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

V. P. Beeram
K. Kompella
Juniper Networks

20 October 2025

A YANG Data Model for Multipath Traffic Engineering Directed Acyclic Graph (MPTED) Tunnels and Junctions
draft-beeram-teas-yang-mpted-01

Abstract

This document defines a YANG data model for representing, retrieving, and manipulating Multipath Traffic Engineering Directed Acyclic Graph (MPTED) Tunnels and Junctions. The model includes two YANG modules, one for managing MPTED Tunnels on an MPTED tunnel originator node and the other for managing MPTED Junctions on an MPTED junction node.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

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

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

This Internet-Draft will expire on 23 April 2026.

Table of Contents

1. Introduction

The notions of a Multipath Traffic Engineering Directed Acyclic Graph (MPTED) tunnel and an MPTED junction are introduced in [I-D.draft-kompella-teas-mpte]. An MPTED tunnel is a Traffic Engineering (TE) construct that contains a constrained set of paths representing an optimized Directed Acyclic Graph (DAG) from one or more ingresses to one or more egresses. The paths that make up an MPTED tunnel traverse a set of junction nodes. An MPTED junction refers to the construct associated with the MPTED tunnel at each junction node and constitutes a set of previous-hops and a set of next-hops over which traffic is load-balanced in a weighted fashion. Provisioning an MPTED tunnel in a TE network involves provisioning MPTED junction state at each junction node.

An MPTED tunnel is instantiated and managed on a tunnel originator node, while an MPTED junction is instantiated and managed on a junction node. A tunnel originator node MAY also be a junction node.

This document defines a YANG data model for representing, retrieving, and manipulating Multipath Traffic Engineering Directed Acyclic Graph (MPTED) Tunnels and Junctions. The model includes two YANG modules, one for managing MPTED Tunnels on a tunnel originator node and the other for managing MPTED Junctions on a junction node.

The YANG modules discussed in this version of the document are scoped to MPLS MPTED tunnels and junctions with signaled label switching. The coverage for other types of MPTED tunnels and junctions will be added in later versions.

1.1. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

1.2. Terminology

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

The reader is expected to be familiar with the terminology used in [I-D.draft-kompella-teas-mpte].

1.3. Tree Diagram

A simplified graphical representation of the data model is used in Appendix A and Appendix B of this document. The meaning of the symbols in these diagrams is defined in [RFC8340].

2. MPTED YANG Module

2.1. Model Structure

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

```
module: ietf-mpted

  augment /te:te:
    +--rw mpted-tunnels
       +--rw tunnel* [originator identifier]
          +--rw originator                        inet:ip-address
          +--rw identifier                        uint32
          +  ..
          +--ro instances
             +--ro instance* [version]
                +--ro version      uint16
                +  ..
                +--ro junctions
                   +--ro junction* [node-id]
                      +--ro node-id                inet:ip-address
                      +  ..
                      +--ro jct-instances
                         +--ro jct-instance* [jct-version]
                            +--ro jct-version            uint16
                            +  ..
                            +--ro phops
                            |  +--ro phop* [hop-address hop-index]
                            |     +--ro hop-address
                            |     |       inet:ip-address
                            |     +--ro hop-index      uint32
                            |     +  ..
                            +--ro nhops
                               +--ro nhop* [hop-address hop-index]
                                  +--ro hop-address
                                  |       inet:ip-address
                                  +--ro hop-index             uint32
                                  |     +  ..
```

Figure 1: MPTED YANG Structure

The top-level 'te' container is [I-D.draft-ietf-teas-yang-te] is augmented with a set of MPTED tunnels. The 'mpted-tunnels' container carries a list of tunnel entries. Each tunnel entry includes the set of parameters required to produce a list of junctions that need to programmed in the network. Each tunnel entry may have more than one instance associated with it, where a unique version identifies each instance. Each tunnel instance has a list of junctions associated with it. Each junction entry may have more than one instance (jct-instance) associated with it, where a unique junction-version identifies each instance. Each junction instance entry consists of the set of previous-hops ('phops' container) and next-hops ('nhops' container) associated with the given junction version.

2.2. YANG Code

```
<CODE BEGINS> file "ietf-mpted@2025-10-20.yang"
module ietf-mpted {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-mpted";
  prefix mpted;

  import ietf-te {
    prefix te;
    reference
      "draft-ietf-teas-yang-te: A YANG Data Model for Traffic
       Engineering Tunnels and Interfaces";
  }
  import ietf-routing-types {
    prefix rt-types;
    reference
      "RFC8294: Common YANG Data Types for the Routing Area";
  }
  import ietf-inet-types {
    prefix inet;
    reference
      "RFC6991: Common YANG Data Types";
  }

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

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

     Editor:   Kireeti Kompella
               <mailto:kireeti@juniper.net>";
  description
    "This module defines a YANG data model for Multipath Traffic
     Engineering Directed Acyclic Graph (MPTED) Tunnels. It is
     intended to be used as part of the IETF TEAS framework to
     enable the configuration and management of MPTED Tunnels on
     an MPTED Tunnel Originator.

     Copyright (c) 2025 IETF Trust and the persons identified as
     authors of the code. All rights reserved.

     Redistribution and use in source and binary forms, with or
```

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

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

    revision 2025-10-20 {
      description
        "Initial revision.
         This revision introduces the YANG data model for
         Multipath Traffic Engineering Directed Acyclic Graph
         (MPTED) Tunnels.";
      reference
        "RFC XXXX: A YANG Data Model for Multipath Traffic
         Engineering Directed Acyclic Graph (MPTED) Tunnels and
         Junctions";
    }

    identity mpted-tunnel-type {
      description
        "Base identity of MPTED tunnel type.";
    }

    identity mpted-tunnel-type-mpls-siglab {
      base mpted-tunnel-type;
      description
        "MPLS MPTED Tunnel with signaled label switching.";
    }

    identity mpted-signaling-type {
      description
        "Base identity for MPTED signaling type.";
    }

    identity mpted-signaling-rsvp {
      base mpted-signaling-type;
      description
        "MPTED RSVP signaling.";
    }

    identity mpted-signaling-pcep {
      base mpted-signaling-type;
      description
        "MPTED PCEP signaling.";
```

```
}

identity mpted-signaling-bgp {
  base mpted-signaling-type;
  description
    "MPTED BGP signaling.";
}

identity mpted-signaling-grpc {
  base mpted-signaling-type;
  description
    "MPTED gRPC signaling.";
}

identity mpted-tunnel-status {
  description
    "Base identity for MPTED Tunnel status.";
}

identity mpted-tunnel-up {
  base mpted-tunnel-status;
  description
    "MPTED Tunnel Up.";
}

identity mpted-tunnel-down {
  base mpted-tunnel-status;
  description
    "MPTED Tunnel Down.";
}

identity mpted-tunnel-degraded {
  base mpted-tunnel-status;
  description
    "MPTED Tunnel Degraded.";
}

identity mpted-junction-status {
  description
    "Base identity for MPTED JUNCTION status.";
}

identity mpted-junction-up {
  base mpted-junction-status;
  description
    "MPTED JUNCTION Up.";
}
```

```
identity mpted-junction-down {
  base mpted-junction-status;
  description
    "MPTED JUNCTION Down.";
}

identity mpted-junction-degraded {
  base mpted-junction-status;
  description
    "MPTED JUNCTION Degraded.";
}

identity mpted-jct-phop-status {
  description
    "Base identity for MPTED JCT-PHOP status.";
}

identity mpted-jct-phop-up {
  base mpted-jct-phop-status;
  description
    "MPTED JCT-PHOP Up.";
}

identity mpted-jct-phop-down {
  base mpted-jct-phop-status;
  description
    "MPTED JCT-PHOP Down.";
}

identity mpted-jct-nhop-status {
  description
    "Base identity for MPTED JCT-NHOP status.";
}

identity mpted-jct-nhop-up {
  base mpted-jct-nhop-status;
  description
    "MPTED JCT-NHOP Up.";
}

identity mpted-jct-nhop-down {
  base mpted-jct-nhop-status;
  description
    "MPTED JCT-NHOP Down.";
}

identity mpted-jct-nhop-degraded {
  base mpted-jct-nhop-status;
```

```
      description
        "MPTED JCT-NHOP Degraded.";
    }

    identity mpted-opt-metric-type {
      description
        "Base identity of MPTED optimization metric type.";
    }

    identity mpted-opt-metric-te {
      base mpted-opt-metric-type;
      description
        "TE Metric.";
    }

    identity mpted-opt-metric-igp {
      base mpted-opt-metric-type;
      description
        "IGP Metric.";
    }

    identity mpted-opt-metric-delay {
      base mpted-opt-metric-type;
      description
        "Delay Metric.";
    }

    identity mpted-bw-spec-type {
      description
        "Base identity for MPTED bandwidth specification type.";
    }

    identity mpted-bw-spec-explicit {
      base mpted-bw-spec-type;
      description
        "Explicit MPTED bandwidth specification type.";
    }

    identity mpted-bw-spec-auto {
      base mpted-bw-spec-type;
      description
        "Automatic MPTED bandwidth specification type.";
    }

    grouping mpted-ingress {
      description
        "Grouping for MPTED ingress information.";
      leaf ingress-id {
```

```
        type inet:ip-address;
        description
          "IP-address of the ingress node.";
      }
      leaf set-bandwidth {
        when "/te:te/mpted-tunnels/tunnel/bandwidth-spec-type = "
           + "'mpted-bw-spec-explicit'" {
          description
            "Applicable only when bandwidth specification
             type is explicit.";
        }
        type uint64;
        description
          "Explicitly specified bandwidth at the ingress of
           the MPTED tunnel.";
      }
      container auto-bandwidth {
        when "/te:te/mpted-tunnels/tunnel/bandwidth-spec-type = "
           + "'mpted-bw-spec-auto'" {
          description
            "Applicable only when bandwidth specification
             type is automatic.";
        }
        description
          "Container for auto-bandwidth parameters.";
        uses mpted-ingress-autobw-config;
        container overflow {
          description
            "Autobandwidth overflow parameters.";
          uses mpted-autobw-overflow-config;
        }
        container underflow {
          description
            "Autobandwidth underflow parameters.";
          uses mpted-autobw-underflow-config;
        }
      }
    }

    grouping mpted-ingress-autobw-config {
      description
        "MPTED tunnel ingress autobandwidth parameters";
      leaf min-bw {
        type uint64;
        description
          "Minimum bandwidth at the ingress of the
           autobandwidth enabled MPTED tunnel.";
      }
```

```
        leaf max-bw {
          type uint64;
          description
            "Maximum bandwidth at the ingress of the
             autobandwidth enabled MPTED tunnel";
        }
        leaf adjust-threshold {
          type rt-types:percentage;
          description
            "Bandwidth demand at the ingress for the current
             bandwidth adjustment interval is compared to the
             current bandwidth allocation for the ingress of
             the MPTED tunnel. If the percentage difference
             in bandwidth is greater than or equal to the percentage
             specified by this leaf, the bandwidth allocation
             for the ingress is adjusted to the current bandwidth
             demand.";
        }
      }

      grouping mpted-autobw-overflow-config {
        description
          "MPTED Tunnel ingress autobandwidth overflow parameters.";
        leaf enabled {
          type boolean;
          default "false";
          description
            "Enables autobandwidth overflow adjustment on the
             MPTED tunnel ingress";
        }
        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 mpted-autobw-underflow-config {
        description
          "MPTED Tunnel ingress autobandwidth underflow parameters.";
        leaf enabled {
```

```
        type boolean;
        default "false";
        description
          "Enables autobandwidth underflow adjustment on the
           MPTED tunnel ingress";
      }
      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 tunnel-status-info {
      description
        "Grouping for tunnel status information.";
      leaf current-version {
        type uint16;
        config false;
        description
          "Current version of the MPTED tunnel.";
      }
      leaf status {
        type identityref {
          base mpted-tunnel-status;
        }
        config false;
        description
          "Status of the MPTED tunnel";
      }
    }

    grouping tunnel-instance-status-info {
      description
        "Grouping for tunnel instance status information.";
      leaf version {
        type uint16;
        config false;
        description
          "Identifier of the MPTED tunnel instance.";
      }
```

```
      leaf status {
        type identityref {
          base mpted-tunnel-status;
        }
        config false;
        description
          "Status of the MPTED tunnel instance";
      }
    }

    grouping junction-info {
      description
        "Grouping for junction information.";
      leaf node-id {
        type inet:ip-address;
        description
          "IP-address of the junction node.";
      }
      leaf current_jct_version {
        type uint16;
        description
          "Current instance identifier of the MPTED tunnel
           junction.";
      }
      leaf bandwidth-requested {
        type uint64;
        description
          "Amount of bandwidth requested for the junction.";
      }
    }

    grouping junction-status-info {
      description
        "Grouping for junction status information.";
      uses junction-info;
      leaf mtu {
        type uint16;
        description
          "MTU at the junction node. This is the smallest MTU value
           among the MTUs received from all JCT-NHOPs.";
      }
      leaf status {
        type identityref {
          base mpted-junction-status;
        }
        description
          "Status of the JUNCTION.";
      }
```

```
  }

  grouping junction-instance-info {
    description
      "Grouping for junction instance information.";
    leaf jct-version {
      type uint16;
      description
        "Instance identifier of the MPTED junction.";
    }
    leaf bandwidth-requested {
      type uint64;
      description
        "Amount of bandwidth requested for the junction.";
    }
  }

  grouping junction-instance-status-info {
    description
      "Grouping for junction status information.";
    uses junction-instance-info;
    leaf mtu {
      type uint16;
      description
        "MTU at the junction node. This is the smallest MTU value
         among the MTUs received from all JCT-NHOPs.";
    }
    leaf status {
      type identityref {
        base mpted-junction-status;
      }
      description
        "Status of the JUNCTION instance.";
    }
  }

  grouping mpted-jct-hop-info {
    description
      "Grouping for junction hop information.";
    leaf hop-address {
      type inet:ip-address;
      description
        "IP-address of the peer.";
    }
    leaf hop-index {
      type uint32;
      description
        "Index of the peer interface.";
```

```
      }
    }

    grouping mpted-jct-phop-status-info {
      description
        "Grouping for junction phop status information.";
      uses mpted-jct-hop-info;
      leaf in-label {
        type rt-types:mpls-label;
        description
          "Label used by incoming packets on the JCT-PHOP.";
      }
      leaf status {
        type identityref {
          base mpted-jct-phop-status;
        }
        description
          "Status of the JCT-PHOP.";
      }
    }

    grouping mpted-jct-nhop-info {
      description
        "Grouping for junction nhop information.";
      uses mpted-jct-hop-info;
      leaf load-share {
        type uint16;
        description
          "Relative load share.";
      }
    }

    grouping mpted-jct-nhop-status-info {
      description
        "Grouping for junction nhop status information.";
      uses mpted-jct-nhop-info;
      leaf bandwidth-reserved {
        type uint64;
        description
          "Amount of bandwidth reserved on the TE link associated
           with the JCT-NHOP.";
      }
      leaf out-label {
        type rt-types:mpls-label;
        description
          "Label used by outgoing packets on the JCT-NHOP.";
      }
      leaf status {
```

```
        type identityref {
          base mpted-jct-nhop-status;
        }
        description
          "Status of the JCT-NHOP.";
      }
    }

    augment "/te:te" {
      description
        "Augmentation for MPTED tunnels.";
      container mpted-tunnels {
        description
          "Container for MPTED tunnels.";
        list tunnel {
          key "originator identifier";
          unique "name";
          description
            "MPTED tunnel entry.";
          leaf originator {
            type inet:ip-address;
            description
              "IP address of the MPTED tunnel originator.";
          }
          leaf identifier {
            type uint32;
            description
              "Identifier of the MPTED tunnel.";
          }
          leaf name {
            type string;
            description
              "Name of the MPTED tunnel.";
          }
          container ingresses {
            description
              "Container for list of MPTED ingresses.";
            list ingress {
              key "ingress-id";
              description
                "MPTED ingress entry.";
              uses mpted-ingress;
            }
          }
          leaf-list egress {
            type inet:ip-address;
            description
              "List of egress nodes associated with the MPTED
```

```
            tunnel.";
        }
        leaf type {
          type identityref {
            base mpted-tunnel-type;
          }
          description
            "Type of tunnel.";
        }
        leaf setup-priority {
          type uint8 {
            range "0..7";
          }
          default "7";
          description
            "The priority of the MPTED tunnel with respect to
             taking resources, in the range of 0 to 7.  The
             value 0 is the highest priority.";
        }
        leaf hold-priority {
          type uint8 {
            range "0..7";
          }
          default "7";
          description
            "The priority of the MPTED tunnel with respect to
             holding resources, in the range of 0 to 7.  The
             value 0 is the highest priority.";
        }
        leaf optimization-metric {
          type identityref {
            base mpted-opt-metric-type;
          }
          description
            "Metric type for which the DAG is optimized.";
        }
        container resource-affinities {
          description
            "Resource-affinity constraints.";
          leaf-list exclude-any {
            type string;
            description
              "Exclude the link if it has any of these
               affinities.";
          }
          leaf-list include-any {
            type string;
            description
```

```
            "Include the link if it has any of these
             affinities";
          }
          leaf-list include-all {
            type string;
            description
              "Include the link if it has all these affinities";
          }
        }
        leaf bandwidth-spec-type {
          type identityref {
            base mpted-bw-spec-type;
          }
          description
            "The method used for setting the bandwidth on the
             tunnel, either explicitly specified or automatically
             derived.";
        }
        leaf auto-bandwidth-adjust-interval {
          when "../bandwidth-spec-type = "
             + "'mpted-bw-spec-auto'" {
            description
              "Applicable only when bandwidth specification
               type is automatic.";
          }
          type uint32;
          description
            "Time in seconds between autobandwidth adjustments.";
        }
        leaf mpted-computer {
          type inet:ip-address;
          description
            "IP-address of the MPTED computer.";
        }
        leaf signaling-type {
          type identityref {
            base mpted-signaling-type;
          }
          description
            "MPTED tunnel signaling type.";
        }
        leaf signaling-source {
          type inet:ip-address;
          description
            "IP-address of the MPTED tunnel signaling source.";
        }
        uses tunnel-status-info;
        container instances {
```

```
            config false;
            description
              "Container for list of instances of the MPTED tunnel.";
            list instance {
              key "version";
              description
                "MPTED tunnel instance entry.";
              uses tunnel-instance-status-info;
              container junctions {
                description
                  "Container for list of JUNCTIONs associated
                   with the MPTED tunnel instance.";
                list junction {
                  key "node-id";
                  description
                    "JUNCTION entry.";
                  uses junction-status-info;
                  container jct-instances {
                    description
                      "Container for list of instances of the MPTED
                       junction.";
                    list jct-instance {
                      key "jct-version";
                      description
                        "MPTED junction instance entry.";
                      uses junction-instance-status-info;
                      container phops {
                        description
                          "Container for associated JCT-PHOPs.";
                        list phop {
                          key "hop-address hop-index";
                          description
                            "JCT-PHOP entry.";
                          uses mpted-jct-phop-status-info;
                        }
                      }
                      container nhops {
                        description
                          "Container for associated JCT-NHOPs.";
                        list nhop {
                          key "hop-address hop-index";
                          description
                            "JCT-NHOP entry.";
                          uses mpted-jct-nhop-status-info;
                        }
                      }
                    }
                  }
```

```
                }
              }
            }
          }
        }
      }
    }
  }
  <CODE ENDS>
```

Figure 2: MPTED YANG module

3. MPTED-JCT YANG Module

3.1. Model Structure

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

module: ietf-mpted-jct

```
  augment /te:te:
    +--rw mpted-junctions
       +--rw junction* [node-id originator tnl-id tnl-vers sig-src]
          +--rw node-id                inet:ip-address
          +--rw originator             inet:ip-address
          +--rw tnl-id                 uint32
          +--rw tnl-vers               uint16
          +--rw sig-src                inet:ip-address
          +  ..
          +--rw jct-instances
             +--rw jct-instance* [jct-version]
                +--rw jct-version            uint16
                +  ..
                +--rw phops
                |  +--rw phop* [hop-address hop-index]
                |     +--rw hop-address    inet:ip-address
                |     +--rw hop-index      uint32
                |     +  ..
                +--rw nhops
                   +--rw nhop* [hop-address hop-index]
                      +--rw hop-address           inet:ip-address
                      +--rw hop-index             uint32
                      |     +  ..
```

Figure 3: MPTED-JCT YANG Structure

The top-level 'te' container is [I-D.draft-ietf-teas-yang-te] is augmented with a set of MPTED junctions. The 'mpted-junctions' container carries a list of junction entries. Each junction entry may be associated with more than one junction instance. And each junction instance includes information about the associated set of previous-hops ('phops' container) and next-hops ('nhops' container) for the given junction version.

3.2. YANG Code

```
<CODE BEGINS> file "ietf-mpted-jct@2025-10-20.yang"
module ietf-mpted-jct {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-mpted-jct";
  prefix mpted-jct;

  import ietf-te {
    prefix te;
    reference
      "draft-ietf-teas-yang-te: A YANG Data Model for Traffic
       Engineering Tunnels and Interfaces";
  }
  import ietf-inet-types {
    prefix inet;
    reference
      "RFC6991: Common YANG Data Types";
  }
  import ietf-routing-types {
    prefix rt-types;
    reference
      "RFC8294: Common YANG Data Types for the Routing Area";
  }

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

     Editor:   Vishnu Pavan Beeram
               <mailto:vbeeram@juniper.net>";
  description
    "This module defines a YANG data model for Multipath Traffic
     Engineering Directed Acyclic Graph (MPTED) Junctions. It is
     intended to be used as part of the IETF TEAS framework to
     enable the configuration and management of MPTED Junctions on
     an MPTED junction node.
```

```
   Copyright (c) 2025 IETF Trust and the persons identified as
   authors of the code. All rights reserved.

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

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

 revision 2025-10-20 {
   description
     "Initial revision.
      This revision introduces the YANG data model for
      Multipath Traffic Engineering Directed Acyclic Graph
      (MPTED) Junctions.";
   reference
     "RFC XXXX: A YANG Data Model for Multipath Traffic
      Engineering Directed Acyclic Graph (MPTED) Tunnels and
      Junctions";
 }

 identity mpted-tunnel-type {
   description
     "Base identity of MPTED tunnel type.";
 }

 identity mpted-tunnel-type-mpls-siglab {
   base mpted-tunnel-type;
   description
     "MPLS MPTED Tunnel with signaled label switching.";
 }

 identity mpted-signaling-type {
   description
     "Base identity for MPTED signaling type.";
 }

 identity mpted-signaling-rsvp {
   base mpted-signaling-type;
   description
     "MPTED RSVP signaling.";
 }
```

```
identity mpted-signaling-pcep {
  base mpted-signaling-type;
  description
    "MPTED PCEP signaling.";
}

identity mpted-signaling-bgp {
  base mpted-signaling-type;
  description
    "MPTED BGP signaling.";
}

identity mpted-signaling-grpc {
  base mpted-signaling-type;
  description
    "MPTED gRPC signaling.";
}

identity mpted-junction-status {
  description
    "Base identity for MPTED JUNCTION status.";
}

identity mpted-junction-up {
  base mpted-junction-status;
  description
    "MPTED JUNCTION Up.";
}

identity mpted-junction-down {
  base mpted-junction-status;
  description
    "MPTED JUNCTION Down.";
}

identity mpted-junction-degraded {
  base mpted-junction-status;
  description
    "MPTED JUNCTION Degraded.";
}

identity mpted-jct-phop-status {
  description
    "Base identity for MPTED JCT-PHOP status.";
}

identity mpted-jct-phop-up {
  base mpted-jct-phop-status;
```

```
    description
      "MPTED JCT-PHOP Up.";
  }

  identity mpted-jct-phop-down {
    base mpted-jct-phop-status;
    description
      "MPTED JCT-PHOP Down.";
  }

  identity mpted-jct-nhop-status {
    description
      "Base identity for MPTED JCT-NHOP status.";
  }

  identity mpted-jct-nhop-up {
    base mpted-jct-nhop-status;
    description
      "MPTED JCT-NHOP Up.";
  }

  identity mpted-jct-nhop-down {
    base mpted-jct-nhop-status;
    description
      "MPTED JCT-NHOP Down.";
  }

  identity mpted-jct-nhop-degraded {
    base mpted-jct-nhop-status;
    description
      "MPTED JCT-NHOP Degraded.";
  }

  grouping mpted-jct-info {
    description
      "Grouping for junction information.";
    leaf current_jct_version {
      type uint16;
      description
        "Current instance identifier of the MPTED tunnel
         junction.";
    }
    leaf bandwidth-requested {
      type uint64;
      description
        "Amount of bandwidth requested for the junction.";
    }
  }
```

```
    grouping mpted-jct-status-info {
      description
        "Grouping for junction status information.";
      uses mpted-jct-info;
      leaf mtu {
        type uint16;
        config false;
        description
          "MTU at the junction node. This is the smallest MTU value
           among the MTUs received from all JCT-NHOPs.";
      }
      leaf status {
        type identityref {
          base mpted-junction-status;
        }
        config false;
        description
          "Status of the JUNCTION.";
      }
    }

    grouping mpted-jct-instance-info {
      description
        "Grouping for junction instance information.";
      leaf jct-version {
        type uint16;
        description
          "Instance identifier of the MPTED junction.";
      }
      leaf bandwidth-requested {
        type uint64;
        description
          "Amount of bandwidth requested for the junction.";
      }
    }

    grouping mpted-jct-instance-status-info {
      description
        "Grouping for junction instance status information.";
      uses mpted-jct-instance-info;
      leaf mtu {
        type uint16;
        config false;
        description
          "MTU at the junction node. This is the smallest MTU value
           among the MTUs received from all JCT-NHOPs.";
      }
      leaf status {
```

```
        type identityref {
          base mpted-junction-status;
        }
        config false;
        description
          "Status of the JUNCTION instance.";
      }
    }

    grouping mpted-jct-hop-info {
      description
        "Grouping for MPTED junction hop information.";
      leaf hop-address {
        type inet:ip-address;
        description
          "IP-address of the peer.";
      }
      leaf hop-index {
        type uint32;
        description
          "Index of the peer interface.";
      }
    }

    grouping mpted-jct-nhop-info {
      description
        "Grouping for MPTED junction next-hop information.";
      uses mpted-jct-hop-info;
      leaf load-share {
        type uint16;
        description
          "Relative load share.";
      }
    }

    augment "/te:te" {
      description
        "Augmentation for MPTED junctions.";
      container mpted-junctions {
        description
          "Container for MPTED junctions.";
        list junction {
          key "node-id originator tnl-id tnl-vers sig-src";
          unique "name";
          description
            "MPTED junction entry.";
          leaf node-id {
            type inet:ip-address;
```

```
        description
          "IP-address of the junction node.";
      }
      leaf originator {
        type inet:ip-address;
        description
          "IP address of the associated MPTED tunnel
           originator.";
      }
      leaf tnl-id {
        type uint32;
        description
          "Identifier of the associated MPTED tunnel.";
      }
      leaf tnl-vers {
        type uint16;
        description
          "Version of the MPTED tunnel.";
      }
      leaf sig-src {
        type inet:ip-address;
        description
          "IP-address of the MPTED tunnel signaling source.";
      }
      leaf name {
        type string;
        description
          "Name of the associated MPTED tunnel.";
      }
      leaf-list ingress {
        type inet:ip-address;
        description
          "List of ingress nodes of the associated MPTED
           tunnel.";
      }
      leaf-list egress {
        type inet:ip-address;
        description
          "List of egress nodes of the associated MPTED
           tunnel.";
      }
      leaf type {
        type identityref {
          base mpted-tunnel-type;
        }
        description
          "Type of associated tunnel.";
      }
```

```
            leaf setup-priority {
              type uint8 {
                range "0..7";
              }
              default "7";
              description
                "The priority of the associated MPTED tunnel with
                 respect to taking resources, in the range of 0 to 7.
                 The value 0 is the highest priority.";
            }
            leaf hold-priority {
              type uint8 {
                range "0..7";
              }
              default "7";
              description
                "The priority of the associated MPTED tunnel with
                 respect to holding resources, in the range of 0 to 7.
                 The value 0 is the highest priority.";
            }
            uses mpted-jct-status-info;
            container jct-instances {
              description
                "Container for list of instances of the MPTED
                 junction.";
              list jct-instance {
                key "jct-version";
                description
                  "MPTED junction instance entry.";
                uses mpted-jct-instance-status-info;
                container phops {
                  description
                    "Container for associated JCT-PHOPs.";
                  list phop {
                    key "hop-address hop-index";
                    uses mpted-jct-hop-info;
                    description
                      "JCT-PHOP entry.";
                    leaf in-label {
                      type rt-types:mpls-label;
                      config false;
                      description
                        "Label used by incoming packets
                         on the JCT-PHOP.";
                    }
                    leaf status {
                      type identityref {
                        base mpted-jct-phop-status;
```

```
                    }
                    config false;
                    description
                      "Status of the JCT-PHOP.";
                  }
                }
              }
              container nhops {
                description
                  "Container for associated JCT-NHOPs.";
                list nhop {
                  key "hop-address hop-index";
                  description
                    "JCT-NHOP entry.";
                  uses mpted-jct-nhop-info;
                  leaf bandwidth-reserved {
                    type uint64;
                    config false;
                    description
                      "Amount of bandwidth reserved on the TE link
                       associated with the JCT-NHOP.";
                  }
                  leaf out-label {
                    type rt-types:mpls-label;
                    config false;
                    description
                      "Label used by outgoing packets
                       on the JCT-NHOP.";
                  }
                  leaf status {
                    type identityref {
                      base mpted-jct-nhop-status;
                    }
                    config false;
                    description
                      "Status of the JCT-NHOP.";
                  }
                }
              }
            }
          }
        }
      }
    }
  }
  <CODE ENDS>
```

Figure 4: MPTED JCT YANG module

4. Security Considerations

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

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

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

"/te/mpted-tunnels": Unauthorized access to this list could influence how traffic is forwarded through the network.

"/te/mpted-junctions": Unauthorized access to this list could influence how traffic is forwarded on a junction node.

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

"/te/mpted-tunnels/tunnel/junctions": Unauthorized read access to this list can disclose how traffic is load-balanced on each junction node.

5. IANA Considerations

This document requests IANA to register the following URIs in the "ns" subregistry within the "IETF XML Registry" [RFC3688].

```
      URI:  urn:ietf:params:xml:ns:yang:ietf-mpted
      Registrant Contact:  The IESG.
      XML: N/A; the requested URI is an XML namespace.

      URI:  urn:ietf:params:xml:ns:yang:ietf-mpted-jct
      Registrant Contact:  The IESG.
      XML: N/A; the requested URI is an XML namespace.
```

This document requests IANA to register the following YANG modules in the "YANG Module Names" subregistry [RFC6020] within the "YANG Parameters" registry.

```
      name:      ietf-mpted
      namespace: urn:ietf:params:xml:ns:yang:ietf-mpted
      prefix:    mpted
      reference: RFC XXXX

      name:      ietf-mpted-jct
      namespace: urn:ietf:params:xml:ns:yang:ietf-mpted-jct
      prefix:    mpted-jct
      reference: RFC XXXX
```

RFC Editor: Please replace XXXX with the RFC number assigned to this document.

6. References

6.1. Normative References

[I-D.draft-ietf-teas-yang-te]
Saad, T., Gandhi, R., Liu, X., Beeram, V. P., and I. Bryskin, "A YANG Data Model for Traffic Engineering Tunnels, Label Switched Paths, and Interfaces", Work in Progress, Internet-Draft, draft-ietf-teas-yang-te-39, 17 October 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-teas-yang-te-39>.

[I-D.draft-kompella-teas-mpte]
Kompella, K., Jalil, L., Khaddam, M., and A. Smith, "Multipath Traffic Engineering", Work in Progress, Internet-Draft, draft-kompella-teas-mpte-01, 7 July 2025, <https://datatracker.ietf.org/doc/html/draft-kompella-teas-mpte-01>.

[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/rfc/rfc2119>.

[RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, RFC 3688, DOI 10.17487/RFC3688, January 2004, <https://www.rfc-editor.org/rfc/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/rfc/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/rfc/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/rfc/rfc6242>.

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

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

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/rfc/rfc8174>.

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

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

6.2. Informative References

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

Appendix A. MPTED YANG Module - Complete Tree Structure

```
   module: ietf-mpted

     augment /te:te:
       +--rw mpted-tunnels
          +--rw tunnel* [originator identifier]
             +--rw originator                          inet:ip-address
             +--rw identifier                          uint32
             +--rw name?                               string
             +--rw ingresses
             |  +--rw ingress* [ingress-id]
             |     +--rw ingress-id        inet:ip-address
             |     +--rw set-bandwidth?    uint64
             |     +--rw auto-bandwidth
             |        +--rw min-bw?             uint64
             |        +--rw max-bw?             uint64
             |        +--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
             +--rw egress*                             inet:ip-address
             +--rw type?                               identityref
             +--rw setup-priority?                     uint8
             +--rw hold-priority?                      uint8
             +--rw optimization-metric?                identityref
             +--rw resource-affinities
             |  +--rw exclude-any*   string
             |  +--rw include-any*   string
             |  +--rw include-all*   string
             +--rw bandwidth-spec-type?                identityref
             +--rw auto-bandwidth-adjust-interval?   uint32
             +--rw mpted-computer?                     inet:ip-address
             +--rw signaling-type?                     identityref
             +--rw signaling-source?                   inet:ip-address
             +--ro current-version?                    uint16
             +--ro status?                             identityref
             +--ro instances
                +--ro instance* [version]
                   +--ro version      uint16
                   +--ro status?      identityref
                   +--ro junctions
                      +--ro junction* [node-id]
```

```
                        +--ro node-id                inet:ip-address
                        +--ro current_jct_version?   uint16
                        +--ro bandwidth-requested?   uint64
                        +--ro mtu?                   uint16
                        +--ro status?                identityref
                        +--ro jct-instances
                           +--ro jct-instance* [jct-version]
                              +--ro jct-version            uint16
                              +--ro bandwidth-requested?   uint64
                              +--ro mtu?                   uint16
                              +--ro status?                identityref
                              +--ro phops
                              |  +--ro phop* [hop-address hop-index]
                              |     +--ro hop-address
                              |     |       inet:ip-address
                              |     +--ro hop-index      uint32
                              |     +--ro in-label?
                              |     |       rt-types:mpls-label
                              |     +--ro status?        identityref
                              +--ro nhops
                                 +--ro nhop* [hop-address hop-index]
                                    +--ro hop-address
                                    |       inet:ip-address
                                    +--ro hop-index             uint32
                                    +--ro load-share?           uint16
                                    +--ro bandwidth-reserved?   uint64
                                    +--ro out-label?
                                    |       rt-types:mpls-label
                                    +--ro status?
                                            identityref
```

Figure 5: MPTED YANG tree

Appendix B. MPTED-JCT YANG Module - Complete Tree Structure

```
module: ietf-mpted-jct

  augment /te:te:
    +--rw mpted-junctions
       +--rw junction* [node-id originator tnl-id tnl-vers sig-src]
          +--rw node-id                inet:ip-address
          +--rw originator             inet:ip-address
          +--rw tnl-id                 uint32
          +--rw tnl-vers               uint16
          +--rw sig-src                inet:ip-address
          +--rw name?                  string
          +--rw ingress*               inet:ip-address
          +--rw egress*                inet:ip-address
          +--rw type?                  identityref
          +--rw setup-priority?        uint8
          +--rw hold-priority?         uint8
          +--rw current_jct_version?   uint16
          +--rw bandwidth-requested?   uint64
          +--ro mtu?                   uint16
          +--ro status?                identityref
          +--rw jct-instances
             +--rw jct-instance* [jct-version]
                +--rw jct-version            uint16
                +--rw bandwidth-requested?   uint64
                +--ro mtu?                   uint16
                +--ro status?                identityref
                +--rw phops
                |  +--rw phop* [hop-address hop-index]
                |     +--rw hop-address    inet:ip-address
                |     +--rw hop-index      uint32
                |     +--ro in-label?      rt-types:mpls-label
                |     +--ro status?        identityref
                +--rw nhops
                   +--rw nhop* [hop-address hop-index]
                      +--rw hop-address           inet:ip-address
                      +--rw hop-index             uint32
                      +--rw load-share?           uint16
                      +--ro bandwidth-reserved?   uint64
                      +--ro out-label?            rt-types:mpls-label
                      +--ro status?               identityref
```

Figure 6: MPTED-JCT YANG tree

Acknowledgments

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

This document was prepared using kramdown.

Contributors

Chandrasekar Ramachandran
Juniper Networks
Email: csekar@juniper.net

Authors' Addresses

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

Kireeti Kompella
Juniper Networks
Email: kireeti@juniper.net