TEAS Working Group
Internet-Draft
Intended status: Standards Track
Expires: 23 April 2026

V.P. Beeram
Juniper Networks
T. Saad
R. Gandhi
Cisco Systems
X. Liu
Alef Edge
20 October 2025

YANG Data Model for Topology Filter
draft-ietf-teas-yang-topology-filter-02

Abstract

This document defines a YANG data model for the management of topology filters/filter-sets on network elements and controllers.

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.

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 23 April 2026.

Copyright Notice

Copyright (c) 2025 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 Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

Table of Contents

1. Introduction

A topology filter is a data construct that is used to filter network topologies [RFC8345]. It can be applied on either a native topology or a customized topology [RFC8795] to produce a filtered set of topological elements. A topology filter-set is a union of multiple topology filters that can be applied in tandem on a topology. This document defines a YANG data model for the management of topology filters/filter-sets on network elements and controllers.

The authors acknowledge that an implementation may maintain network topologies that are learnt via routing protocols in a Routing Information Base (RIB) [RFC8431] and use routing policies [RFC9067] to filter the entries in the RIB. Such an implementation is not the target of this document.

1.1. Use-Cases

* Specification of topology related constraints for TE Path Computation: A few examples of this are -
  - Compute a path within a specified topology.
  - Compute a path within the topology associated with a specific IGP domain.
  - Compute a path within the topology learnt from a specific TE Information Source.
  - Compute a path within the topology defined by the application of one or more topology filters:
    o Use a topology with elements learnt via ISIS Level-2 and include resource-affinity "RED"
    o Use a topology with elements associated with ISIS Flexible Algorithm 128 and exclude resource-affinity "BLUE"
* Specification of topology associated with an Network Resource Partition (NRP): A few examples of rules for determining the topology associated with the NRP [RFC9543] are:
  - All the elements in the specified topology are part of the NRP topology.
  - All the topological elements associated with a specific IGP domain are part of the NRP topology.
  - All the topological elements that include resource-affinity "RED" and exclude resource-affinity "BLUE" are part of the NRP topology.

1.2. Terminology

The terminology for describing YANG data models is found in [RFC7950].

The reader is expected to be familiar with the topology modeling terminology specified in [RFC8345], [RFC8776] and [RFC8795].

1.3. Tree Structure

A simplified graphical representation of the data model is presented in Appendix A of this document. The tree format defined in [RFC8340] is used for the YANG data model tree representation.

2. Topology Filter Data Model

2.1. Model Structure

The high-level model structure defined by this document is as shown below:

```
module: ietf-topology-filter
  augment /nw:networks:
    +--rw topology-filters!
    |  +--rw topology-filter* [name]
    |     +--rw name            string
    |     +--rw topology-ref
    |     |  ..........
    |     +--rw include-any
    |     |  ..........
    |     +--rw include-all
    |     |  ..........
    |     +--rw exclude
    |        ..........
    +--rw topology-filter-sets!
       +--rw topology-filter-set* [name]
          +--rw name               string
          +  ..........
```

The top-level 'networks' container [RFC8345] is augmented with a set of topology filters and a set of topology filter-sets.

2.1.1. Topology Filters

The 'topology-filters' container carries a list of topology filters. Each topology-filter entry specifies a set of include-any, include-all and exclude filtering rules that can be applied on either the native topology or a user specified topology.

2.1.1.1. Topology Reference

The 'topology-reference' container indicates the topology on which the filtering rules need to be applied. The referenced topology could be a predefined TE topology and/or a specific IGP domain. The absence of the 'topology-reference' indicates that the filtering rules are to be applied on the native topology.

```
            +--rw topology-ref
               +--rw igp-domain-identifier
               |  +--rw protocol-id?   igp-protocol
               |  +--rw instance-id?   uint32
               |  +--rw division-id?   uint32
               |  +--rw algo-id?       uint8
               |  +--rw mt-id?         uint16
               +--rw te-topology-identifier
                  +--rw provider-id?   te-global-id
                  +--rw client-id?     te-global-id
                  +--rw topology-id?   te-topology-id
```

2.1.1.2. Filters

The 'include-any', 'include-all' and 'exclude' containers carry a varied set of attributes that can be used as rules to filter the topology. If the topology-filter entry carries no filtering rules and only references a specific topology, then the set of filtered topological elements produced is the same as the one defined by the referenced topology.

```
            +--rw include-any
            |  +--rw link-affinity*   string
            |  +--rw link-name*       string
            |  +--rw node-prefix*     inet:ip-prefix
            |  +--rw as*              inet:as-number
            |  +--rw info-source* [source-id instance-id division-id]
            |     +--rw source-id      tet:te-info-source
            |     +--rw instance-id    uint32
            |     +--rw division-id    uint32
            +--rw include-all
            |  +--rw link-affinity*   string
            |  +--rw link-name*       string
            |  +--rw node-prefix*     inet:ip-prefix
            |  +--rw as*              inet:as-number
            |  +--rw info-source* [source-id instance-id division-id]
            |     +--rw source-id      tet:te-info-source
            |     +--rw instance-id    uint32
            |     +--rw division-id    uint32
            +--rw exclude
               +--rw link-affinity*   string
               +--rw link-name*       string
               +--rw node-prefix*     inet:ip-prefix
               +--rw as*              inet:as-number
               +--rw info-source* [source-id instance-id division-id]
                  +--rw source-id      tet:te-info-source
                  +--rw instance-id    uint32
                  +--rw division-id    uint32
```

2.1.2. Topology Filter-Sets

The 'topology-filter-sets' container carries a list of topology filter-sets. Each topology-filter-set entry constitutes a list of topology-filter references. This is used when there is a need to create a union of multiple topology filters.

```
    +--rw topology-filter-sets!
       +--rw topology-filter-set* [name]
          +--rw name                string
          +--rw topology-filter*
                  -> ../../../topology-filters/topology-filter/name
```

2.2. YANG Module

```
<CODE BEGINS> file "ietf-topology-filter@2025-10-19.yang"
module ietf-topology-filter {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-topology-filter";
  prefix topo-filt;

  import ietf-inet-types {
    prefix inet;
    reference
      "RFC 6991: Common YANG Data Types";
  }
  import ietf-network {
    prefix nw;
    reference
      "RFC 8345: A YANG Data Model for Network Topologies";
  }
  import ietf-te-types {
    prefix te-types;
    reference
      "RFC 8776: Common YANG Data Types for Traffic Engineering";
  }
  import ietf-te-topology {
    prefix tet;
    reference
      "RFC 8795: YANG Data Model for Traffic Engineering Topologies";
  }

  organization
    "IETF Traffic Engineering Architecture and Signaling (TEAS)
     Working Group.";
  contact
    "WG Web:   <http://tools.ietf.org/wg/teas/>
     WG List:  <mailto:teas@ietf.org>
```

```
        Editor:   Vishnu Pavan Beeram
                  <mailto:vbeeram@juniper.net>

        Editor:   Tarek Saad
                  <mailto:tsaad.net@gmail.com>

        Editor:   Rakesh Gandhi
                  <mailto:rgandhi@cisco.com>

        Editor:   Xufeng Liu
                  <mailto:xufeng.liu.ietf@gmail.com>";
    description
      "This YANG module defines data definitions for managing
       topology filters.

       Copyright (c) 2025 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).

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

    revision 2025-10-19 {
      description
        "Initial revision.";
      reference
        "RFC XXXX: YANG Data Model for Topology Filters.";
    }

    /*
     * T Y P E D E F S
     */

    typedef igp-protocol {
      type enumeration {
        enum ospfv2 {
          description
            "OSPFv2.";
        }
        enum ospfv3 {
          description
            "OSPFv3.";
```

```
        }
        enum isis {
          description
            "IS-IS.";
        }
      }
      description
        "IGP Protocol Type.";
    }

    /*
     * G R O U P I N G S
     */
    /*
     * Grouping - Topology Information Source.
     */

    grouping igp-topology-info-source {
      description
        "Grouping for igp topology information source.";
      leaf protocol-id {
        type igp-protocol;
        description
          "IGP Protocol Type.";
      }
      leaf instance-id {
        type uint32;
        description
          "Information Source Instance.";
      }
      leaf division-id {
        type uint32;
        description
          "Information Source Division
           (e.g., OSPF Area, ISIS Level).";
      }
    }

    /*
     * Grouping - IGP Domain Identifier.
     */

    grouping igp-domain-identifier {
      description
        "Grouping for igp domain identifier.";
      container igp-domain-identifier {
        description
          "Container for igp domain identifier.";
```

```
        uses igp-topology-info-source;
        leaf algo-id {
          type uint8;
          description
            "Algorithm ID.";
        }
        leaf mt-id {
          type uint16;
          description
            "Multi Topology ID.";
        }
      }
    }

    /*
     * Grouping - Topology Reference
     */

    grouping topology-reference {
      description
        "Grouping for topology reference.";
      container topology-ref {
        description
          "Container for topology reference.";
        uses igp-domain-identifier;
        uses te-types:te-topology-identifier;
      }
    }

    /*
     * Grouping - Topology Information Sources
     */

    grouping topology-info-sources {
      description
        "Grouping for topology information sources.";
      list info-source {
        key "source-id instance-id division-id";
        description
          "List of information-sources.";
        leaf source-id {
          type tet:te-info-source;
          description
            "Information Source.";
        }
        leaf instance-id {
          type uint32;
          description
```

```
            "Information Source Instance.";
        }
        leaf division-id {
          type uint32;
          description
            "Information Source Division.";
        }
      }
    }

    /*
     * Grouping - Custom Topology Filters
     */

    grouping custom-topology-filters {
      description
        "Grouping for custom topology filters.";
      leaf-list link-affinity {
        type string;
        description
          "List of link affinities.";
      }
      leaf-list link-name {
        type string;
        description
          "List of link names.";
      }
      leaf-list node-prefix {
        type inet:ip-prefix;
        description
          "List of node IDs.";
      }
      leaf-list as {
        type inet:as-number;
        description
          "List of AS numbers.";
      }
      uses topology-info-sources;
    }

    /*
     * Grouping - Topology Filters
     */

    grouping topology-filters {
      description
        "Grouping for topology filters.";
      container topology-filters {
```

```
        presence "Enable Topology Filters.";
        description
          "Container for topology filters.";
        list topology-filter {
          key "name";
          description
            "List of topology filters.";
          leaf name {
            type string;
            description
              "A string that uniquely identifies the topology filter.";
          }
          uses topology-reference;
          container include-any {
            description
              "Include-any filters.";
            uses custom-topology-filters;
          }
          container include-all {
            description
              "Include-all filters.";
            uses custom-topology-filters;
          }
          container exclude {
            description
              "Exclude filters.";
            uses custom-topology-filters;
          }
        }
      }
    }

    /*
     * Grouping - Topology Filter Sets
     */

    grouping topology-filter-sets {
      description
        "Grouping for topology filter sets.";
      container topology-filter-sets {
        presence "Enable Topology Filter-Sets.";
        description
          "Container for topology filter sets.";
        list topology-filter-set {
          key "name";
          description
            "List of topology filter sets.";
          leaf name {
```

```
            type string;
            description
              "A string that uniquely identifies the topology
               filter-set.";
          }
          leaf-list topology-filter {
            type leafref {
              path "../../../topo-filt:topology-filters/"
                 + "topo-filt:topology-filter/topo-filt:name";
            }
            description
              "Reference to a specific topology filter from the list
               of topology filters.";
          }
        }
      }
    }

    /*
     *  Augment - Topology Filters / Topology Filter-Sets
     */

    augment "/nw:networks" {
      description
        "Augment networks with topology-filters and
         topology-filter-sets.";
      uses topology-filters;
      uses topology-filter-sets;
    }
  }
  <CODE ENDS>
```

3. Acknowledgements

The authors would like to thank Sudharsana Venkatraman for her input from discussions.

4. Contributors

The following individuals contributed to this document:

Colby Barth
Juniper Networks
Email: cbarth@juniper.net

Srihari R. Sangli
Juniper Networks
Email: ssangli@juniper.net

Chandra Ramachandran
Juniper Networks
Email: csekar@juniper.net

5. IANA Considerations

This document registers the following URI in the IETF XML registry [RFC3688]. Following the format in [RFC3688], the following registration is requested to be made.

URI: urn:ietf:params:xml:ns:yang:ietf-topology-filter
Registrant Contact: The TEAS WG of the IETF.
XML: N/A, the requested URI is an XML namespace.

This document registers a YANG module in the YANG Module Names registry [RFC6020].

name: ietf-topology-filter
namespace: urn:ietf:params:xml:ns:yang:ietf-topology-filter
prefix: topo-filt
reference: RFCXXXX

6. Security Considerations

The YANG module specified in this document defines a schema for data that is designed to be accessed via network management protocols such as NETCONF [RFC6241] or RESTCONF [RFC8040]. The lowest NETCONF layer is the secure transport layer, and the mandatory-to-implement secure transport is Secure Shell (SSH) [RFC6242]. The lowest RESTCONF layer is HTTPS, and the mandatory-to-implement secure transport is TLS [RFC8446].

The Network Configuration Access Control Model (NACM) [RFC8341] provides the means to restrict access for particular NETCONF or RESTCONF users to a preconfigured subset of all available NETCONF or RESTCONF protocol operations and content.

The data nodes defined in this YANG module that are writable/creatable/deletable (i.e., config true, which is the default) may be considered sensitive or vulnerable in some network environments. Write operations (e.g., edit-config) to these data nodes without proper protection can have a negative effect on network operations. These are the subtrees and data nodes and their sensitivity/vulnerability:

* "/networks/topology-filters/": This subtree specifies the

configurations for topology filters. By manipulating these data nodes, a malicious attacker may cause unauthorized and improper behavior to any service that is making use of the filtered set of topological elements produced by the application of the compromised topology filter.

* "/networks/topology-filter-sets": This subtree specifies the configurations for topology filter-sets. By manipulating these data nodes, a malicious attacker may cause unauthorized and improper behavior to any service that is making use of the filtered set of topological elements produced by the application of the compromised topology filter-set.

The readable data nodes in this YANG module may be considered sensitive or vulnerable in some network environments. It is thus important to control read access (e.g., via get, get-config, or notification) to these data nodes. These are the subtrees and data nodes and their sensitivity/vulnerability:

* "/networks/topology-filter": Unauthorized access to this subtree can disclose the topology filters used in the network.

* "/networks/topology-filter-sets": Unauthorized access to this subtree can disclose the topology filter-sets used in the network.

7. References

7.1. Normative References

[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>.

[RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, RFC 3688, DOI 10.17487/RFC3688, January 2004, <https://www.rfc-editor.org/info/rfc3688>.

[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>.

[RFC6242] Wasserman, M., "Using the NETCONF Protocol over Secure Shell (SSH)", RFC 6242, DOI 10.17487/RFC6242, June 2011, <https://www.rfc-editor.org/info/rfc6242>.

[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>.

[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>.

[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>.

[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>.

[RFC8345] Clemm, A., Medved, J., Varga, R., Bahadur, N., Ananthakrishnan, H., and X. Liu, "A YANG Data Model for Network Topologies", RFC 8345, DOI 10.17487/RFC8345, March 2018, <https://www.rfc-editor.org/info/rfc8345>.

[RFC8446] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", RFC 8446, DOI 10.17487/RFC8446, August 2018, <https://www.rfc-editor.org/info/rfc8446>.

[RFC8776] Saad, T., Gandhi, R., Liu, X., Beeram, V., and I. Bryskin, "Common YANG Data Types for Traffic Engineering", RFC 8776, DOI 10.17487/RFC8776, June 2020, <https://www.rfc-editor.org/info/rfc8776>.

[RFC8795] Liu, X., Bryskin, I., Beeram, V., Saad, T., Shah, H., and O. Gonzalez de Dios, "YANG Data Model for Traffic Engineering (TE) Topologies", RFC 8795, DOI 10.17487/RFC8795, August 2020, <https://www.rfc-editor.org/info/rfc8795>.

7.2. Informative References

[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>.

[RFC8431] Wang, L., Chen, M., Dass, A., Ananthakrishnan, H., Kini, S., and N. Bahadur, "A YANG Data Model for the Routing Information Base (RIB)", RFC 8431, DOI 10.17487/RFC8431, September 2018, <https://www.rfc-editor.org/info/rfc8431>.

[RFC9067] Qu, Y., Tantsura, J., Lindem, A., and X. Liu, "A YANG Data Model for Routing Policy", RFC 9067, DOI 10.17487/RFC9067, October 2021, <https://www.rfc-editor.org/info/rfc9067>.

[RFC9543] Farrel, A., Ed., Drake, J., Ed., Rokui, R., Homma, S., Makhijani, K., Contreras, L., and J. Tantsura, "A Framework for Network Slices in Networks Built from IETF Technologies", RFC 9543, DOI 10.17487/RFC9543, March 2024, <https://www.rfc-editor.org/info/rfc9543>.

Appendix A. Complete Model Tree Structure

```
module: ietf-topology-filter

  augment /nw:networks:
    +--rw topology-filters!
    |  +--rw topology-filter* [name]
    |     +--rw name             string
    |     +--rw topology-ref
    |     |  +--rw igp-domain-identifier
    |     |  |  +--rw protocol-id?   igp-protocol
    |     |  |  +--rw instance-id?   uint32
    |     |  |  +--rw division-id?   uint32
    |     |  |  +--rw algo-id?       uint8
    |     |  |  +--rw mt-id?         uint16
    |     |  +--rw te-topology-identifier
    |     |     +--rw provider-id?   te-global-id
    |     |     +--rw client-id?     te-global-id
    |     |     +--rw topology-id?   te-topology-id
    |     +--rw include-any
    |     |  +--rw link-affinity*   string
    |     |  +--rw link-name*       string
    |     |  +--rw node-prefix*     inet:ip-prefix
    |     |  +--rw as*              inet:as-number
    |     |  +--rw info-source* [source-id instance-id division-id]
    |     |     +--rw source-id      tet:te-info-source
    |     |     +--rw instance-id    uint32
    |     |     +--rw division-id    uint32
    |     +--rw include-all
    |     |  +--rw link-affinity*   string
    |     |  +--rw link-name*       string
    |     |  +--rw node-prefix*     inet:ip-prefix
    |     |  +--rw as*              inet:as-number
```

```
       |     |  +--rw info-source* [source-id instance-id division-id]
       |     |     +--rw source-id      tet:te-info-source
       |     |     +--rw instance-id    uint32
       |     |     +--rw division-id    uint32
       |     +--rw exclude
       |        +--rw link-affinity*   string
       |        +--rw link-name*       string
       |        +--rw node-prefix*     inet:ip-prefix
       |        +--rw as*              inet:as-number
       |        +--rw info-source* [source-id instance-id division-id]
       |           +--rw source-id      tet:te-info-source
       |           +--rw instance-id    uint32
       |           +--rw division-id    uint32
       +--rw topology-filter-sets!
          +--rw topology-filter-set* [name]
             +--rw name               string
             +--rw topology-filter*
                    -> ../../../topology-filters/topology-filter/name
```

Authors' Addresses

Vishnu Pavan Beeram
Juniper Networks
Email: vbeeram@juniper.net

Tarek Saad
Cisco Systems
Email: tsaad.net@gmail.com

Rakesh Gandhi
Cisco Systems
Email: rgandhi@cisco.com

Xufeng Liu
Alef Edge
Email: xufeng.liu.ietf@gmail.com