

LSR
Internet-Draft
Intended status: Standards Track
Expires: July 12, 2020

M. Rasool, Ed.
S. Sreenivasan, Ed.
M. Negi, Ed.
Huawei Technologies
January 9, 2020

YANG Model for IGP Flexible Algorithm
draft-rasool-lsr-flex-algo-yang-01

Abstract

This document describes a YANG data model for IGP Flexible Algorithm. The model serves as a base framework for configuring and managing Flexible Algorithms which is used by IGPs to compute constraint based paths over the network.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of [BCP 78](#) and [BCP 79](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on July 12, 2020.

Copyright Notice

Copyright (c) 2020 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](#) and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1.	Introduction	2
2.	Requirements Language	3
3.	Terminology and Notation	3
4.	Design of Data Model	3
4.1.	Overview	3
4.2.	Flex-Algorithm Types	4
4.3.	Flex-Algorithm Data Model	4
4.3.1.	YANG Model Tree	4
4.3.2.	Notifications	7
5.	Interaction with Other YANG Modules	7
6.	Flex-Algorithm YANG Module	8
7.	Acknowledgements	16
8.	IANA Considerations	16
9.	Security Considerations	16
10.	References	16
10.1.	Normative References	16
10.2.	Informative References	18
	Authors' Addresses	18

[1.](#) Introduction

The Network Configuration Protocol (NETCONF) [[RFC6241](#)] is one of the network management protocols that defines mechanisms to manage network devices. YANG [[RFC6020](#)] is a modular language that represents data structures in an XML tree format, and is used as a data modeling language for the NETCONF.

IGP Flexible Algorithm as defined in [[I-D.ietf-lsr-flex-algo](#)], proposes a solution that allows IGP themselves to compute constraint based paths over the network. It also specifies a way of using Segment Routing Prefix-SIDs to steer packets along the constraint-based paths.

This document describes a YANG [[RFC7950](#)] data model for IGP Flexible Algorithms. The model serves as a base framework for configuring and managing Flexible Algorithm which is used by IGP to compute constraint based paths over the network.

IGP protocols like IS-IS and OSPF uses this Flexible algorithm definition in their YANG models defined in :

- o [[I-D.hu-lsr-ospf-srv6-yang](#)]
- o [[I-D.hu-isis-srv6-yang](#)]
- o [[I-D.ietf-isis-sr-yang](#)],

- o [\[I-D.ietf-ospf-sr-yang\]](#).

The model currently defines the following constructs that are used for managing Flexible Algorithm:

- o Configuration
- o Operational State
- o Notifications

2. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [\[RFC2119\]](#).

3. Terminology and Notation

This document uses the terminology defined in [\[I-D.ietf-lsr-flex-algo\]](#) . In particular, it uses the following acronyms.

- o FAD: Flexible Algorithm Definition is the set consisting of (a) calculation-type, (b) metric-type and (c) a set of constraints.
- o Flexible Algorithm (Flex-Algorithm) - a numeric identifier in the range 128-255 that is associated via provisioning with the Flexible-Algorithm Definition.
- o IGP Algorithm - value from the the "IGP Algorithm Types" registry defined under "Interior Gateway Protocol (IGP) Parameters" IANA registries. IGP Algorithms represents the triplet (Calculation Type, Metric, Constraints).
- o EAG : Extended Administrative Group: as defined in [\[RFC7308\]](#).

4. Design of Data Model

4.1. Overview

This document defines the following new YANG module:

- o ietf-flex-algo: This defines common and basic types related to Flex-Algorithm.
and specifies configuration and management model for Flex-Algorithm.

The modeling in this document complies with the Network Management Datastore Architecture (NMDA) defined in [\[RFC8342\]](#). The operational state data is combined with the associated configuration data in the same hierarchy [\[RFC8407\]](#). When protocol states are retrieved from the NMDA operational state datastore, the returned states cover all "config true" (rw) and "config false" (ro) nodes defined in the schema.

In this document, when a simplified graphical representation of YANG model is presented in a tree diagram, the meaning of the symbols in these tree diagrams is defined in [\[RFC8340\]](#).

[4.2.](#) Flex-Algorithm Types

Flex-Algorithm common types and definitions are defined in this module. The main types defined in this module include:

- o flex-algo-id: Flexible-Algorithm is single octet value between 128 and 255 both inclusive.
- o flex-algo-metric-type: Type of metric to be used during the calculation.
- o flex-algo-calc-type: Calculation type value from 0 to 127 inclusive from the "IGP Algorithm Types" registry defined under "Interior Gateway Protocol (IGP) Parameters" IANA registries.

The above fields are described in details in [\[I-D.ietf-lsr-flex-algo\]](#).

[4.3.](#) Flex-Algorithm Data Model

The Flex-Algorithm model is captured by ietf-flex-algo module. This data model specifies the configuration, operational state, and notification events required to manage Flex-Algorithm.

The associated YANG specification for this module is captured in Section Flex-Algorithm YANG Module.

[4.3.1.](#) YANG Model Tree

The module defines some fundamental items required to configure and maintain FAD.

The operational state is included in the same tree as configuration consistent with Network Management Datastore Architecture [\[RFC8342\]](#).

fad-bindings: This is used for configuring the FAD in the IGP. IGP advertises this information.

fad-tlvs: This is used to display the received information from the IGP Peers.

Following is a simplified graphical representation of the data model for Flex Algorithm configuration:

```
module: ietf-flex-algo
  augment /rt:routing:
    +--rw flex-algos
      +--rw flex-algo* [flex-algo-id]
        +--rw flex-algo-id          flex-algo-id
        +--rw metric-type?          identityref
        +--rw calc-type?            flex-algo-calc-type
        +--rw priority?             uint8
        +--rw extended-admin-groups
          | +--rw extended-admin-group* [eag-val]
          |   +--rw eag-val      te-types:extended-admin-group
        +--ro routing-protocols* []
          | +--ro routing-protocol? leafref
        +--ro flex-algo-counters
          +--ro igp-use-count? yang:counter64
  augment /rt:routing/rt:control-plane-protocols
    /rt:control-plane-protocol/isis:isis/isis:database
    /isis:levels/isis:lsp/isis:router-capabilities:
    +--ro fad-tlvs
      +--ro fad-tlv* [flex-algo-id]
        +--ro flex-algo-id          flex-algo-id
        +--ro metric-type?          identityref
        +--ro calc-type?            flex-algo-calc-type
        +--ro priority?             uint8
        +--ro extended-admin-groups
          +--ro extended-admin-group* [eag-val]
          +--ro eag-val      te-types:extended-admin-group
  augment /rt:routing/rt:control-plane-protocols
    /rt:control-plane-protocol/isis:isis:
    +--rw fad-bindings* [flex-algo-id]
      +--rw flex-algo-id
        | -> /rt:routing/flex-algos/flex-algo/flex-algo-id
      +--rw is-level?      isis:level
  augment /rt:routing/srv6:srv6/srv6:locators/srv6:locator:
    +--rw flex-algo-id?
      -> /rt:routing/flex-algos/flex-algo/flex-algo-id
  augment /rt:routing/rt:control-plane-protocols
    /rt:control-plane-protocol/ospf:ospf:
```



```
+--rw fad-bindings
  +--rw flex-algo-id*
    -> /rt:routing/flex-algos/flex-algo/flex-algo-id
augment /rt:routing/rt:control-plane-protocols
  /rt:control-plane-protocol/ospf:ospf/ospf:areas
  /ospf:area:
+--rw fad-bindings
  +--rw flex-algo-id*
    -> /rt:routing/flex-algos/flex-algo/flex-algo-id
augment /rt:routing/rt:control-plane-protocols
  /rt:control-plane-protocol/ospf:ospf/ospf:database
  /ospf:as-scope-lsa-type/ospf:as-scope-lsas
  /ospf:as-scope-lsa/ospf:version/ospf:ospfv2/ospf:ospfv2
  /ospf:body/ospf:opaque:
+--ro fad-tlvs
  +--ro fad-tlv* [flex-algo-id]
    +--ro flex-algo-id          flex-algo-id
    +--ro metric-type?         identityref
    +--ro calc-type?           flex-algo-calc-type
    +--ro priority?            uint8
    +--ro extended-admin-groups
      +--ro extended-admin-group* [eag-val]
        +--ro eag-val          te-types:extended-admin-group
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:
+--ro fad-tlvs
  +--ro fad-tlv* [flex-algo-id]
    +--ro flex-algo-id          flex-algo-id
    +--ro metric-type?         identityref
    +--ro calc-type?           flex-algo-calc-type
    +--ro priority?            uint8
    +--ro extended-admin-groups
      +--ro extended-admin-group* [eag-val]
        +--ro eag-val          te-types:extended-admin-group
augment /rt:routing/rt:control-plane-protocols
  /rt:control-plane-protocol/ospf:ospf/ospf:database
  /ospf:as-scope-lsa-type/ospf:as-scope-lsas
  /ospf:as-scope-lsa/ospf:version/ospf:ospfv3/ospf:ospfv3
  /ospf:body/ospf:router-information:
+--ro fad-tlvs
  +--ro fad-tlv* [flex-algo-id]
    +--ro flex-algo-id          flex-algo-id
    +--ro metric-type?         identityref
    +--ro calc-type?           flex-algo-calc-type
    +--ro priority?            uint8
```



```

    +--ro extended-admin-groups
      +--ro extended-admin-group* [eag-val]
        +--ro eag-val      te-types:extended-admin-group
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 fad-tlvs
  +--ro fad-tlv* [flex-algo-id]
    +--ro flex-algo-id      flex-algo-id
    +--ro metric-type?      identityref
    +--ro calc-type?        flex-algo-calc-type
    +--ro priority?         uint8
    +--ro extended-admin-groups
      +--ro extended-admin-group* [eag-val]
        +--ro eag-val      te-types:extended-admin-group

```

Figure 1: Flex-Algorithm Tree

4.3.2. Notifications

To be updated if any notification is required.

5. Interaction with Other YANG Modules

The "flex-algo" module augments the "/rt:routing/" container of the ietf-routing [[RFC8349](#)] module by defining flex-algorithm specific parameters.

The "flex-algo" module augments the "/rt:routing/rt:control-plane-protocols/rt:control-plane-protocol/isis:isis" container and "/rt:routing/rt:control-plane-protocols/rt:control-plane-protocol/ospf:ospf" for advertisement of the FAD in the IGP protocols. It also augments the specified modules for displaying the received flex-algorithm information in the database.

The "flex-algo" module also augments the "/rt:routing/srv6:srv6/srv6:locators/" for attaching the flex-algorithm to the srv6 locators.

6. Flex-Algorithm YANG Module

Following are actual YANG definition for Flexible Algorithm modules defined earlier in the document.

```
<CODE BEGINS> file "ietf-flex-algo@2019-04-26.yang" -->

module ietf-flex-algo {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-flex-algo";
  prefix flex-algo;

  import ietf-routing {
    prefix rt;
  }
  import ietf-yang-types {
    prefix yang;
  }
  import ietf-ospf {
    prefix ospf;
  }
  import ietf-isis {
    prefix isis;
  }
  import ietf-srv6-base {
    prefix srv6;
  }

  /* Import TE generic types */
  import ietf-te-types {
    prefix te-types;
  }

  organization
    "IETF LSR Working Group";
  contact
    "WG Web:   <http://tools.ietf.org/wg/lsr/>
     WG List:  <mailto:lsr@ietf.org>

     Editor:
     <mailto:mobash.rasool@gmail.com>
     <mailto:sreekanths@huawei.com>
     <mailto:mahendrasingh@huawei.com>";
  description
    "This YANG module defines the essential types for the
     management of Flex-Algorithm for IGP. It also defines the
```



```
module for configuration and management of IGP Flex-Algorithms.

Copyright (c) 2019 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 Simplified BSD License
set forth in Section 4.c of the IETF Trust's Legal Provisions
Relating to IETF Documents
(http://trustee.ietf.org/license-info).";
reference "RFC XXXX";

// RFC Editor: replace XXXX with actual RFC number and remove
// this note

revision 2019-04-26 {
  description
    "Initial revision.";
  reference "RFC XXXX";
}

/* Types definitions */

identity flex-metric-type {
  description
    "Identity from which Metric type for Flex-algorithm are
    derived";
}

identity igp {
  base flex-metric-type;
  description
    "IGP metric to be used during the calculation.";
}

identity link-delay {
  base flex-metric-type;
  description
    "Min Unidirectional Link Delay as defined in [RFC7810] to
    be used during the calculation.";
}

identity te {
  base flex-metric-type;
  description
    "TE default metric as defined in [RFC5305] to be used during
    the calculation.";
```



```
}

typedef flex-algo-id {
  type uint8 {
    range "128 .. 255";
  }
  description
    "Flex-Algorithm value is a single octet value between
    128 and 255 both inclusive.";
}

typedef flex-algo-calc-type {
  type uint8 {
    range "0 .. 127";
  }
  default "0";
  description
    "Calculation type value ranges from 0 to 127 both inclusive
    from the IGP Algorithm Types registry defined under
    Interior Gateway Protocol (IGP)
    Parameters IANA registries.
    If the required calculation type is Shortest Path First,
    the value 0 SHOULD appear in this field.";
}

/* Groupings */

grouping fad-definition {
  description
    "Flexible Algorithm Definition";
  leaf flex-algo-id {
    type flex-algo-id;
    mandatory true;
    description
      "Flex Algorithm Identifier";
  }
  leaf metric-type {
    type identityref {
      base flex-metric-type;
    }
    description
      "Indicates which is the corresponding metric type for
      this algorithm, default is IGP";
  }
  leaf calc-type {
    type flex-algo-calc-type;
    description
      "Indicates the calculation type";
  }
}
```



```
    }
    leaf priority {
      type uint8;
      description
        "Indicates the priority for this algorithm";
    }
    container extended-admin-groups {
      description
        "Extended Admin group configuration for this
        flex algorithm";
      list extended-admin-group {
        key "eag-val";
        description
          "Configure an Extended admin group";
        leaf eag-val {
          type te-types:extended-admin-group;
          description
            "The Flexible-Algorithm definition can specify
            'colors' that are used by the operator to include
            or exclude links during the Flex-Algorithm
            path computation.";
        }
      }
    }
  }
}

grouping fad-tlvs {
  description
    "Flexible Algorithm Definition TLV";
  container fad-tlvs {
    description
      "Flex Algorithms Definition TLV";
    list fad-tlv {
      key "flex-algo-id";
      description
        "List of FAD TLVs.";
      uses fad-definition;
    }
  }
}

/* Configuration and State Info */

augment "/rt:routing" {
  description
    "This augments routing-instance configuration with
    Flexible Algorithm.";
  container flex-algos {
```



```
    description
      "Flex Algorithms";
  list flex-algo {
    key "flex-algo-id";
    description
      "Configure a Flex Algorithm";
    uses fad-definition;
    list routing-protocols {
      config false;
      description
        "Details of Routing Protocols using this Flexible
        algorithm";
      leaf routing-protocol {
        type leafref {
          path "/rt:routing/rt:control-plane-protocols"
            + "/rt:control-plane-protocol/rt:name";
        }
        config false;
        description
          "Indicates which routing protocol used this flex
          algorithm";
      }
    }
  }
  container flex-algo-counters {
    config false;
    description
      "Flex Algorithm Counters";
    leaf igp-use-count {
      type yang:counter64;
      config false;
      description
        "This counter helps to find the number of IGP instances
        using this FAD";
    }
  }
}

/* Configuration and state information for ISIS */
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 ISIS routing protocol when used";
```



```
    }
    description
      "This augments ISIS protocol LSDB router capability.";
    uses fad-tlvs;
  }

augment "/rt:routing/" +
  "rt:control-plane-protocols/rt:control-plane-protocol"+
  "/isis:isis" {
  description
    "This augments isis configuration.";
  list fad-bindings {
    key "flex-algo-id";
    description
      "Flex Algorithm Counters";
    leaf flex-algo-id {
      type leafref {
        path "/rt:routing/flex-algos/flex-algo/flex-algo-id";
      }
      description
        "Flex Algorithm Identifier binded to ISIS";
    }
    leaf is-level {
      type isis:level;
      description
        "ISIS Level associated with this Flex Algorithm";
    }
  }
}

/*SRV6 Locator config and state */

augment "/rt:routing/srv6:srv6/srv6:locators/srv6:locator" {
  description
    "This augments locator configuration in segment-routing
    SRv6.";
  leaf flex-algo-id {
    type leafref {
      path "/rt:routing/flex-algos/flex-algo/flex-algo-id";
    }
    description
      "Flex Algorithm Identifier that is used by this locator";
  }
}

/* OSPF configuration */

augment "/rt:routing/rt:control-plane-protocols"
```



```
        + "/rt:control-plane-protocol/ospf:ospf" {
description
  "Flex Algorithm is binded to OSPF. OSPF then
    advertises FAD TLVs in RI LSA in AS-scope";
container fad-bindings {
  description
    "Flex Algorithm is binded to OSPF.";
  leaf-list flex-algo-id {
    type leafref {
      path "/rt:routing/flex-algos/flex-algo/flex-algo-id";
    }
    description
      "Flex Algorithm that the router will use and advertise.";
  }
}
}

augment "/rt:routing/rt:control-plane-protocols"
  + "/rt:control-plane-protocol/ospf:ospf"
  + "/ospf:areas/ospf:area" {
description
  "Flex Algorithm is binded to OSPF area. OSPF then
    advertises FAD TLVs in RI LSA in Area-scope";
container fad-bindings {
  description
    "Flex Algorithm is binded to OSPF.";
  leaf-list flex-algo-id {
    type leafref {
      path "/rt:routing/flex-algos/flex-algo/flex-algo-id";
    }
    description
      "Flex Algorithm that the router will use and advertise.";
  }
}
}

/* State information for OSPFv2*/
augment "/rt:routing/"
  + "rt:control-plane-protocols/rt:control-plane-protocol/"
  + "ospf:ospf/ospf:database/"
  + "ospf:as-scope-lsa-type/ospf:as-scope-lsas/"
  + "ospf:as-scope-lsa/ospf:version/ospf:ospfv2/"
  + "ospf:ospfv2/ospf:body/ospf:opaque" {
when "../.../.../.../.../.../.../..."
  + "rt:type = 'ospf:ospfv2'" {
description
  "This augmentation is only valid for OSPFv2.";
}
```



```
description
  "SR specific TLVs for OSPFv2 type 11 opaque LSA.";
uses fad-tlvs;
}

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" {
when "../../../../../../../../../../../"
+ "rt:type = 'ospf:ospfv2'" {
description
  "This augmentation is only valid for OSPFv2.";
}
description
  "FAD TLVs for OSPFv2 type 10 opaque LSA.";
uses fad-tlvs;
}

/* State information for OSPFv3 */
augment "/rt:routing/"
+ "rt:control-plane-protocols/rt:control-plane-protocol/"
+ "ospf:ospf/ospf:database/ospf:as-scope-lsa-type/"
+ "ospf:as-scope-lsas/ospf:as-scope-lsa/"
+ "ospf:version/ospf:ospfv3/ospf:ospfv3/ospf:body/"
+ "ospf:router-information" {
description "This is augmented only for OSPFv3";
uses fad-tlvs;
}

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 "/rt:routing/rt:control-plane-protocols/" +
  "rt:control-plane-protocol/rt:type = 'ospf:ospfv3'" {
description
  "This augment OSPFv3 routing protocol when used";
}
description
  "This displays FAD TLVs in router capability";
uses fad-tlvs;
```



```
}  
  
// RFC Editor: replace XXXX with actual RFC number and remove  
// this note  
}  
<CODE ENDS>
```

7. Acknowledgements

The authors wish to thank Dhruv Dhody and Rohit Ranade for their helpful comments and suggestions.

8. IANA Considerations

None

9. Security Considerations

The configuration, state, and notification data defined using YANG data models in this document are likely to be accessed via the protocols such as NETCONF [[RFC6241](#)] etc.

Hence, YANG implementations MUST comply with the security requirements specified in [section 15 of \[RFC6020\]](#). Additionally, NETCONF implementations MUST comply with the security requirements specified in sections [2.2](#), [2.3](#) and [9](#) of [[RFC6241](#)] as well as [section 3.7 of \[RFC8341\]](#).

The security considerations are same as defined in [[I-D.ietf-lsr-flex-algo](#)]

10. References

10.1. Normative References

[I-D.hu-isis-srv6-yang]

Hu, Z., Ye, D., Qu, Y., and J. Dong, "YANG Data Model for IS-IS SRv6", [draft-hu-isis-srv6-yang-02](#) (work in progress), November 2019.

[I-D.hu-lsr-ospf-srv6-yang]

Hu, Z., Raza, K., Qu, Y., and J. Dong, "YANG Data Model for OSPF SRv6", [draft-hu-lsr-ospf-srv6-yang-01](#) (work in progress), November 2019.

[I-D.ietf-isis-sr-yang]

Litkowski, S., Qu, Y., Sarkar, P., Chen, I., and J. Tantsura, "YANG Data Model for IS-IS Segment Routing", [draft-ietf-isis-sr-yang-07](#) (work in progress), January 2020.

[I-D.ietf-ospf-sr-yang]

Yeung, D., Qu, Y., Zhang, Z., Chen, I., and A. Lindem, "YANG Data Model for OSPF SR (Segment Routing) Protocol", [draft-ietf-ospf-sr-yang-10](#) (work in progress), August 2019.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.

[RFC6020] Bjorklund, M., Ed., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", [RFC 6020](#), DOI 10.17487/RFC6020, October 2010, <<https://www.rfc-editor.org/info/rfc6020>>.

[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>>.

[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>>.

[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>>.

[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>>.

[RFC8341] Bierman, A. and M. Bjorklund, "Network Configuration Access Control Model", STD 91, [RFC 8341](#), DOI 10.17487/RFC8341, March 2018, <<https://www.rfc-editor.org/info/rfc8341>>.

- [RFC8342] Bjorklund, M., Schoenwaelder, J., Shafer, P., Watsen, K., and R. Wilton, "Network Management Datastore Architecture (NMDA)", [RFC 8342](#), DOI 10.17487/RFC8342, March 2018, <<https://www.rfc-editor.org/info/rfc8342>>.
- [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>>.
- [RFC8407] Bierman, A., "Guidelines for Authors and Reviewers of Documents Containing YANG Data Models", [BCP 216](#), [RFC 8407](#), DOI 10.17487/RFC8407, October 2018, <<https://www.rfc-editor.org/info/rfc8407>>.

10.2. Informative References

- [I-D.ietf-lsr-flex-algo]
Psenak, P., Hegde, S., Filsfils, C., Talaulikar, K., and A. Gulko, "IGP Flexible Algorithm", [draft-ietf-lsr-flex-algo-05](#) (work in progress), November 2019.

Authors' Addresses

Mobashshera Rasool (editor)
Huawei Technologies
Divyashree Techno Park, Whitefield
Bengaluru, Karnataka 560066
India

Email: mobash.rasool@gmail.com

Sreekanth S (editor)
Huawei Technologies
Divyashree Techno Park, Whitefield
Bengaluru, Karnataka 560066
India

Email: sreekanths@huawei.com

Mahendra Singh Negi (editor)
Huawei Technologies
Divyashree Techno Park, Whitefield
Bengaluru, Karnataka 560066
India

Email: mahendrasingh@huawei.com