

TEAS Working Group
Internet-Draft
Intended status: Standards Track
Expires: March 16, 2019

T. Saad, Ed.
R. Gandhi
Cisco Systems Inc
X. Liu
Volta Networks
V. Beeram
Juniper Networks
H. Shah
Ciena
I. Bryskin
Y. Lee
Huawei Technologies
September 12, 2018

Traffic Engineering Common YANG Types
draft-ietf-teas-yang-te-types-00

Abstract

This document defines a collection of common data types and groupings in YANG data modeling language. These derived common types and groupings are intended to be imported by modules that model Traffic Engineering (TE) configuration and state capabilities.

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 March 16, 2019.

Copyright Notice

Copyright (c) 2018 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
2.	Overview	3
3.	IETF TE Types YANG Module	3
4.	IETF MPLS TE Types YANG Module	53
5.	IANA Considerations	57
6.	Security Considerations	57
7.	Acknowledgement	57
8.	Normative References	57
	Authors' Addresses	59

[1.](#) Introduction

YANG [[RFC6020](#)] [[RFC7950](#)] is a data modeling language used to model configuration data, state data, Remote Procedure Calls, and notifications for network management protocols. The YANG language supports a small set of built-in data types and provides mechanisms to derive other types from the built-in types.

This document introduces a collection of common data types derived from the built-in YANG data types. The derived types are designed to be the common types applicable for modeling for TE features (e.g. in models defined in [[I-D.ietf-teas-yang-te](#)] and [[I-D.ietf-teas-yang-rsvp](#)]).

[1.1.](#) Terminology

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 [BCP 14](#), [RFC 2119](#) [[RFC2119](#)].

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.

Prefix	YANG module	Reference
yang	ietf-yang-types	[RFC6991]
inet	ietf-inet-types	[RFC6991]
te-types	ietf-te-types	this document
te-mpls-types	ietf-te-mpls-types	this document

Table 1: Prefixes and corresponding YANG modules

2. Overview

The TE generic types module covers the building blocks that are independent and agnostic of any specific technology or control plane instance. The MPLS TE types modules covers the common types reusable groupings specific to MPLS technology. Other technology specific TE types are outside the scope of this document.

This document defines two YANG modules for common TE types: `ietf-te-types` and `ietf-te-mpls-types`. The TE module imports (`ietf-yang-types`, `ietf-inet-types` and `ietf-routing-types`; see [Section 3](#)) are from [\[RFC6991\]](#) and [\[RFC8294\]](#).

3. IETF TE Types YANG Module

```
<CODE BEGINS> file "ietf-te-types@2018-09-13.yang"
module ietf-te-types {

  namespace "urn:ietf:params:xml:ns:yang:ietf-te-types";

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

  import ietf-inet-types {
    prefix inet;
  }

  import ietf-yang-types {
    prefix "yang";
  }
}
```



```
import ietf-routing-types {
  prefix "rt-types";
}

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>

  WG Chair: Lou Berger
             <mailto:lberger@labn.net>

  WG Chair: Vishnu Pavan Beeram
             <mailto:vbeeram@juniper.net>

  Editor:   Tarek Saad
             <mailto:tsaad@cisco.com>

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

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

  Editor:   Himanshu Shah
             <mailto:hshah@ciena.com>

  Editor:   Xufeng Liu
             <mailto:xufeng.liu.ietf@gmail.com>

  Editor:   Igor Bryskin
             <mailto:Igor.Bryskin@huawei.com>

  Editor:   Young Lee
             <mailto:leeyoung@huawei.com>";

description
  "This module contains a collection of generally
  useful TE specific YANG data type definitions.";

revision "2018-09-13" {
  description "Latest revision of TE types";
  reference "RFC3209";
}
```



```
/**
 * Typedefs
 */
typedef te-bandwidth {
    type string {
        pattern
            '0[xX](0((\.0?)?[pP](\+)?0?|(\.0?))|'
            + '1(\.([\da-fA-F]{0,5}[02468aAcCeE]?))?[pP](\+)?(12[0-7]|'
            + '1[01]\d|0?\d?\d)?)|0[xX][\da-fA-F]{1,8}|\d+'
            + '(\.0[xX](0((\.0?)?[pP](\+)?0?|(\.0?))|'
            + '1(\.([\da-fA-F]{0,5}[02468aAcCeE]?))?[pP](\+)?(12[0-7]|'
            + '1[01]\d|0?\d?\d)?)|0[xX][\da-fA-F]{1,8}|\d+))*';
    }
    description
        "This is the generic bandwidth type that is a string containing
        a list of numbers separated by commas, with each of these
        number can be non-negative decimal, hex integer, or hex float:
        (dec | hex | float)[*(','(dec | hex | float))].
        For packet switching type, a float number is used, such as
        0x1p10.
        For OTN switching type, a list of integers can be used, such
        as '0,2,3,1', indicating 2 odu0's and 1 odu3.
        For DWDM, a list of pairs of slot number and width can be
        used, such as '0, 2, 3, 3', indicating a frequency slot 0 with
        slot width 2 and a frequency slot 3 with slot width 3.";
} // te-bandwidth

typedef te-ds-class {
    type uint8 {
        range "0..7";
    }
    description
        "The Differentiated Class-Type of traffic.";
    reference "RFC4124: section-4.3.1";
}

typedef te-link-direction {
    type enumeration {
        enum INCOMING {
            description
                "explicit route represents an incoming link on a node";
        }
        enum OUTGOING {
            description
                "explicit route represents an outgoing link on a node";
        }
    }
    description
```



```
    "enumerated type for specifying direction of link on a node";  
}
```

```
typedef te-label-direction {  
    type enumeration {  
        enum FORWARD {  
            description  
                "Label allocated for the forward LSP direction";  
        }  
        enum REVERSE {  
            description  
                "Label allocated for the reverse LSP direction";  
        }  
    }  
    description  
        "enumerated type for specifying the forward or reverse  
        label";  
}
```

```
typedef te-hop-type {  
    type enumeration {  
        enum LOOSE {  
            description  
                "loose hop in an explicit path";  
        }  
        enum STRICT {  
            description  
                "strict hop in an explicit path";  
        }  
    }  
    description  
        "enumerated type for specifying loose or strict  
        paths";  
    reference "RFC3209: section-4.3.2";  
}
```

```
typedef optimization-goal {  
    type enumeration {  
        enum minimize {  
            description "Pick lowest path metric goal";  
        }  
        enum maximize {  
            description "Pick highest path metric goal";  
        }  
        enum randomize {  
            description  
                "Pick a path at random from list of  
                equally favorable ones";  
        }  
    }  
}
```



```
    }
  }
  description "TE optimization goal";
}

typedef percentage {
  type uint8 {
    range "0..100";
  }
  description
    "Integer indicating a percentage value";
}

typedef performance-metric-normality {
  type enumeration {
    enum "unknown" {
      value 0;
      description
        "Unknown.";
    }
    enum "normal" {
      value 1;
      description
        "Normal.";
    }
    enum "abnormal" {
      value 2;
      description
        "Abnormal. The anomalous bit is set.";
    }
  }
  description
    "Indicates whether a performance metric is normal, abnormal, or
    unknown.";
  reference
    "RFC7471: OSPF Traffic Engineering (TE) Metric Extensions.
    RFC7810: IS-IS Traffic Engineering (TE) Metric Extensions.
    RFC7823: Performance-Based Path Selection for Explicitly
    Routed Label Switched Paths (LSPs) Using TE Metric
    Extensions";
}

typedef te-admin-status {
  type enumeration {
    enum up {
      description
        "Enabled.";
    }
  }
}
```



```
    enum down {
        description
            "Disabled.";
    }
    enum testing {
        description
            "In some test mode.";
    }
    enum preparing-maintenance {
        description
            "Resource is disabled in the control plane to prepare for
            graceful shutdown for maintenance purposes.";
        reference
            "RFC5817: Graceful Shutdown in MPLS and Generalized MPLS
            Traffic Engineering Networks";
    }
    enum maintenance {
        description
            "Resource is disabled in the data plane for maintenance
            purposes.";
    }
}
description
    "Defines a type representing the administrative status of
    a TE resource.";
}

typedef te-global-id {
    type uint32;
    description
        "An identifier to uniquely identify an operator, which can be
        either a provider or a client.
        The definition of this type is taken from RFC6370 and RFC5003.
        This attribute type is used solely to provide a globally
        unique context for TE topologies.";
}

typedef te-link-access-type {
    type enumeration {
        enum point-to-point {
            description
                "The link is point-to-point.";
        }
        enum multi-access {
            description
                "The link is multi-access, including broadcast and NBMA.";
        }
    }
}
```



```
    description
      "Defines a type representing the access type of a TE link.";
    reference
      "RFC3630: Traffic Engineering (TE) Extensions to OSPF
      Version 2.";
  }

typedef te-node-id {
  type yang:dotted-quad;
  description
    "An identifier for a node in a topology.
    The identifier is represented as 32-bit unsigned integer in
    the dotted-quad notation.
    This attribute is mapped to Router ID in
    RFC3630, RFC5329, RFC5305, and RFC6119.";
}

typedef te-oper-status {
  type enumeration {
    enum up {
      description
        "Operational up.";
    }
    enum down {
      description
        "Operational down.";
    }
    enum testing {
      description
        "In some test mode.";
    }
    enum unknown {
      description
        "Status cannot be determined for some reason.";
    }
    enum preparing-maintenance {
      description
        "Resource is disabled in the control plane to prepare for
        graceful shutdown for maintenance purposes.";
      reference
        "RFC5817: Graceful Shutdown in MPLS and Generalized MPLS
        Traffic Engineering Networks";
    }
    enum maintenance {
      description
        "Resource is disabled in the data plane for maintenance
        purposes.";
    }
  }
}
```



```
    }
    description
      "Defines a type representing the operational status of
      a TE resource.";
  }

typedef te-path-disjointness {
  type bits {
    bit node {
      position 0;
      description "Node disjoint.";
    }
    bit link {
      position 1;
      description "Link disjoint.";
    }
    bit srlg {
      position 2;
      description "SRLG (Shared Risk Link Group) disjoint.";
    }
  }
  description
    "Type of the resource disjointness for a TE tunnel path.";
  reference
    "RFC4872: RSVP-TE Extensions in Support of End-to-End
    Generalized Multi-Protocol Label Switching (GMPLS)
    Recovery";
} // te-path-disjointness

typedef te-recovery-status {
  type enumeration {
    enum normal {
      description
        "Both the recovery and working spans are fully
        allocated and active, data traffic is being
        transported over (or selected from) the working
        span, and no trigger events are reported.";
    }
    enum recovery-started {
      description
        "The recovery action has been started, but not completed.";
    }
    enum recovery-succeeded {
      description
        "The recovery action has succeeded. The working span has
        reported a failure/degrade condition and the user traffic
        is being transported (or selected) on the recovery span.";
    }
  }
}
```



```
enum recovery-failed {
    description
        "The recovery action has failed.";
}
enum reversion-started {
    description
        "The reversion has started.";
}
enum reversion-failed {
    description
        "The reversion has failed.";
}
enum recovery-unavailable {
    description
        "The recovery is unavailable -- either as a result of an
        operator Lockout command or a failure condition detected
        on the recovery span.";
}
enum recovery-admin {
    description
        "The operator has issued a command switching the user
        traffic to the recovery span.";
}
enum wait-to-restore {
    description
        "The recovery domain is recovering from a failure/degrade
        condition on the working span that is being controlled by
        the Wait-to-Restore (WTR) timer.";
}
}
description
    "Defines the status of a recovery action.";
reference
    "RFC4427: Recovery (Protection and Restoration) Terminology
    for Generalized Multi-Protocol Label Switching (GMPLS).
    RFC6378: MPLS Transport Profile (MPLS-TP) Linear Protection";
}

typedef te-template-name {
    type string {
        pattern '/?([a-zA-Z0-9\-\_\.]+)(/[a-zA-Z0-9\-\_\.]+)*';
    }
    description
        "A type for the name of a TE node template or TE link
        template.";
}

typedef te-topology-event-type {
```



```
type enumeration {
  enum "add" {
    value 0;
    description
      "A TE node or te-link has been added.";
  }
  enum "remove" {
    value 1;
    description
      "A TE node or te-link has been removed.";
  }
  enum "update" {
    value 2;
    description
      "A TE node or te-link has been updated.";
  }
}
description "TE Event type for notifications";
} // te-topology-event-type

typedef te-topology-id {
  type string {
    pattern
      '([a-zA-Z0-9\-\_\.]+:)*'
      + '/?([a-zA-Z0-9\-\_\.]+)(/[a-zA-Z0-9\-\_\.]+)*';
  }
  description
    "An identifier for a topology.
    It is optional to have one or more prefixes at the beginning,
    separated by colons. The prefixes can be the network-types,
    defined in ietf-network.yang, to help user to understand the
    topology better before further inquiry.";
}

typedef te-tp-id {
  type union {
    type uint32; // Unnumbered
    type inet:ip-address; // IPv4 or IPv6 address
  }
  description
    "An identifier for a TE link endpoint on a node.
    This attribute is mapped to local or remote link identifier in
    RFC3630 and RFC5305.";
}

typedef admin-group {
  type binary {
    length 4;
```



```
    }
    description
      "Administrative group/Resource class/Color.";
    reference "RFC3630 and RFC5305";
  }

typedef extended-admin-group {
  type binary;
  description
    "Extended administrative group/Resource class/Color.";
  reference "RFC7308";
}

typedef admin-groups {
  type union {
    type admin-group;
    type extended-admin-group;
  }
  description "TE administrative group derived type";
}

typedef srlg {
  type uint32;
  description "SRLG type";
  reference "RFC4203 and RFC5307";
}

typedef te-metric {
  type uint32;
  description
    "TE link metric";
  reference "RFC3785";
}

/* TE features */
feature p2mp-te {
  description
    "Indicates support for P2MP-TE";
  reference "RFC4875";
}

feature frr-te {
  description
    "Indicates support for TE FastReroute (FRR)";
  reference "RFC4090";
}

feature extended-admin-groups {
```



```
    description
      "Indicates support for TE link extended admin
      groups.";
    reference "RFC7308";
  }

  feature named-path-affinities {
    description
      "Indicates support for named path affinities";
  }

  feature named-extended-admin-groups {
    description
      "Indicates support for named extended admin groups";
  }

  feature named-srlg-groups {
    description
      "Indicates support for named SRLG groups";
  }

  feature named-path-constraints {
    description
      "Indicates support for named path constraints";
  }

  feature path-optimization-metric {
    description
      "Indicates support for path optimization metric";
  }

  feature path-optimization-objective-function {
    description
      "Indicates support for path optimization objective function";
  }

  /*
   * Identities
   */
  identity association-type {
    description "Base identity for tunnel association";
    reference "RFC6780, RFC4872, RFC4873";
  }
  identity association-type-recovery {
    base association-type;
    description
      "Association Type Recovery used to association LSPs of
      same tunnel for recovery";
```



```
    reference "RFC4872";
}
identity association-type-resource-sharing {
    base association-type;
    description
        "Association Type Resource Sharing used to enable resource
        sharing during make-before-break.";
    reference "RFC4873";
}
identity association-type-double-sided-bidir {
    base association-type;
    description
        "Association Type Double Sided bidirectional used to associate
        two LSPs of two tunnels that are independently configured on
        either endpoint";
    reference "RFC7551";
}
identity association-type-single-sided-bidir {
    base association-type;
    description
        "Association Type Single Sided bidirectional used to associate
        two LSPs of two tunnels, where a tunnel is configured on one
        side/endpoint, and the other tunnel is dynamically created on
        the other endpoint";
    reference "RFC7551";
}

identity objective-function-type {
    description "Base objective function type";
    reference "RFC4657";
}
identity of-minimize-cost-path {
    base objective-function-type;
    description
        "Minimize cost of path objective function";
    reference "RFC5541";
}
identity of-minimize-load-path {
    base objective-function-type;
    description
        "Minimize the load on path(s) objective
        function";
}
identity of-maximize-residual-bandwidth {
    base objective-function-type;
    description
        "Maximize the residual bandwidth objective
        function";
}
```



```
}
identity of-minimize-agg-bandwidth-consumption {
  base objective-function-type;
  description
    "minimize the aggregate bandwidth consumption
    objective function";
}
identity of-minimize-load-most-loaded-link {
  base objective-function-type;
  description
    "Minimize the load on the most loaded link
    objective function";
}
identity of-minimize-cost-path-set {
  base objective-function-type;
  description
    "Minimize the cost on a path set objective
    function";
}

identity path-computation-method {
  description
    "base identity for supported path computation
    mechanisms";
}
identity path-locally-computed {
  base path-computation-method;
  description
    "indicates a constrained-path LSP in which the
    path is computed by the local LER";
}
identity path-externally-queried {
  base path-computation-method;
  description
    "Constrained-path LSP in which the path is
    obtained by querying an external source, such as a PCE server.
    In the case that an LSP is defined to be externally queried, it
    may also have associated explicit definitions (provided
    to the external source to aid computation); and the path that is
    returned by the external source is not required to provide a
    wholly resolved path back to the originating system - that is to
    say, some local computation may also be required";
}
identity path-explicitly-defined {
  base path-computation-method;
  description
    "constrained-path LSP in which the path is
    explicitly specified as a collection of strict or/and loose
```



```
        hops";
    }

    identity LSP_METRIC_TYPE {
        description
            "Base identity for types of LSP metric specification";
    }
    identity LSP_METRIC_RELATIVE {
        base LSP_METRIC_TYPE;
        description
            "The metric specified for the LSPs to which this identity refers
            is specified as a relative value to the IGP metric cost to the
            LSP's tail-end.";
    }
    identity LSP_METRIC_ABSOLUTE {
        base LSP_METRIC_TYPE;
        description
            "The metric specified for the LSPs to which this identity refers
            is specified as an absolute value";
    }
    identity LSP_METRIC_INHERITED {
        base LSP_METRIC_TYPE;
        description
            "The metric for the LSPs to which this identity refers is
            not specified explicitly - but rather inherited from the IGP
            cost directly";
    }

    identity tunnel-type {
        description
            "Base identity from which specific tunnel types are
            derived.";
    }
    identity tunnel-p2p {
        base tunnel-type;
        description
            "TE point-to-point tunnel type.";
    }
    identity tunnel-p2mp {
        base tunnel-type;
        description
            "TE point-to-multipoint tunnel type.";
        reference "RFC4875";
    }

    identity tunnel-action-type {
        description
            "Base identity from which specific tunnel action types
```



```
        are derived.";
    }
    identity tunnel-action-resetup {
        base tunnel-action-type;
        description
            "TE tunnel action resetup. Tears the
            tunnel's current LSP (if any) and
            attempts to re-establish a new LSP";
    }
    identity tunnel-action-reoptimize {
        base tunnel-action-type;
        description
            "TE tunnel action reoptimize.
            Reoptimizes placement of the tunnel LSP(s)";
    }
    identity tunnel-action-switchpath {
        base tunnel-action-type;
        description
            "TE tunnel action switchpath
            Switches the tunnel's LSP to use the specified path";
    }
    identity te-action-result {
        description
            "Base identity from which specific TE action results
            are derived.";
    }
    identity te-action-success {
        base te-action-result;
        description "TE action successful.";
    }
    identity te-action-fail {
        base te-action-result;
        description "TE action failed.";
    }
    identity tunnel-action-inprogress {
        base te-action-result;
        description "TE action inprogress.";
    }
    identity tunnel-admin-state-type {
        description
            "Base identity for TE tunnel admin states";
    }
    identity tunnel-admin-state-up {
        base tunnel-admin-state-type;
        description "Tunnel administratively state up";
    }
}
```



```
identity tunnel-admin-state-down {
  base tunnel-admin-state-type;
  description "Tunnel administratively state down";
}

identity tunnel-state-type {
  description
    "Base identity for TE tunnel states";
}

identity tunnel-state-up {
  base tunnel-state-type;
  description "Tunnel state up";
}

identity tunnel-state-down {
  base tunnel-state-type;
  description "Tunnel state down";
}

identity lsp-state-type {
  description
    "Base identity for TE LSP states";
}

identity lsp-path-computing {
  base lsp-state-type;
  description
    "State path compute in progress";
}

identity lsp-path-computation-ok {
  base lsp-state-type;
  description
    "State path compute successful";
}

identity lsp-path-computation-failed {
  base lsp-state-type;
  description
    "State path compute failed";
}

identity lsp-state-setting-up {
  base lsp-state-type;
  description
    "State setting up";
}

identity lsp-state-setup-ok {
  base lsp-state-type;
  description
    "State setup successful";
}

identity lsp-state-setup-failed {
```



```
    base lsp-state-type;
    description
        "State setup failed";
}
identity lsp-state-up {
    base lsp-state-type;
    description "State up";
}
identity lsp-state-tearing-down {
    base lsp-state-type;
    description
        "State tearing down";
}
identity lsp-state-down {
    base lsp-state-type;
    description "State down";
}

identity path-invalidation-action-type {
    description
        "Base identity for TE path invalidation action types";
}
identity path-invalidation-action-drop-type {
    base path-invalidation-action-type;
    description
        "TE path invalidation action drop";
}
identity path-invalidation-action-drop-tear {
    base path-invalidation-action-type;
    description
        "TE path invalidation action tear";
}

identity lsp-restoration-type {
    description
        "Base identity from which LSP restoration types are
        derived.";
}
identity lsp-restoration-restore-any {
    base lsp-restoration-type;
    description
        "Restores when any of the LSPs is affected by a failure";
}
identity lsp-restoration-restore-all {
    base lsp-restoration-type;
    description
        "Restores when all the tunnel LSPs are affected by failure";
}
```



```
identity restoration-scheme-type {
  description
    "Base identity for LSP restoration schemes";
  reference "RFC4872";
}
identity restoration-scheme-preconfigured {
  base restoration-scheme-type;
  description
    "Restoration LSP is preconfigured prior to the failure";
}
identity restoration-scheme-precomputed {
  base restoration-scheme-type;
  description
    "Restoration LSP is precomputed prior to the failure";
}
identity restoration-scheme-presignaled {
  base restoration-scheme-type;
  description
    "Restoration LSP is presignaled prior to the failure";
}

identity lsp-protection-type {
  description
    "Base identity from which LSP protection types are
    derived.";
}
identity lsp-protection-unprotected {
  base lsp-protection-type;
  description
    "LSP protection 'Unprotected'";
  reference "RFC4872";
}
identity lsp-protection-reroute-extra {
  base lsp-protection-type;
  description
    "LSP protection '(Full) Rerouting'";
  reference "RFC4872";
}
identity lsp-protection-reroute {
  base lsp-protection-type;
  description
    "LSP protection 'Rerouting without Extra-Traffic'";
  reference "RFC4872";
}
identity lsp-protection-1-for-n {
  base lsp-protection-type;
  description
    "LSP protection '1:N Protection with Extra-Traffic'";
```



```
    reference "RFC4872";
}
identity lsp-protection-unidir-1-to-1 {
    base lsp-protection-type;
    description
        "LSP protection '1+1 Unidirectional Protection'";
    reference "RFC4872";
}
identity lsp-protection-bidir-1-to-1 {
    base lsp-protection-type;
    description
        "LSP protection '1+1 Bidirectional Protection'";
    reference "RFC4872";
}
identity lsp-protection-extra-traffic {
    base lsp-protection-type;
    description
        "LSP protection 'Extra-Traffic'";
    reference
        "ITU-T G.808, RFC 4427.";
}

identity lsp-protection-state {
    description
        "Base identity of protection states for reporting
        purposes.";
}
identity normal {
    base lsp-protection-state;
    description "Normal state.";
}
identity signal-fail-of-protection {
    base lsp-protection-state;
    description
        "There is a SF condition on the protection transport
        entity which has higher priority than the FS command.";
    reference
        "ITU-T G.873.1, G.8031, G.8131";
}
identity lockout-of-protection {
    base lsp-protection-state;
    description
        "A Loss of Protection (LoP) command is active.";
    reference
        "ITU-T G.808, RFC 4427";
}
identity forced-switch {
    base lsp-protection-state;
```



```
    description
      "A forced switch (FS) command is active.";
    reference
      "ITU-T G.808, RFC 4427";
  }
  identity signal-fail {
    base lsp-protection-state;
    description
      "There is a SF condition on either the working
      or the protection path.";
    reference
      "ITU-T G.808, RFC 4427";
  }
  identity signal-degrade {
    base lsp-protection-state;
    description
      "There is an SD condition on either the working or the
      protection path.";
    reference
      "ITU-T G.808, RFC 4427";
  }
  identity manual-switch {
    base lsp-protection-state;
    description
      "A manual switch (MS) command is active.";
    reference
      "ITU-T G.808, RFC 4427";
  }
  identity wait-to-restore {
    base lsp-protection-state;
    description
      "A wait time to restore (WTR) is running.";
    reference
      "ITU-T G.808, RFC 4427";
  }
  identity do-not-revert {
    base lsp-protection-state;
    description
      "A DNR condition is active because of a non-revertive
      behavior.";
    reference
      "ITU-T G.808, RFC 4427";
  }
  identity failure-of-protocol {
    base lsp-protection-state;
    description
      "The protection is not working because of a failure of
      protocol condition.";
```



```
    reference
      "ITU-T G.873.1, G.8031, G.8131";
  }

  identity protection-external-commands {
    description
      "Protection external commands for trouble shooting
      purposes.";
  }
  identity action-freeze {
    base protection-external-commands;
    description
      "A temporary configuration action initiated by an operator
      command to prevent any switch action to be taken and as such
      freezes the current state.";
    reference
      "ITU-T G.808, RFC 4427";
  }
  identity clear-freeze {
    base protection-external-commands;
    description
      "An action that clears the active freeze state.";
    reference
      "ITU-T G.808, RFC 4427";
  }
  identity action-lockout-of-normal {
    base protection-external-commands;
    description
      "A temporary configuration action initiated by an operator
      command to ensure that the normal traffic is not allowed
      to use the protection transport entity.";
    reference
      "ITU-T G.808, RFC 4427";
  }
  identity clear-lockout-of-normal {
    base protection-external-commands;
    description
      "An action that clears the active lockout of normal state.";
    reference
      "ITU-T G.808, RFC 4427";
  }
  identity action-lockout-of-protection {
    base protection-external-commands;
    description
      "A temporary configuration action initiated by an operator
      command to ensure that the protection transport entity is
      temporarily not available to transport a traffic signal
      (either normal or extra traffic).";
```



```
    reference
      "ITU-T G.808, RFC 4427";
  }
  identity action-forced-switch {
    base protection-external-commands;
    description
      "A switch action initiated by an operator command to switch
       the extra traffic signal, the normal traffic signal, or the
       null signal to the protection transport entity, unless an
       equal or higher priority switch command is in effect.";
    reference
      "ITU-T G.808, RFC 4427";
  }
  identity action-manual-switch {
    base protection-external-commands;
    description
      "A switch action initiated by an operator command to switch
       the extra traffic signal, the normal traffic signal #i, or
       the null signal to the protection transport entity, unless
       a fault condition exists on other transport entities or an
       equal or higher priority switch command is in effect.";
    reference
      "ITU-T G.808, RFC 4427";
  }
  identity action-exercise {
    base protection-external-commands;
    description
      "An action to start testing if the APS communication is
       operating correctly. It is lower priority than any other
       state or command.";
    reference
      "ITU-T G.808, RFC 4427";
  }
  identity clear {
    base protection-external-commands;
    description
      "An action that clears the active near-end lockout of
       protection, forced switch, manual switch, WTR state,
       or exercise command.";
    reference
      "ITU-T G.808, RFC 4427";
  }
  identity switching-capabilities {
    description
      "Base identity for interface switching capabilities";
    reference "RFC3471";
  }
```



```
identity switching-psc1 {
  base switching-capabilities;
  description
    "Packet-Switch Capable-1 (PSC-1)";
  reference "RFC3471";
}
identity switching-evpl {
  base switching-capabilities;
  description
    "Ethernet Virtual Private Line (EVPL)";
}
identity switching-l2sc {
  base switching-capabilities;
  description
    "Layer-2 Switch Capable (L2SC)";
  reference "RFC3471";
}
identity switching-tdm {
  base switching-capabilities;
  description
    "Time-Division-Multiplex Capable (TDM)";
  reference "RFC3471";
}
identity switching-otn {
  base switching-capabilities;
  description
    "OTN-TDM capable";
}
identity switching-dcsc {
  base switching-capabilities;
  description
    "Data Channel Switching Capable (DCSC)";
}
identity switching-lsc {
  base switching-capabilities;
  description
    "Lambda-Switch Capable (LSC)";
  reference "RFC3471";
}
identity switching-fsc {
  base switching-capabilities;
  description
    "Fiber-Switch Capable (FSC)";
  reference "RFC3471";
}

identity lsp-encoding-types {
  description
```



```
    "Base identity for encoding types";
    reference "RFC3471";
}
identity lsp-encoding-packet {
    base lsp-encoding-types;
    description
        "Packet LSP encoding";
    reference "RFC3471";
}
identity lsp-encoding-ethernet {
    base lsp-encoding-types;
    description
        "Ethernet LSP encoding";
    reference "RFC3471";
}
identity lsp-encoding-pdh {
    base lsp-encoding-types;
    description
        "ANSI/ETSI LSP encoding";
    reference "RFC3471";
}
identity lsp-encoding-sdh {
    base lsp-encoding-types;
    description
        "SDH ITU-T G.707 / SONET ANSI T1.105 LSP encoding";
    reference "RFC3471";
}
identity lsp-encoding-digital-wrapper {
    base lsp-encoding-types;
    description
        "Digital Wrapper LSP encoding";
    reference "RFC3471";
}
identity lsp-encoding-lambda {
    base lsp-encoding-types;
    description
        "Lambda (photonic) LSP encoding";
    reference "RFC3471";
}
identity lsp-encoding-fiber {
    base lsp-encoding-types;
    description
        "Fiber LSP encoding";
    reference "RFC3471";
}
identity lsp-encoding-fiber-channel {
    base lsp-encoding-types;
    description
```



```
        "FiberChannel LSP encoding";
        reference "RFC3471";
    }
    identity lsp-encoding-oduk {
        base lsp-encoding-types;
        description
            "G.709 ODUK (Digital Path)LSP encoding";
    }
    identity lsp-encoding-optical-channel {
        base lsp-encoding-types;
        description
            "Line (e.g., 8B/10B) LSP encoding";
    }
    identity lsp-encoding-line {
        base lsp-encoding-types;
        description
            "Line (e.g., 8B/10B) LSP encoding";
    }
    identity path-signaling-type {
        description
            "base identity from which specific LSPs path
            setup types are derived";
    }
    identity path-setup-static {
        base path-signaling-type;
        description
            "Static LSP provisioning path setup";
    }
    identity path-setup-rsvp {
        base path-signaling-type;
        description
            "RSVP-TE signaling path setup";
        reference "RFC3209";
    }
    identity path-setup-sr {
        base path-signaling-type;
        description
            "Segment-routing path setup";
    }
    identity path-scope-type {
        description
            "base identity from which specific path
            scope types are derived";
    }
    identity path-scope-segment {
        base path-scope-type;
        description
```



```
    "Path scope segment";
}
identity path-scope-end-to-end {
    base path-scope-type;
    description
        "Path scope end to end";
}

identity route-usage-type {
    description
        "Base identity for route usage";
}
identity route-include-ero {
    base route-usage-type;
    description
        "Include ERO resource in route";
}
identity route-exclude-ero {
    base route-usage-type;
    description
        "Exclude ERO resource from route";
}
identity route-exclude-srlg {
    base route-usage-type;
    description
        "Exclude SRLG from route";
}

identity path-metric-type {
    description
        "Base identity for path metric type";
}
identity path-metric-te {
    base path-metric-type;
    description
        "TE path metric";
    reference "RFC3785";
}
identity path-metric-igp {
    base path-metric-type;
    description
        "IGP path metric";
    reference "RFC3785";
}
identity path-metric-hop {
    base path-metric-type;
    description
        "Hop path metric";
```



```
}
identity path-metric-delay-average {
  base path-metric-type;
  description
    "Unidirectional average link delay";
  reference "RFC7471";
}

identity path-metric-residual-bandwidth {
  base path-metric-type;
  description
    "Unidirectional Residual Bandwidth, which is defined to be
    Maximum Bandwidth [RFC3630] minus the bandwidth currently
    allocated to LSPs.";
  reference "RFC7471";
}

identity path-metric-optimize-includes {
  base path-metric-type;
  description
    "A metric that optimizes the number of included resources
    specified in a set";
}

identity path-metric-optimize-excludes {
  base path-metric-type;
  description
    "A metric that optimizes the number of excluded resources
    specified in a set";
}

identity path-tiebreaker-type {
  description
    "Base identity for path tie-breaker type";
}

identity path-tiebreaker-minfill {
  base path-tiebreaker-type;
  description
    "Min-Fill LSP path placement";
}

identity path-tiebreaker-maxfill {
  base path-tiebreaker-type;
  description
    "Max-Fill LSP path placement";
}

identity path-tiebreaker-random {
  base path-tiebreaker-type;
  description
    "Random LSP path placement";
}
```



```
identity bidir-provisioning-mode {
  description
    "Base identity for bidirectional provisioning
    mode.";
  reference "RFC7551";
}
identity bidir-provisioning-single-sided {
  base bidir-provisioning-mode;
  description
    "Single-sided bidirectional provisioning mode";
  reference "RFC7551";
}
identity bidir-provisioning-double-sided {
  base bidir-provisioning-mode;
  description
    "Double-sided bidirectional provisioning mode";
  reference "RFC7551";
}

identity bidir-association-type {
  description
    "Base identity for bidirectional association type";
  reference "RFC7551";
}
identity bidir-assoc-corouted {
  base bidir-association-type;
  description
    "Co-routed bidirectional association type";
  reference "RFC7551";
}
identity bidir-assoc-non-corouted {
  base bidir-association-type;
  description
    "Non co-routed bidirectional association type";
  reference "RFC7551";
}

identity resource-affinities-type {
  description
    "Base identity for resource affinities";
  reference "RFC2702";
}
identity resource-aff-include-all {
  base resource-affinities-type;
  description
    "The set of attribute filters associated with a
    tunnel all of which must be present for a link
    to be acceptable";
```



```
    reference "RFC2702 and RFC3209";
  }
  identity resource-aff-include-any {
    base resource-affinities-type;
    description
      "The set of attribute filters associated with a
       tunnel any of which must be present for a link
       to be acceptable";
    reference "RFC2702 and RFC3209";
  }
  identity resource-aff-exclude-any {
    base resource-affinities-type;
    description
      "The set of attribute filters associated with a
       tunnel any of which renders a link unacceptable";
    reference "RFC2702 and RFC3209";
  }

  identity te-optimization-criterion {
    description
      "Base identity for TE optimization criterion.";
    reference
      "RFC3272: Overview and Principles of Internet Traffic
       Engineering.";
  }
  identity not-optimized {
    base te-optimization-criterion;
    description "Optimization is not applied.";
  }
  identity cost {
    base te-optimization-criterion;
    description "Optimized on cost.";
  }
  identity delay {
    base te-optimization-criterion;
    description "Optimized on delay.";
  }

  identity path-computation-srlg-type {
    description
      "Base identity for SRLG path computation";
  }
  identity srlg-ignore {
    base path-computation-srlg-type;
    description
      "Ignores SRLGs in path computation";
  }
  identity srlg-strict {
```



```
    base path-computation-srlg-type;
    description
        "Include strict SRLG check in path computation";
}
identity srlg-preferred {
    base path-computation-srlg-type;
    description
        "Include preferred SRLG check in path computation";
}
identity srlg-weighted {
    base path-computation-srlg-type;
    description
        "Include weighted SRLG check in path computation";
}

identity otn-rate-type {
    description
        "Base type to identify OTN bit rates of various information
        structures.";
    reference "RFC7139";
}
identity odu0 {
    base otn-rate-type;
    description
        "ODU0 bit rate.";
}
identity odu1 {
    base otn-rate-type;
    description
        "ODU1 bit rate.";
}
identity odu2 {
    base otn-rate-type;
    description
        "ODU2 bit rate.";
}
identity odu3 {
    base otn-rate-type;
    description
        "ODU3 bit rate.";
}
identity odu4 {
    base otn-rate-type;
    description
        "ODU4 bit rate.";
}
identity odu2e {
    base otn-rate-type;
```



```
    description
        "ODU2e bit rate.";
}
identity oduc {
    base otn-rate-type;
    description
        "ODUCn bit rate.";
}
identity oduflex {
    base otn-rate-type;
    description
        "ODUflex bit rate.";
}

identity wdm-spectrum-type {
    description
        "Base type to identify WDM spectrum type.";
}
identity cwdm {
    base wdm-spectrum-type;
    description "CWDM.";
    reference "RFC6205";
}
identity dwdm {
    base wdm-spectrum-type;
    description "DWDM.";
    reference "RFC6205";
}
identity flexible-grid {
    base wdm-spectrum-type;
    description "Flexible grid.";
    reference "RFC6205";
}

/**
 * TE bandwidth groupings
 **/
grouping te-bandwidth {
    description
        "This grouping defines the generic TE bandwidth.
        For some known data plane technologies, specific modeling
        structures are specified. The string encoded te-bandwidth
        type is used for un-specified technologies.
        The modeling structure can be augmented later for other
        technologies.";
    container te-bandwidth {
        description
            "Container that specifies TE bandwidth.";
```



```
    choice technology {
      default generic;
      description
        "Data plane technology type.";
      case generic {
        leaf generic {
          type te-bandwidth;
          description
            "Bandwidth specified in a generic format.";
        }
      }
    }
  }
}

/**
 * TE label groupings
 **/
grouping te-label {
  description
    "This grouping defines the generic TE label.
    The modeling structure can be augmented for each technology.
    For un-specified technologies, rt-types:generalized-label
    is used.";
  container te-label {
    description
      "Container that specifies TE label.";
    choice technology {
      default generic;
      description
        "Data plane technology type.";
      case generic {
        leaf generic {
          type rt-types:generalized-label;
          description
            "TE label specified in a generic format.";
        }
      }
    }
    leaf direction {
      type te-label-direction;
      description "Label direction";
    }
  }
}

grouping te-topology-identifier {
  description
```



```
    "Augmentation for TE topology.";
  container te-topology-identifier {
    description "TE topology identifier container";
    leaf provider-id {
      type te-types:te-global-id;
      description
        "An identifier to uniquely identify a provider.";
    }
    leaf client-id {
      type te-types:te-global-id;
      description
        "An identifier to uniquely identify a client.";
    }
    leaf topology-id {
      type te-types:te-topology-id;
      description
        "It is presumed that a datastore will contain many
        topologies. To distinguish between topologies it is
        vital to have UNIQUE topology identifiers.";
    }
  }
}

/**
 * TE performance metric groupings
 **/
grouping performance-metric-one-way-delay-loss {
  description
    "Performance metric information in real time that can
    be applicable to links or connections. PM defined
    in this grouping is applicable to generic TE performance
    metrics as well as packet TE performance metrics.";
  reference
    "RFC7471: OSPF Traffic Engineering (TE) Metric Extensions.
    RFC7810: IS-IS Traffic Engineering (TE) Metric Extensions.
    RFC7823: Performance-Based Path Selection for Explicitly
    Routed Label Switched Paths (LSPs) Using TE Metric
    Extensions";
  leaf one-way-delay {
    type uint32 {
      range 0..16777215;
    }
    description "Delay or latency in micro seconds.";
  }
  leaf one-way-min-delay {
    type uint32 {
      range 0..16777215;
    }
  }
}
```



```
    description "Minimum delay or latency in micro seconds.";
  }
  leaf one-way-max-delay {
    type uint32 {
      range 0..16777215;
    }
    description "Maximum delay or latency in micro seconds.";
  }
  leaf one-way-delay-variation {
    type uint32 {
      range 0..16777215;
    }
    description "Delay variation in micro seconds.";
  }
  leaf one-way-packet-loss {
    type decimal64 {
      fraction-digits 6;
      range "0 .. 50.331642";
    }
    description
      "Packet loss as a percentage of the total traffic sent
      over a configurable interval. The finest precision is
      0.000003%.";
  }
}
```

```
grouping performance-metric-two-way-delay-loss {
  description
    "Performance metric information in real time that can
    be applicable to links or connections. PM defined
    in this grouping is applicable to generic TE performance
    metrics as well as packet TE performance metrics.";
  reference
    "RFC7471: OSPF Traffic Engineering (TE) Metric Extensions.
    RFC7810: IS-IS Traffic Engineering (TE) Metric Extensions.
    RFC7823: Performance-Based Path Selection for Explicitly
    Routed Label Switched Paths (LSPs) Using TE Metric
    Extensions";
  leaf two-way-delay {
    type uint32 {
      range 0..16777215;
    }
    description "Delay or latency in micro seconds.";
  }
  leaf two-way-min-delay {
    type uint32 {
      range 0..16777215;
    }
  }
}
```



```
    description "Minimum delay or latency in micro seconds.";
  }
  leaf two-way-max-delay {
    type uint32 {
      range 0..16777215;
    }
    description "Maximum delay or latency in micro seconds.";
  }
  leaf two-way-delay-variation {
    type uint32 {
      range 0..16777215;
    }
    description "Delay variation in micro seconds.";
  }
  leaf two-way-packet-loss {
    type decimal64 {
      fraction-digits 6;
      range "0 .. 50.331642";
    }
    description
      "Packet loss as a percentage of the total traffic sent
      over a configurable interval. The finest precision is
      0.000003%.";
  }
}

grouping performance-metric-one-way-bandwidth {
  description
    "Performance metric information in real time that can
    be applicable to links. PM defined
    in this grouping is applicable to generic TE performance
    metrics as well as packet TE performance metrics.";
  reference
    "RFC7471: OSPF Traffic Engineering (TE) Metric Extensions.
    RFC7810: IS-IS Traffic Engineering (TE) Metric Extensions.
    RFC7823: Performance-Based Path Selection for Explicitly
    Routed Label Switched Paths (LSPs) Using TE Metric
    Extensions";
  leaf one-way-residual-bandwidth {
    type rt-types:bandwidth-ieee-float32;
    description
      "Residual bandwidth that subtracts tunnel
      reservations from Maximum Bandwidth (or link capacity)
      [RFC3630] and provides an aggregated remainder across QoS
      classes.";
  }
  leaf one-way-available-bandwidth {
```



```
    type rt-types:bandwidth-ieee-float32;
    description
      "Available bandwidth that is defined to be residual
      bandwidth minus the measured bandwidth used for the
      actual forwarding of non-RSVP-TE LSP packets.  For a
      bundled link, available bandwidth is defined to be the
      sum of the component link available bandwidths.";
  }
  leaf one-way-utilized-bandwidth {
    type rt-types:bandwidth-ieee-float32;
    description
      "Bandwidth utilization that represents the actual
      utilization of the link (i.e. as measured in the router).
      For a bundled link, bandwidth utilization is defined to
      be the sum of the component link bandwidth
      utilizations.";
  }
}

grouping performance-metric-container {
  description
    "A container containing performance metric attributes.";
  container performance-metric-one-way {
    description
      "One-way link performance information in real time.";
    reference
      "RFC7471: OSPF Traffic Engineering (TE) Metric Extensions.
      RFC7810: IS-IS Traffic Engineering (TE) Metric Extensions.
      RFC7823: Performance-Based Path Selection for Explicitly
      Routed Label Switched Paths (LSPs) Using TE Metric
      Extensions";
    uses performance-metric-one-way-delay-loss;
    uses performance-metric-one-way-bandwidth;
  }
  container performance-metric-two-way {
    description
      "Two-way link performance information in real time.";
    reference
      "RFC7471: OSPF Traffic Engineering (TE) Metric Extensions.
      RFC7810: IS-IS Traffic Engineering (TE) Metric Extensions.
      RFC7823: Performance-Based Path Selection for Explicitly
      Routed Label Switched Paths (LSPs) Using TE Metric
      Extensions";
    uses performance-metric-two-way-delay-loss;
  }
}

/**
```



```
* TE tunnel generic groupings
**/
grouping explicit-route-hop {
  description
    "The explicit route subobject grouping";
  choice type {
    description
      "The explicit route subobject type";
    case num-unnum-hop {
      container num-unnum-hop {
        leaf node-id {
          type te-types:te-node-id;
          description
            "The identifier of a node in the TE topology.";
        }
        leaf link-tp-id {
          type te-types:te-tp-id;
          description
            "TE link termination point identifier. The combination
            of TE link ID and the TE node ID is used to identify an
            unnumbered TE link.";
        }
        leaf hop-type {
          type te-hop-type;
          description "strict or loose hop";
        }
        leaf direction {
          type te-link-direction;
          default INCOMING;
          description "Link ERO direction";
        }
      }
      description
        "Numbered and Unnumbered link/node explicit route
        subobject";
      reference
        "RFC3209: section 4.3 for EXPLICIT_ROUTE in RSVP-TE
        RFC3477: Signalling Unnumbered Links in RSVP-TE";
    }
  }
  case as-number {
    container as-number-hop {
      leaf as-number {
        type binary {
          length 16;
        }
        description "AS number";
      }
      leaf hop-type {
```



```
        type te-hop-type;
        description
            "strict or loose hop";
    }
    description
        "Autonomous System explicit route subobject";
}
}
case label {
    container label-hop {
        description "Label hop type";
        uses te-label;
    }
    description
        "The Label ERO subobject";
}
}
}

grouping record-route-subobject_state {
    description
        "The record route subobject grouping";
    leaf index {
        type uint32;
        description "RRO subobject index";
    }
    choice type {
        description
            "The record route subobject type";
        case numbered {
            leaf address {
                type te-types:te-tp-id;
                description
                    "Numbered link TE termination point address.";
            }
            leaf ip-flags {
                type binary {
                    length 8;
                }
                description
                    "RRO IP address sub-object flags";
                reference "RFC3209";
            }
        }
    }
    case unnumbered {
        leaf node-id {
            type te-types:te-node-id;
            description
```



```
        "The identifier of a node in the TE topology.";
    }
    leaf link-tp-id {
        type te-types:te-tp-id;
        description
            "TE link termination point identifier, used
            together with te-node-id to identify the
            link termination point";
    }
    description
        "Unnumbered link record route subobject";
    reference
        "RFC3477: Signalling Unnumbered Links in
        RSVP-TE";
}
case label {
    container label-hop {
        description "Label hop type";
        uses te-label;
        leaf label-flags {
            type binary {
                length 8;
            }
            description
                "Label sub-object flags";
            reference "RFC3209";
        }
    }
    description
        "The Label RRO subobject";
}
}
}

grouping label-restriction-info {
    description "Label set item info";
    leaf restriction {
        type enumeration {
            enum inclusive {
                description "The label or label range is inclusive.";
            }
            enum exclusive {
                description "The label or label range is exclusive.";
            }
        }
    }
    description
        "Whether the list item is inclusive or exclusive.";
}
```



```
leaf index {
  type uint32;
  description
    "Then index of the label restriction list entry.";
}
container label-start {
  must "not(..../label-end/te-label/direction) or "
    + "not(te-label/direction) "
    + "or ../label-end/te-label/direction = te-label/direction" {
    error-message
      "label-start and label-end must have the same direction.";
  }
  description
    "This is the starting label if a label range is specified.
    This is the label value if a single label is specified,
    in which case, attribute 'label-end' is not set.";
  uses te-label;
}
container label-end {
  must "not(..../label-end/te-label/direction) or "
    + "not(te-label/direction) "
    + "or ../label-end/te-label/direction = te-label/direction" {
    error-message
      "label-start and label-end must have the same direction.";
  }
  description
    "The ending label if a label range is specified;
    This attribute is not set, If a single label is
    specified.";
  uses te-label;
}
container label-step {
  description
    "The step increment between labels in the label range.
    The label start/end values will have to be consistent
    with the sign of label step. For example,
    label-start < label-end enforces label-step > 0
    label-start > label-end enforces label-step < 0";
  choice technology {
    default generic;
    description
      "Data plane technology type.";
    case generic {
      leaf generic {
        type int32;
        default 1;
        description "Label range step";
      }
    }
  }
}
```



```
    }
  }
}
leaf range-bitmap {
  type binary;
  description
    "When there are gaps between label-start and label-end,
    this attribute is used to specify the positions
    of the used labels.";
}
}

grouping label-set-info {
  description
    "Grouping for List of label restrictions specifying what labels
    may or may not be used on a link connectivity.";
  container label-restrictions {
    description
      "The label restrictions container";
    list label-restriction {
      key "index";
      description
        "The absence of label-set implies that all labels are
        acceptable; otherwise only restricted labels are
        available.";
      reference
        "RFC7579: General Network Element Constraint Encoding
        for GMPLS-Controlled Networks";
      uses label-restriction-info;
    }
  }
}

grouping optimizations_config {
  description "Optimization metrics configuration grouping";
  leaf metric-type {
    type identityref {
      base te-types:path-metric-type;
    }
    description "TE path metric type";
  }
  leaf weight {
    type uint8;
    description "TE path metric normalization weight";
  }
  container explicit-route-exclude-objects {
    when "../metric-type = " +
      "'te-types:path-metric-optimize-excludes'";
  }
}
```



```
    description
      "Container for the exclude route object list";
    uses path-route-exclude-objects;
  }
  container explicit-route-include-objects {
    when "../metric-type = " +
      "'te-types:path-metric-optimize-includes'";
    description
      "Container for the include route object list";
    uses path-route-include-objects;
  }
}

grouping common-constraints_config {
  description
    "Common constraints grouping that can be set on
    a constraint set or directly on the tunnel";

  uses te-types:te-bandwidth {
    description
      "A requested bandwidth to use for path computation";
  }

  leaf setup-priority {
    type uint8 {
      range "0..7";
    }
    description
      "TE LSP requested setup priority";
    reference RFC3209;
  }
  leaf hold-priority {
    type uint8 {
      range "0..7";
    }
    description
      "TE LSP requested hold priority";
    reference RFC3209;
  }
  leaf signaling-type {
    type identityref {
      base te-types:path-signaling-type;
    }
    description "TE tunnel path signaling type";
  }
}

grouping tunnel-constraints_config {
```



```
    description
      "Tunnel constraints grouping that can be set on
      a constraint set or directly on the tunnel";
    uses te-types:te-topology-identifier;
    uses te-types:common-constraints_config;
  }

  grouping path-metrics-bounds_config {
    description "TE path metric bounds grouping";
    leaf metric-type {
      type identityref {
        base te-types:path-metric-type;
      }
      description "TE path metric type";
    }
    leaf upper-bound {
      type uint64;
      description "Upper bound on end-to-end TE path metric";
    }
  }

  grouping path-objective-function_config {
    description "Optimization metrics configuration grouping";
    leaf objective-function-type {
      type identityref {
        base te-types:objective-function-type;
      }
      description
        "Objective function entry";
    }
  }

  grouping path-route-objects {
    description
      "List of EROs to be included or excluded when performing
      the path computation.";
    container explicit-route-objects {
      description
        "Container for the exclude route object list";
      list route-object-exclude-always {
        key index;
        description
          "List of explicit route objects to always exclude
          from path computation";
        leaf index {
          type uint32;
          description "ERO subobject index";
        }
      }
    }
  }
```



```
    uses te-types:explicit-route-hop;
  }
  list route-object-include-exclude {
    key index;
    description
      "List of explicit route objects to include or
      exclude in path computation";
    leaf explicit-route-usage {
      type identityref {
        base te-types:route-usage-type;
      }
      description "Explicit-route usage.";
    }
    leaf index {
      type uint32;
      description "ERO subobject index";
    }
    uses te-types:explicit-route-hop {
      augment "type" {
        case srlg {
          container srlg {
            description "SRLG container";
            leaf srlg {
              type uint32;
              description "SRLG value";
            }
          }
        }
        description "An SRLG value to be included or excluded";
      }
      description
        "Augmentation to generic explicit route for SRLG
        exclusion";
    }
  }
}

grouping path-route-include-objects {
  description
    "List of EROs to be included when performing
    the path computation.";
  list route-object-include-object {
    key index;
    description
      "List of explicit route objects to be included
      in path computation";
    leaf index {
```



```
        type uint32;
        description "ERO subobject index";
    }
    uses te-types:explicit-route-hop;
}
}

grouping path-route-exclude-objects {
    description
        "List of EROs to be included when performing
        the path computation.";
    list route-object-exclude-object {
        key index;
        description
            "List of explicit route objects to be excluded
            in path computation";
        leaf index {
            type uint32;
            description "ERO subobject index";
        }
        uses te-types:explicit-route-hop {
            augment "type" {
                case srlg {
                    container srlg {
                        description "SRLG container";
                        leaf srlg {
                            type uint32;
                            description "SRLG value";
                        }
                    }
                }
                description "An SRLG value to be included or excluded";
            }
            description
                "Augmentation to generic explicit route for SRLG exclusion";
        }
    }
}

grouping generic-path-metric-bounds {
    description "TE path metric bounds grouping";
    container path-metric-bounds {
        description "TE path metric bounds container";
        list path-metric-bound {
            key metric-type;
            description "List of TE path metric bounds";
            uses path-metrics-bounds_config;
        }
    }
}
```



```
    }
  }

  grouping generic-path-optimization {
    description "TE generic path optimization grouping";

    container optimizations {
      description
        "The objective function container that includes
        attributes to impose when computing a TE path";

      choice algorithm {
        description "Optimizations algorithm.";
        case metric {
          if-feature path-optimization-metric;
          /* Optimize by metric */
          list optimization-metric {
            key "metric-type";
            description "TE path metric type";
            uses optimizations_config;
          }
          /* Tiebreakers */
          container tiebreakers {
            description
              "The list of tiebreaker criterion to apply
              on an equally favored set of paths to pick best";
            list tiebreaker {
              key "tiebreaker-type";
              description
                "The list of tiebreaker criterion to apply
                on an equally favored set of paths to pick best";
              leaf tiebreaker-type {
                type identityref {
                  base te-types:path-metric-type;
                }
                description "The objective function";
              }
            }
          }
        }
      }
    }
  }

  case objective-function {
    if-feature path-optimization-objective-function;
    /* Objective functions */
    container objective-function {
      description
        "The objective function container that includes
        attributes to impose when computing a TE path";
      uses path-objective-function_config;
    }
  }
}
```



```
    }
  }
}

grouping generic-path-affinities {
  description
    "Path affinities grouping";
  container path-affinities-values {
    description
      "Path affinities values representation";
    list path-affinities-value {
      key "usage";
      description
        "List of named affinity constraints";
      leaf usage {
        type identityref {
          base resource-affinities-type;
        }
        description "Affinities usage";
      }
      leaf value {
        type admin-groups;
        description "Affinity value";
      }
    }
  }
}

container path-affinity-names {
  description
    "Path affinities named representation style";
  list path-affinity-name {
    key "usage";
    description "List of named affinity constraints";
    leaf usage {
      type identityref {
        base te-types:resource-affinities-type;
      }
      description "Affinities usage";
    }
    list affinity-name {
      key "name";
      leaf name {
        type string;
        description "Affinity name";
      }
      description "List of named affinities";
    }
  }
}
```



```
    }
  }
}

grouping generic-path-srlgs {
  description
    "Path SRLG grouping";
  container path-srlgs-values {
    description
      "Path SRLG properties container";
    leaf usage {
      type identityref {
        base te-types:route-exclude-srlg;
      }
      description "SRLG usage";
    }
    leaf-list values {
      type srlg;
      description "SRLG value";
    }
  }
  container path-srlgs-names {
    description "Container for named SRLG list";
    list path-srlgs-name {
      key "usage";
      description "List of named SRLGs";
      leaf usage {
        type identityref {
          base te-types:route-exclude-srlg;
        }
        description "SRLG usage";
      }
      list srlg-name {
        key "name";
        leaf name {
          type string;
          description "The SRLG name";
        }
        description "List named SRLGs";
      }
    }
  }
}

grouping generic-path-disjointness {
  description "Path disjointness grouping";
  leaf disjointness {
    type te-types:te-path-disjointness;
```



```
    description
      "The type of resource disjointness.
      Under primary path, disjointness level applies to
      all secondary LSPs. Under secondary, disjointness
      level overrides the one under primary";
  }
}

grouping common-path-constraints-attributes {
  description
    "Common path constraints configuration grouping";
  uses common-constraints_config;
  uses generic-path-metric-bounds;
  uses generic-path-affinities;
  uses generic-path-srlgs;
}

grouping generic-path-constraints {
  description
    "Global named path constraints configuration
    grouping";
  container path-constraints {
    description "TE named path constraints container";
    uses common-path-constraints-attributes;
    uses generic-path-disjointness;
  }
}

grouping generic-path-properties {
  description "TE generic path properties grouping";
  container path-properties {
    config false;
    description "The TE path properties";
    list path-metric {
      key metric-type;
      description "TE path metric type";
      leaf metric-type {
        type identityref {
          base te-types:path-metric-type;
        }
        description "TE path metric type";
      }
      leaf accumulative-value {
        type uint64;
        description "TE path metric accumulative value";
      }
    }
  }
  uses generic-path-affinities;
}
```



```

uses generic-path-srlgs;
container path-route-objects {
  description
    "Container for the list of route objects either returned by
    the computation engine or actually used by an LSP";
  list path-route-object {
    key index;
    description
      "List of route objects either returned by the computation
      engine or actually used by an LSP";
    leaf index {
      type uint32;
      description "ERO subobject index";
    }
    uses explicit-route-hop;
  }
}
}
}
}
}
<CODE ENDS>

```

Figure 1: TE basic types YANG module

4. IETF MPLS TE Types YANG Module

```

<CODE BEGINS> file "ietf-te-mpls-types@2018-09-13.yang"
module ietf-te-mpls-types {

  namespace "urn:ietf:params:xml:ns:yang:ietf-te-mpls-types";

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

  organization
    "IETF TEAS Working Group";

  contact
    "WG Web:  <http://tools.ietf.org/wg/teas/>
    WG List:  <mailto:teas@ietf.org>

    WG Chair: Lou Berger
               <mailto:lberger@labn.net>

    WG Chair: Vishnu Pavan Beeram
               <mailto:vbeeram@juniper.net>

    Editor:   Tarek Saad

```


<mailto:tsaad@cisco.com>

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

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

Editor: Himanshu Shah
<mailto:hshah@ciena.com>

Editor: Xufeng Liu
<mailto:xufeng.liu.ietf@gmail.com>

Editor: Igor Bryskin
<mailto:Igor.Bryskin@huawei.com>

Editor: Young Lee
<mailto:leeyoung@huawei.com>";

description

"This module contains a collection of generally
useful MPLS TE specific YANG data type definitions.";

revision "2018-09-13" {
 description "Latest revision of TE MPLS types";
 reference "[RFC3209](#)";
}

identity backup-protection-type {
 description
 "Base identity for backup protection type";
}

identity backup-protection-link {
 base backup-protection-type;
 description
 "backup provides link protection only";
}

identity backup-protection-node-link {
 base backup-protection-type;
 description
 "backup offers node (preferred) or link protection";
}

identity bc-model-type {
 description


```
    "Base identity for Diffserv-TE bandwidth constraint
    model type";
}

identity bc-model-rdm {
    base bc-model-type;
    description
        "Russian Doll bandwidth constraint model type.";
}

identity bc-model-mam {
    base bc-model-type;
    description
        "Maximum Allocation bandwidth constraint
        model type.";
}

identity bc-model-mar {
    base bc-model-type;
    description
        "Maximum Allocation with Reservation
        bandwidth constraint model type.";
}

typedef bandwidth-kbps {
    type uint64;
    units "Kbps";
    description
        "Bandwidth values expressed in kilobits per second";
}

typedef bandwidth-mbps {
    type uint64;
    units "Mbps";
    description
        "Bandwidth values expressed in megabits per second";
}

typedef bandwidth-gbps {
    type uint64;
    units "Gbps";
    description
        "Bandwidth values expressed in gigabits per second";
}

typedef te-bandwidth-type {
    type enumeration {
        enum SPECIFIED {
```



```
        description
            "Bandwidth is explicitly specified";
    }
    enum AUTO {
        description
            "Bandwidth is automatically computed";
    }
}
description
    "enumerated type for specifying whether bandwidth is
    explicitly specified or automatically computed";
}

typedef bfd-type {
    type enumeration {
        enum classical {
            description "BFD classical session type.";
        }
        enum seamless {
            description "BFD seamless session type.";
        }
    }
}
default "classical";
description
    "Type of BFD session";
}

typedef bfd-encap-mode-type {
    type enumeration {
        enum gal {
            description
                "BFD with GAL mode";
        }
        enum ip {
            description
                "BFD with IP mode";
        }
    }
}
default ip;
description
    "Possible BFD transport modes when running over TE
    LSPs.";
}
}
<CODE ENDS>
```

Figure 2: TE MPLS types YANG module

5. 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.

URI: urn:ietf:params:xml:ns:yang:ietf-te-types XML: N/A, the requested URI is an XML namespace.

URI: urn:ietf:params:xml:ns:yang:ietf-te-mpls-types 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-types namespace: urn:ietf:params:xml:ns:yang:ietf-te-types prefix: ietf-te-types reference: [RFC3209](#)

name: ietf-te-mpls-types namespace: urn:ietf:params:xml:ns:yang:ietf-te-mpls-types prefix: ietf-te-mpls-types reference: [RFC3209](#)

6. 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.

7. Acknowledgement

The authors would like to thank the members of the multi-vendor YANG design team who are involved in the definition of these data types.

The authors would also like to thank Loa Andersson, Lou Berger, Sergio Belotti, Italo Busi, Carlo Perocchio, Francesco Lazzeri, Aihua Guo, Dhruv Dhody, Anurag Sharma, and Xian Zhang for their comments and providing valuable feedback on this document.

8. Normative References

[I-D.ietf-teas-yang-rsvp]
Beeram, V., Saad, T., Gandhi, R., Liu, X., Bryskin, I., and H. Shah, "A YANG Data Model for Resource Reservation Protocol (RSVP)", [draft-ietf-teas-yang-rsvp-09](#) (work in progress), May 2018.

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

Saad, T., Gandhi, R., Liu, X., Beeram, V., Shah, H., and I. Bryskin, "A YANG Data Model for Traffic Engineering Tunnels and Interfaces", [draft-ietf-teas-yang-te-16](#) (work in progress), July 2018.

[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>>.

[RFC6991] Schoenwaelder, J., Ed., "Common YANG Data Types", [RFC 6991](#), DOI 10.17487/RFC6991, July 2013, <<https://www.rfc-editor.org/info/rfc6991>>.

[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>>.

[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, <<https://www.rfc-editor.org/info/rfc8294>>.

[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>>.

Authors' Addresses

Tarek Saad (editor)
Cisco Systems Inc

Email: tsaad@cisco.com

Rakesh Gandhi
Cisco Systems Inc

Email: rgandhi@cisco.com

Xufeng Liu
Volta Networks

Email: xufeng.liu.ietf@gmail.com

Vishnu Pavan Beeram
Juniper Networks

Email: vbeeram@juniper.net

Himanshu Shah
Ciena

Email: hshah@ciena.com

Igor Bryskin
Huawei Technologies

Email: Igor.Bryskin@huawei.com

Young Lee
Huawei Technologies

Email: leeyoung@huawei.com

