A YANG Data Model for MPLS Traffic Engineering Tunnels
draft-ietf-teas-yang-te-mpls-01

Abstract

This document defines a YANG data model for the configuration and management of Multiprotocol Label Switching (MPLS) Traffic Engineering (TE) tunnels, Label Switched Paths (LSPs) and interfaces. The model augments the TE generic YANG model for MPLS packet dataplane technology.

This model covers data for configuration, operational state, remote procedural calls, and event notifications.

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 August 27, 2019.

Copyright Notice

Copyright (c) 2019 IETF Trust and the persons identified as the document authors. All rights reserved.
This document is subject to BCP 78 and the IETF Trust’s Legal Provisions Relating to IETF Documents (https://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction ..................................................... 2
   1.1. Terminology ................................................. 2
   1.2. Prefixes in Data Node Names ............................... 3
   1.3. Acronyms and Abbreviations ............................... 3
2. MPLS TE YANG Model .............................................. 3
   2.1. Module(s) Relationship ................................... 3
   2.2. Model Tree Diagram ....................................... 4
   2.3. MPLS TE YANG Module ..................................... 7
3. IANA Considerations ........................................... 17
4. Security Considerations ....................................... 18
5. Contributors .................................................. 18
6. Normative References ......................................... 18
Authors’ Addresses ............................................... 20

1. Introduction

YANG [RFC6020] and [RFC7950] is a data modeling language used to define the contents of a conceptual data store that allows networked devices to be managed using NETCONF [RFC6241]. YANG has proved relevant beyond its initial confines, as bindings to other interfaces (e.g. RESTCONF [RFC8040]) and encoding other than XML (e.g. JSON) are being defined. Furthermore, YANG data models can be used as the basis of implementation for other interfaces, such as CLI and programmatic APIs.

This document describes the YANG data model for configuration and management of MPLS TE tunnels, LSPs, and interfaces. Other YANG module(s) that model the establishment of MPLS LSP(s) via signaling protocols such as RSVP-TE ([RFC3209], [RFC3473]) are described in separate document(s).

1.1. Terminology

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...
1.2. Prefixes in Data Node Names

In this document, names of data nodes and other data model objects are prefixed using the standard prefix associated with the corresponding YANG imported modules, as shown in Table 1.

<table>
<thead>
<tr>
<th>Prefix</th>
<th>YANG module</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>yang</td>
<td>ietf-yang-types</td>
<td>[RFC6991]</td>
</tr>
<tr>
<td>inet</td>
<td>ietf-inet-types</td>
<td>[RFC6991]</td>
</tr>
<tr>
<td>rt-types</td>
<td>ietf-routing-types</td>
<td>[RFC8294]</td>
</tr>
<tr>
<td>te</td>
<td>ietf-te</td>
<td>[I-D.ietf-teas-yang-te]</td>
</tr>
<tr>
<td>te-dev</td>
<td>ietf-te-device</td>
<td>[I-D.ietf-teas-yang-te]</td>
</tr>
<tr>
<td>te-mpls</td>
<td>ietf-te-mpls</td>
<td>This document</td>
</tr>
<tr>
<td>te-types</td>
<td>ietf-te-types</td>
<td>[I-D.ietf-teas-yang-te-types]</td>
</tr>
<tr>
<td>te-mpls-types</td>
<td>ietf-te-mpls-types</td>
<td>[I-D.ietf-teas-yang-te-types]</td>
</tr>
</tbody>
</table>

Table 1: Prefixes and corresponding YANG modules

1.3. Acronyms and Abbreviations

MPLS: Multiprotocol Label Switching LSP: Label Switched Path LSR: Label Switching Router LER: Label Edge Router TE: Traffic Engineering

2. MPLS TE YANG Model

The MPLS TE YANG model covers the configuration, state, RPC and notifications data pertaining to MPLS TE interfaces, tunnels and LSPs parameters. The data specific to the signaling protocol used to establish MPLS LSP(s) is outside the scope of this document and is covered in other documents, e.g. in [I-D.ietf-teas-yang-rsvp] and [I-D.ietf-teas-yang-rsvp-te].

2.1. Module(s) Relationship

The MPLS TE YANG module "ietf-te-mpls" imports the following modules:

- ietf-te and ietf-te-device defined in [I-D.ietf-teas-yang-te]
o ietf-te-types and ietf-te-packet-types defined in [I-D.ietf-teas-yang-te-types]

o ietf-routing-types defined in [RFC8294]

o ietf-mpls-static defined in [I-D.ietf-mpls-static-yang]

The MPLS TE YANG module "ietf-te-mpls" augments the "ietf-te" TE generic YANG module as shown in Figure 1.

2.2. Model Tree Diagram

Figure 2 shows the tree diagram of the MPLS TE YANG model that is defined in ietf-te-mpls.yang.

```yang
mODULE ietf-te-mpls
  AUGMENT /te:te/te-dev:performance-thresholds:
    ---RW throttle
      ---RW one-way-delay-offset? uint32
      ---RW measure-interval? uint32
      ---RW advertisement-interval? uint32
      ---RW suppression-interval? uint32
      ---RW threshold-out
        ---RW one-way-delay? uint32
        ---RW one-way-residual-bandwidth?
          rt-types:bandwidth-ieee-float32
        ---RW one-way-available-bandwidth?
          rt-types:bandwidth-ieee-float32
        ---RW one-way-utilized-bandwidth?
          rt-types:bandwidth-ieee-float32
        ---RW two-way-delay? uint32
        ---RW one-way-min-delay? uint32
        ---RW one-way-max-delay? uint32
```
Augment /te:te/te:tunnels/te:tunnel:
  +--rw tunnel-igp-shortcut
        |     +--rw shortcut-eligible?    boolean
        |     +--rw metric-type?        identityref
        |     +--rw metric?            int32
        |     +--rw routing-afs*       inet:ip-version
  +--rw forwarding
        |     +--rw binding-label?     rt-types:mpls-label

++--rw threshold-in
    +--rw one-way-delay?     uint32
    +--rw one-way-residual-bandwidth?
        rt-types:bandwidth-ieee-float32
    +--rw one-way-available-bandwidth?
        rt-types:bandwidth-ieee-float32
    +--rw one-way-used-bandwidth?
        rt-types:bandwidth-ieee-float32
    +--rw two-way-delay?      uint32
    +--rw one-way-min-delay?  uint32
    +--rw one-way-max-delay?  uint32
    +--rw one-way-delay-variation?  uint32
    +--rw one-way-packet-loss? decimal64
    +--rw two-way-min-delay?  uint32
    +--rw two-way-max-delay?  uint32
    +--rw two-way-delay-variation?  uint32
    +--rw two-way-packet-loss? decimal64
++--rw threshold-accelerated-advertisement
    +--rw one-way-delay?     uint32
    +--rw one-way-residual-bandwidth?
        rt-types:bandwidth-ieee-float32
    +--rw one-way-available-bandwidth?
        rt-types:bandwidth-ieee-float32
    +--rw one-way-used-bandwidth?
        rt-types:bandwidth-ieee-float32
    +--rw two-way-delay?      uint32
    +--rw one-way-min-delay?  uint32
    +--rw one-way-max-delay?  uint32
    +--rw one-way-delay-variation?  uint32
    +--rw one-way-packet-loss? decimal64
    +--rw two-way-min-delay?  uint32
    +--rw two-way-max-delay?  uint32
    +--rw two-way-delay-variation?  uint32
    +--rw two-way-packet-loss? decimal64

++--rw one-way-delay-variation?     uint32
++--rw one-way-packet-loss?           decimal64
++--rw two-way-min-delay?             uint32
++--rw two-way-max-delay?             uint32
++--rw two-way-delay-variation?       uint32
++--rw two-way-packet-loss?           decimal64
++--rw threshold-in
    +--rw one-way-delay?     uint32
    +--rw one-way-residual-bandwidth?
        rt-types:bandwidth-ieee-float32
    +--rw one-way-available-bandwidth?
        rt-types:bandwidth-ieee-float32
    +--rw one-way-used-bandwidth?
        rt-types:bandwidth-ieee-float32
    +--rw two-way-delay?      uint32
    +--rw one-way-min-delay?  uint32
    +--rw one-way-max-delay?  uint32
    +--rw one-way-delay-variation?  uint32
    +--rw one-way-packet-loss? decimal64
    +--rw two-way-min-delay?  uint32
    +--rw two-way-max-delay?  uint32
    +--rw two-way-delay-variation?  uint32
    +--rw two-way-packet-loss? decimal64
++--rw threshold-accelerated-advertisement
    +--rw one-way-delay?     uint32
    +--rw one-way-residual-bandwidth?
        rt-types:bandwidth-ieee-float32
    +--rw one-way-available-bandwidth?
        rt-types:bandwidth-ieee-float32
    +--rw one-way-used-bandwidth?
        rt-types:bandwidth-ieee-float32
    +--rw two-way-delay?      uint32
    +--rw one-way-min-delay?  uint32
    +--rw one-way-max-delay?  uint32
    +--rw one-way-delay-variation?  uint32
    +--rw one-way-packet-loss? decimal64
    +--rw two-way-min-delay?  uint32
    +--rw two-way-max-delay?  uint32
    +--rw two-way-delay-variation?  uint32
    +--rw two-way-packet-loss? decimal64
| ---rw load-share?    uint32
| ---rw policy-class? uint8

++-rw bandwidth-mpls
++-rw specification-type?
  | te-packet-types:te-bandwidth-requested-type
++-rw set-bandwidth?  te-packet-types:bandwidth-kbps
++-rw class-type?     te-types:te-ds-class
++-ro state
  | ++-ro signaled-bandwidth?  te-packet-types:bandwidth-kbps
++-rw auto-bandwidth
  | ++-rw enabled?         boolean
  | ++-rw min-bw?          te-packet-types:bandwidth-kbps
  | ++-rw max-bw?          te-packet-types:bandwidth-kbps
  | ++-rw adjust-interval? uint32
  | ++-rw adjust-threshold? rt-types:percentage
  | ++-rw overflow
    | | ++-rw enabled?         boolean
    | | ++-rw overflow-threshold? rt-types:percentage
    | | ++-rw trigger-event-count? uint16
  | ++-rw underflow
    | | ++-rw enabled?         boolean
    | | ++-rw underflow-threshold? rt-types:percentage
    | | ++-rw trigger-event-count? uint16

augment /te:te/te:tunnels/te:tunnel/te:p2p-primary-paths
/te:p2p-primary-path:
  | ++-rw static-lsp-name?  mpls-static:static-lsp-ref
augment /te:te/te:tunnels/te:tunnel/te:p2p-secondary-paths
/te:p2p-secondary-path:
  | ++-rw static-lsp-name?  mpls-static:static-lsp-ref
augment /te:te/te:globals/te:named-path-constraints
/te:named-path-constraint:
  | ++-rw bandwidth
    | | ++-rw specification-type?
      | | te-packet-types:te-bandwidth-requested-type
    | | ++-rw set-bandwidth?  te-packet-types:bandwidth-kbps
    | | ++-rw class-type?     te-types:te-ds-class
    | | ++-ro state
      | | ++-ro signaled-bandwidth?  te-packet-types:bandwidth-kbps
augment /te:te/te:tunnels/te:tunnel/te:p2p-primary-paths
/te:p2p-primary-path/te:lsps/te:lsp:
  | ++-ro performance-metrics-one-way
    | | ++-ro one-way-delay?      uint32
    | | ++-ro one-way-delay-normality?
      | | te-types:performance-metrics-normality
    | | ++-ro one-way-residual-bandwidth?
      | | rt-types:bandwidth-ieee-float32
    | | ++-ro one-way-residual-bandwidth-normality?
      | | te-types:performance-metrics-normality
Figure 2: MPLS TE model configuration and state tree

2.3. MPLS TE YANG Module

<CODE BEGINS> file "ietf-te-mpls@2019-02-23.yang"
module ietf-te-mpls {
  yang-version 1.1;

  /* Replace with IANA when assigned */
  prefix "te-mpls";
}

/* Import TE base model */
import ietf-te {
  prefix te;
  reference "draft-ietf-teas-yang-te: A YANG Data Model for Traffic
  Engineering Tunnels and Interfaces";
}

import ietf-te-device {
  prefix te-dev;
  reference "draft-ietf-teas-yang-te: A YANG Data Model for Traffic
  Engineering Tunnels and Interfaces";
}

/* Import TE MPLS types */
import ietf-te-packet-types {
  prefix "te-packet-types";
  reference "draft-ietf-teas-yang-te-types: A YANG Data Model for
  Common Traffic Engineering Types";
}

/* Import TE generic types */
import ietf-te-types {
  prefix te-types;
  reference "draft-ietf-teas-yang-te-types: A YANG Data Model for
  Common Traffic Engineering Types";
}

/* Import routing types */
import ietf-routing-types {
  prefix rt-types;
  reference "RFC8294: Common YANG Data Types for the Routing Area";
}

import ietf-mpls-static {
  prefix mpls-static;
  reference "draft-ietf-mpls-static-yang: A YANG Data Model
  for MPLS Static LSPs";
}

import ietf-inet-types {
  prefix inet;
  reference "RFC6991: Common YANG Data Types";
}

organization
"IETF Traffic Engineering Architecture and Signaling (TEAS)
Working Group";
revision "2019-02-23" {
  description "Latest update to MPLS TE YANG module.";
  reference
    "RFCXXXX: A YANG Data Model for MPLS-TE Tunnels and LSP(s)";
}

/* MPLS TE tunnel properties*/

grouping tunnel-igp-shortcut-config {
  description "TE tunnel IGP shortcut configs";
  leaf shortcut-eligible {
    type boolean;
    default "true";
    description
      "Whether this LSP is considered to be eligible for us as a
       shortcut in the IGP. In the case that this leaf is set to
       true, the IGP SPF calculation uses the metric specified to
       determine whether traffic should be carried over this LSP";
  }
  leaf metric-type {
    type identityref {
      base te-types:lsp-metric-type;
    }
    default te-types:lsp-metric-inherited;
    description
      "The type of metric specification that should be used to set
       the LSP(s) metric";
  }
  leaf metric {
    type int32;
    description
      "The value of the metric that should be specified. The value
       supplied in this leaf is used in conjunction with the metric
       type to determine the value of the metric used by the system.
       Where the metric-type is set to lsp-metric-absolute - the
       value of this leaf is used directly; where it is set to
       lsp-metric-relative, the relevant (positive or negative)
       offset is used to formulate the metric; where metric-type
       is lsp-metric-inherited, the value of this leaf is not
       utilized";
  }
  leaf-list routing-afs {
    type inet:ip-version;
    description
      "Address families";
  }
}
grouping tunnel-igp-shortcuts {
    description "TE tunnel IGP shortcut grouping";
    container tunnel-igp-shortcut {
        description "Tunnel IGP shortcut properties";
        uses tunnel-igp-shortcut-config;
    }
}

grouping tunnel-forwarding-adjacency-configs {
    description "Tunnel forwarding adjacency grouping";
    leaf binding-label {
        type rt-types:mpls-label;
        description "MPLS tunnel binding label";
    }
    leaf load-share {
        type uint32 {
            range "1..4294967295";
        }
        description "ECMP tunnel forwarding load-share factor.";
    }
    leaf policy-class {
        type uint8 {
            range "1..7";
        }
        description "The class associated with this tunnel";
    }
}

grouping tunnel-forwarding-adjacency {
    description "Properties for using tunnel in forwarding.";
    container forwarding {
        description "Tunnel forwarding properties container";
        uses tunnel-forwarding-adjacency-configs;
    }
}

/*** End of MPLS TE tunnel configuration/state */
grouping te-lsp-auto-bandwidth-config {
    description "Configuration parameters related to autobandwidth";

    leaf enabled {
        type boolean;
    }
}
default false;
description
"Enables MPLS auto-bandwidth on the
LSP";
}

leaf min-bw {
type te-packet-types:bandwidth-kbps;
description
"set the minimum bandwidth in Kbps for an
auto-bandwidth LSP";
}

leaf max-bw {
type te-packet-types:bandwidth-kbps;
description
"set the maximum bandwidth in Kbps for an
auto-bandwidth LSP";
}

leaf adjust-interval {
type uint32;
description
"time in seconds between adjustments to
LSP bandwidth";
}

leaf adjust-threshold {
type rt-types:percentage;
description
"percentage difference between the LSP’s
specified bandwidth and its current bandwidth
allocation -- if the difference is greater than the
specified percentage, auto-bandwidth adjustment is
triggered";
}

grouping te-lsp-overflow-config {
description
"configuration for MPLS LSP bandwidth
overflow adjustment";

leaf enabled {
type boolean;
default false;
description
"Enables MPLS LSP bandwidth overflow
leaf overflow-threshold {
  type rt-types:percentage;
  description
  "bandwidth percentage change to trigger an overflow event";
}

leaf trigger-event-count {
  type uint16;
  description
  "number of consecutive overflow sample events needed to trigger an overflow adjustment";
}

grouping te-lsp-underflow-config {
  description
  "configuration for MPLS LSP bandwidth underflow adjustment";

  leaf enabled {
    type boolean;
    default false;
    description
    "enables bandwidth underflow adjustment on the LSP";
  }

  leaf underflow-threshold {
    type rt-types:percentage;
    description
    "bandwidth percentage change to trigger and underflow event";
  }

  leaf trigger-event-count {
    type uint16;
    description
    "number of consecutive underflow sample events needed to trigger an underflow adjustment";
  }
}
grouping te-tunnel-bandwidth-config {
  description
"Configuration parameters related to bandwidth for a tunnel";

leaf specification-type {
  type te-packet-types:te-bandwidth-requested-type;
  default specified;
  description
    "The method used for setting the bandwidth, either explicitly
    specified or configured";
}

leaf set-bandwidth {
  when ".../specification-type = 'specified'" {
    description
      "The bandwidth value when bandwidth is explicitly
      specified";
  }
  type te-packet-types:bandwidth-kbps;
  description
    "set bandwidth explicitly, e.g., using
    offline calculation";
}

leaf class-type {
  type te-types:te-ds-class;
  description
    "The Class-Type of traffic transported by the LSP."
    reference "RFC4124: section-4.3.1";
}

grouping te-tunnel-bandwidth-state {
  description
    "Operational state parameters relating to bandwidth for a tunnel";

  leaf signaled-bandwidth {
    type te-packet-types:bandwidth-kbps;
    description
      "The currently signaled bandwidth of the LSP. In the case where
      the bandwidth is specified explicitly, then this will match the
      value of the set-bandwidth leaf; in cases where the bandwidth is
      dynamically computed by the system, the current value of the
      bandwidth should be reflected.";
  }
}

grouping tunnel-bandwidth_top {
  description
    "Top level grouping for specifying bandwidth for a tunnel";
}

container bandwidth-mpls {
    description "Bandwidth configuration for TE LSPs";

    uses te-tunnel-bandwidth-config;
}

container state {
    config false;
    description "State parameters related to bandwidth configuration of TE tunnels";
    uses te-tunnel-bandwidth-state;
}

container auto-bandwidth {
    when ".../specification-type = 'auto'" {
        description "Include this container for auto bandwidth specific configuration";
    }
    description "Parameters related to auto-bandwidth";

    uses te-lsp-auto-bandwidth-config;
}

container overflow {
    description "configuration of MPLS overflow bandwidth adjustment for the LSP";

    uses te-lsp-overflow-config;
}

container underflow {
    description "configuration of MPLS underflow bandwidth adjustment for the LSP";

    uses te-lsp-underflow-config;
}
}
}

grouping te-path-bandwidth_top {
    description "Top level grouping for specifying bandwidth for a TE path";
}
container bandwidth {
  description "Bandwidth configuration for TE LSPs";
  uses te-tunnel-bandwidth-config;
}

container state {
  config false;
  description "State parameters related to bandwidth configuration of TE tunnels";
  uses te-tunnel-bandwidth-state;
}

/**
 * MPLS TE augmentations
 */
augment "/te:te/te-dev:performance-thresholds" {
  uses te-packet-types:performance-metrics-throttle-container-packet;
  description "Performance parameters configurable thresholds";
}

/* MPLS TE interface augmentations */

/* MPLS TE tunnel augmentations */
augment "/te:te:te:tunnels/te:tunnel" {
  description "MPLS TE tunnel config augmentations";
  uses tunnel-igp-shortcuts;
  uses tunnel-forwarding-adjacency;
  uses tunnel-bandwidth_top;
}

/* MPLS TE LSPs augmentations */
augment "/te:te:tunnels/te:tunnel/" +
  "/te:p2p-primary-paths/te:p2p-primary-path" {
    when "/te:te:tunnels/te:tunnel/" +
      "/te:p2p-primary-paths/te:p2p-primary-path" +
      "/te:path-setup-protocol = 'te-types:path-setup-static'" {
      description "When the path is statically provisioned";
    }
  description "MPLS TE LSP augmentation";
  leaf static-lsp-name {
  ...
3. IANA Considerations

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

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-te-mpls
prefix:     ietf-te-mpls
reference:  RFC3209

4. Security Considerations

The YANG module defined in this memo is designed to be accessed via the NETCONF protocol [RFC6241]. The lowest NETCONF layer is the secure transport layer and the mandatory-to-implement secure transport is SSH [RFC6242]. The NETCONF access control model [RFC8341] provides means to restrict access for particular NETCONF users to a pre-configured subset of all available NETCONF protocol operations and content.

A number of data nodes defined in this YANG module are writable/creatable/deletable (i.e., config true, which is the default). These data nodes 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 MPLS network operations. Following are the subtrees and data nodes and their sensitivity/vulnerability:

"/te/tunnels": The augmentation to this list specifies configuration to TE tunnels on a device. Unauthorized access to this list could cause the device to ignore packets it should receive and process.

"/te/globals": The augmentation to this target specifies configuration applicable to the to all or one TE device. Unauthorized access to this list could cause the device to ignore packets it should receive and process.

5. Contributors

Himanshu Shah
Ciena
Email: hshah@ciena.com

6. Normative References
[I-D.ietf-mpls-static-yang]

[I-D.ietf-teas-yang-rsvp]

[I-D.ietf-teas-yang-rsvp-te]

[I-D.ietf-teas-yang-te]

[I-D.ietf-teas-yang-te-types]


Internet-Draft       MPLS TE Tunnels YANG Data Model       February 2019

[RFc6020] Bjorklund, M., Ed., "YANG - A Data Modeling Language for
the Network Configuration Protocol (NETCONF)", RFC 6020,
DOI 10.17487/RFC6020, October 2010,

and A. Bierman, Ed., "Network Configuration Protocol
(NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011,

[RFc6242] Wasserman, M., "Using the NETCONF Protocol over Secure
Shell (SSH)", RFC 6242, DOI 10.17487/RFC6242, June 2011,

RFC 6991, DOI 10.17487/RFC6991, July 2013,

RFC 7950, DOI 10.17487/RFC7950, August 2016,

[RFc8040] Bierman, A., Bjorklund, M., and K. Watsen, "RESTCONF
Protocol", RFC 8040, DOI 10.17487/RFC8040, January 2017,

[RFc8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC
2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174,

[RFc8294] Liu, X., Qu, Y., Lindem, A., Hopps, C., and L. Berger,
"Common YANG Data Types for the Routing Area",
RFC 8294, DOI 10.17487/RFC8294, December 2017,

[RFc8341] Bierman, A. and M. Bjorklund, "Network Configuration
Access Control Model", STD 91, RFC 8341,
DOI 10.17487/RFC8341, March 2018,

Authors’ Addresses

Tarek Saad
Cisco Systems Inc

Email: tsaad@cisco.com