF Protocol", RFC 8040, DOI 10.17487/RFC8040, January 2017, <https://www.rfc-editor.org/info/rfc8040>.

[RFC8340] Bjorklund, M. and L. Berger, Ed., "YANG Tree Diagrams", BCP 215, RFC 8340, DOI 10.17487/RFC8340, March 2018, <https://www.rfc-editor.org/info/rfc8340>.

[RFC8349] Lhotka, L., Lindem, A., and Y. Qu, "A YANG Data Model for Routing Management (NMDA Version)", RFC 8349, DOI 10.17487/RFC8349, March 2018, <https://www.rfc-editor.org/info/rfc8349>.

[RFC9129] Yeung, D., Qu, Y., Zhang, Z., Chen, I., and A. Lindem, "YANG Data Model for the OSPF Protocol", RFC 9129, DOI 10.17487/RFC9129, October 2022, <https://www.rfc-editor.org/info/rfc9129>.

[RFC9130] Litkowski, S., Ed., Yeung, D., Lindem, A., Zhang, J., and L. Lhotka, "YANG Data Model for the IS-IS Protocol", RFC 9130, DOI 10.17487/RFC9130, October 2022, <https://www.rfc-editor.org/info/rfc9130>.

8. Informative References

[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>.

[RFC9667] Li, T., Ed., Psenak, P., Ed., Chen, H., Jalil, L., and S. Dontula, "Dynamic Flooding on Dense Graphs", RFC 9667, DOI 10.17487/RFC9667, October 2024, <https://www.rfc-editor.org/info/rfc9667>.

Authors’ Addresses

Tony Przygienda
HPE
1701 East Mossy Oaks Road
Spring, TX 77389
United States of America
Email: antoni.przygienda@hpe.com

Shraddha Hegde
HPE
1701 East Mossy Oaks Road
Spring, TX 77389
United States of America
Email: shraddha.hegde@hpe.com

Jordan Head
HPE
1701 East Mossy Oaks Road
Spring, TX 77389
United States of America
Email: jordan.head@hpe.com

Acee Lindem
Arrcus, Inc
301 Midenhall Way
Cary, NC 27513
United States of America
Email: acee.ietf@gmail.com