BIER WG
Internet-Draft
Intended status: Standards Track
Expires: 16 February 2026

Z. Zhang
ZTE Corporation
C. Wang
Individual
R. Chen
ZTE Corporation
F. Hu
Individual
M. Sivakumar
Juniper networks
H. Chen
China Telecom

15 August 2025

A YANG data model for Tree Engineering for Bit Index Explicit Replication (BIER-TE)
draft-ietf-bier-te-yang-09

Abstract

This document defines a YANG data module for Tree Engineering for Bit Index Explicit Replication (BIER-TE) configuration and operation.

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 16 February 2026.

Copyright Notice

Table of Contents

1. Introduction

[RFC9262] introduces an architecture for BIER-TE: Tree Engineering for Bit Index Explicit Replication (BIER). This document defines a YANG data module for BIER TE. The content is consistent with the TE architecture draft. This YANG data module also contains BIER TE FRR items. For BIER TE FRR usage, please refer to [I-D.eckert-bier-te-frr]. Therefore, this YANG data module is also applicable to [I-D.eckert-bier-rbs].

2. Conventions Used in This Document

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.

3. Design of the Data Model

The BIER TE YANG model includes BIER TE adjacency configuration and forwarding item configuration. Some features can also be used to enhance BIER TE functionality, such as ECMP and FRR.

```
module: ietf-bier-te
  augment /rt:routing/rt:control-plane-protocols
            /rt:control-plane-protocol:
    +--rw bier-te
       +--rw te-adj
       |  +--rw adj-if* [name]
       |     +--rw name         if:interface-ref
       |     +--rw subdomain* [subdomain-id]
       |     |  +--rw subdomain-id    uint16
       |     |  +--rw si* [si]
       |     |     +--rw si        uint16
       |     |     +--rw adj-id*   uint16
       |     +--rw adj-type?    enumeration
       +--rw te-fwd
          +--rw subdomain* [subdomain-id]
             +--rw subdomain-id    uint16
             +--rw bsl* [fwd-bsl]
             |  +--rw fwd-bsl    uint16
             |  +--rw si* [si]
             |     +--rw si            uint16
             |     +--rw te-bift-id
             |     |  +--rw encap-type?   encapsulation-type
             |     |  +--rw value         rt-types:mpls-label
             |     +--rw fwd-items* [te-bp]
             |        +--rw te-bp           uint16
             |        +--rw fwd-next-hop* [next-hop]
             |        |  +--rw next-hop          inet:ip-address
             |        |  +--rw dnr-flag?         boolean
             |        |  +--rw fwd-type
             |        |  |  +--rw (fwd-type)
             |        |  |     +--:(connected)
             |        |  |     +--:(routed)
             |        |  |     +--:(local-decap)
             |        |  |     +--:(other)
             |        |  +--rw te-out-bift-id
             |        |  |  +--rw te-out-bift-id* [encap-type]
             |        |  |     +--rw encap-type    encapsulation-type
             |        |  |     +--rw value
             |        |  |              rt-types:mpls-label
             |        |  +--rw out-if-list* [fwd-intf]
             |        |     +--rw fwd-intf    if:interface-ref
```

```
                |        +--rw te-frr {bier-te-frr}?
                |           +--rw frr-index?      leafref
                |           +--rw resetbitmask
                |              +--rw bit-string* [index]
                |                 +--rw index      uint8
                |                 +--rw bitmask?   uint32
                +--rw te-frr-items {bier-te-frr}?
                   +--rw btaft* [frr-index]
                      +--rw frr-index     uint16
                      +--rw frr-si        uint16
                      +--rw frr-bsl       uint16
                      +--rw addbitmask
                         +--rw bit-string* [index]
                            +--rw index      uint8
                            +--rw bitmask?   uint32

  notifications:
    +---n bier-te-notification
       +--ro bp-is-zero* [if-index]
          +--ro if-index    if:interface-ref
          +--ro adj-type?   enumeration
```

4. BIER-TE configuration

BIER-TE forwarding items are indexed by a combination of sub-domain-id, BitStringLength, and set identifier.

An interface can be used in different subdomains, so BIER TE adjacency information is managed by the BIER TE function rather than by the interface itself.

BIER-TE key information notifications carried by IGP can also be implemented based on this module.

5. Notifications

If the adjacency ID of an adjacency is set to zero, the value is invalid. Notification should be sent to the controller and network manager.

6. RPCs

TBD.

7. BIER TE YANG model

```
<CODE BEGINS> file "ietf-bier-te@2025-01-20.yang"
module ietf-bier-te {

  yang-version 1.1;

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

  prefix bier-te;

  import ietf-inet-types {
    prefix "inet";
    reference
      "RFC 6991: Common YANG Data Types";
  }

  import ietf-routing {
    prefix "rt";
    reference
      "RFC 8349: A YANG Data Model for Routing Management
                 (NMDA Version)";
  }

  import ietf-interfaces {
    prefix "if";
    reference
      "RFC 8343: A YANG Data Model for Interface Management";
  }

  import ietf-routing-types {
    prefix "rt-types";
    reference
      "RFC 8294: Common YANG Data Types for the Routing Area";
  }

  organization " IETF BIER (Bit Indexed Explicit Replication)
                 Working Group";
  contact
    "WG Web:   <http://tools.ietf.org/wg/bier/>
     WG List:  <mailto:bier@ietf.org>

     Editor:   Zheng Zhang
               <mailto:zhang.zheng@zte.com.cn>
     Editor:   Cui Wang
               <mailto:lindawangjoy@gmail.com>
     Editor:   Ran Chen
               <mailto:chen.ran@zte.com.cn>
     Editor:   Fangwei Hu
               <mailto:hufwei@gmail.com>
```

```
        Editor:   Mahesh Sivakumar
                  <mailto:sivakumar.mahesh@gmail.com>
        Editor:   Huanan Chen
                  <mailto:chenhuanan@189.cn>
      ";

    description
      "The module defines the YANG definitions for Traffic
       Engineering for Bit Index Explicit Replication (BIER-TE).

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

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

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

       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 (RFC 2119)
       (RFC 8174) when, and only when, they appear in all
       capitals, as shown here.";

    revision 2025-01-20 {
      description
      "Initial revision.";
      reference
        "RFC9262: Tree Engineering for Bit Index
         Explicit Replication (BIER-TE)";
    }

    /*
    * Features
    */
    feature bier-te-frr {
      description
        "Support Fast Re-route feature in BIER TE.";
      reference
        "I-D.eckert-bier-te-frr: Protection Methods for BIER-TE";
    }
```

```
     /*
      * Identities
      */

     identity bier-te {
       base rt:control-plane-protocol;
       description
         "Identity for the Tree Engineering for Bit Index Explicit
          Replication (BIER-TE).";
       reference
         "RFC9262: Tree Engineering for Bit Index
          Explicit Replication (BIER-TE)";
     }

     typedef encapsulation-type {
       type enumeration {
         enum MPLS {
           description
             "The forwarding encapsulation is MPLS
              described in RFC8296 section 2.1.";
         }
         enum Ethernet {
           description
             "The forwarding encapsulation is Ethernet,
              which is belonging to non-mpls part
              described in RFC8296 section 2.2.";
         }
         enum IPv6 {
           description
             "The forwarding encapsulation is IPv6,
              which is belonging to non-mpls part
              described in RFC8296 section 2.2.";
         }
       }
       description
         "The encapsulation type of the BIER-TE packet.
          If this type is not set, MPLS is the default type.";
     } // encapsulation-type

     grouping bit-string {
       description
         "The bit string which each bit represents an adjacency.
          It is encapsulated in BIER header.";
       reference
           "RFC9262: Tree Engineering for Bit Index
            Explicit Replication (BIER-TE), section 3.3.
            RFC8279: Multicast Using Bit Index Explicit Replication
            (BIER).
```

```
            RFC8296: Encapsulation for Bit Index Explicit Replication
            (BIER) in MPLS and Non-MPLS Networks.";

        list bit-string {
          key "index";
          description
            "As the definition in RFC 8279, the bit-string lengths are
             64, 128, 256, 512, 1024, 2048, 4096 bits. The according
             encapsulation is defined in RFC8296. BIER-TE uses the
             similar function for bit string.";

          leaf index {
            type uint8 {
              range "2..128";
            }
            description
              "The index of bit-mask. The minimum index value is 64 and
               the corresponding bit string length is 64 bits. The
               maximum index value is 128 and the corresponding
               bit-string length is 4096 bits.";
          }
          leaf bitmask {
            type uint32;
            description "The bit-string in 4-octet units.";
          }
        }
      } // bit-string

      grouping adj-type {
        description "The collection of all possible adjacency type.";

        leaf adj-type {
          type enumeration {
            enum p2p {
              description "Describes p2p adjacency.";
            }
            enum bfer {
              description "Describes bfer adjacency.";
            }
            enum leaf-bfer {
              description
                "Describes leaf-bfer adjacency. There is no next BFR that
                 the packet should be forwarded.";
            }
            enum lan {
              description "Describes lan adjacency.";
            }
            enum spoke {
```

```
            description "Describes spoke adjacency of hub-and-spoke.";
          }
          enum ring-clockwise {
            description "Describes clockwise adjacency in ring.";
          }
          enum ring-counterclockwise {
            description
              "Describes counterclockwise adjacency in ring.";
          }
          enum ecmp {
            description
              "Describes ecmp adjacency. When the type is set to ecmp,
               the corresponding ecmp entry should be used to balance
               the load.";
          }
          enum virtual-link {
            description
              "Describes virtual adjacency between two indirect connect
               nodes.";
          }
          enum other {
            description "Describes other id type of adjacency.";
          }
        }
        description "The collection of all possible adjacency type.";
      }
    } // adj-type

    grouping te-items {
      description "The BIER TE forwarding items collection.";

      list fwd-next-hop {
        key "next-hop";
        description
          "The next hop information for forwarding.
           If ECMP function defined in section 3.2.3 is used,
           multiple next hops may be existed.
           If ECMP function is not enabled,
           the next hop may be one only.";

        leaf next-hop {
          type inet:ip-address;
          mandatory true;
          description "Next hop address.";
        }
        leaf dnr-flag {
          type boolean;
          description
```

```
            "When the flag is set to 1, the BP of adjacency should not
             be reset when packet copies are created. The flag makes
             sense only when the forwarding type is 'connected'.";
          }
          container fwd-type {
            description
              "The collection of all possible forwarding types.";
            choice fwd-type {
              mandatory true;
              case connected {
                description
                  "The forwarding type is connected. Mostly connected
                   interfaces.";
              }
              case routed {
                description
                  "The forwarding type is routed. Mostly not connected
                   interfaces.";
              }
              case local-decap {
                description
                  "Means that the packet should be decapsulated and
                   forward out of BIER domain.";
               }
              case other {
                description
                  "Means that the packet should be discarded.";
              }
              description
                "The collection of all possible forwarding types.";
            }
          } // fwd-type

          container te-out-bift-id {
            description
              "The bift-id information corresponding to a specific
               outbound interface.";

            list te-out-bift-id {
              key "encap-type";
              leaf encap-type {
                type encapsulation-type;
                description
                  "The encapsulation type of BIER-TE packet.";
              }
              leaf value {
                type rt-types:mpls-label;
                mandatory true;
```

```
              description
                "The bift-id value of the forwarding item.
                 It can be a mpls label or an index for ethernet
                 or IPv6 encapsulation, which is used to represent
                 specific combination of [SD, BSL, SI].
                 The ethernet or IPv6 index value is the same range
                 (20bits) as mpls label.
                 This value MUST NOT be set to 0.";
            }
            description
              "The bift-id value and the encapsulation type
               for the BIER-TE packet.";
          }
        }

        list out-if-list {
          key "fwd-intf";
          description
            "The outbound interface information for forwarding.";

          leaf fwd-intf {
            type if:interface-ref;
            mandatory true;
            description
              "The out interface of this forwarding item.";
          }
        }
      } // next-hop
    } // te-items

    /*
    * data nodes
    */
    augment "/rt:routing/rt:control-plane-protocols/"
         +"rt:control-plane-protocol" {
      description "The BIER TE information.";
      container bier-te {
        description "The BIER TE information container.";

        container te-adj {
          description "The BIER TE adjacency information.";
          list adj-if {
            key "name";
            description "List of BIER-TE interfaces.";
            leaf name {
              type if:interface-ref;
              description "Interface name reference.";
            }
```

```
            list subdomain {
              key "subdomain-id";
              description
                "The sub-domain which the interface belongs to.
                 One interface can belong to many subdomains.";

              leaf subdomain-id {
                type uint16;
                description "The sub-domain-id of this sub-domain.";
                reference
                  "RFC 8279: Multicast Using Bit Index Explicit
                   Replication (BIER)";
              }
              list si {
                key "si";
                description "The set identifier value.";

                leaf si{
                  type uint16;
                  mandatory true;
                  description
                    "The set identifier of this forwarding item.";
                }
                leaf-list adj-id {
                  type uint16;
                  description "The ID of an adjacency.";
                }
              }
            }
            uses adj-type;
          }
        } // te-adj

        container te-fwd {
          description "The BIER TE forwarding information.";

          list subdomain {
            key "subdomain-id";
            description
              "The sub-domain which the interface belongs to.
               One interface can belong to many subdomains.";

            leaf subdomain-id {
              type uint16;
              description "The sub-domain-id of this sub-domain.";
              reference
                "RFC 8279: Multicast Using Bit Index Explicit
                 Replication (BIER)";
```

```
            }
            list bsl {
              key "fwd-bsl";
              description "The forwarding items in one BSL.";
              leaf fwd-bsl {
                type uint16;
                description "The value of bitstringlength.";
              }
                list si {
                  key "si";
                  description
                    "The forwarding items in one combination of SD,
                     BSL and SI.";
                  leaf si{
                    type uint16;
                    mandatory true;
                    description
                      "The set identifier of this forwarding item.";
                  }
                             container te-bift-id {
                    description
                      "The bift-id which is used to locate the specific
                       forwarding item.";
                    leaf encap-type {
                      type encapsulation-type;
                      description
                        "The encapsulation type for BIER-TE packet.";
                    }

                    leaf value {
                      type rt-types:mpls-label;
                      mandatory true;
                      description
                        "The bift-id value of the forwarding item.
                         It can be a mpls label or an index for
                         ethernet or IPv6 encapsulation,
                         which is used to represent specific
                         combination of [SD, BSL, SI].
                         The ethernet or IPv6 index value is the same
                         range (20bits) as mpls label.";
                    }
                  }

                  list fwd-items {
                    key "te-bp";
                    description
                      "The forwarding information of one BIER TE
                      item.";
```

```
                leaf te-bp {
                  type uint16;
                  mandatory true;
                  description
                    "The bit index of a BIER TE forwarding item.";
                }
                uses te-items;

                container te-frr {
                  if-feature bier-te-frr;
                  leaf frr-index {
                    type leafref {
                      path "../../../../../" +
                           "te-frr-items/btaft/frr-index";
                    }
                    description "The index of this frr path.";
                  }
                  container resetbitmask {
                    description
                      "The deleting bitmask of the forwarding
                       item.";
                    uses bit-string;
                  }
                  description
                    "If this link is protected, frr items can be
                     used to forward flows when this link is
                     down.";
                } // te-frr
              } // fwd-items
            } // si
          } // bsl

          container te-frr-items {
            if-feature bier-te-frr;
            description "The TE fast re-route information.";
              list btaft {
              key "frr-index";
              description
                "The index of the frr paths. This item can be used
                 for multiple links protection in different SI.";
              leaf frr-index {
                type uint16;
                mandatory true;
                description "The frr item index.";
              }
              leaf frr-si{
                type uint16;
                mandatory true;
```

```
                  description
                    "The set identifier of this forwarding item.";
                }
                leaf frr-bsl {
                  type uint16;
                  mandatory true;
                  description "The value of bitstringlength.";
                }
                container addbitmask {
                  description
                    "The adding bitmask of the forwarding item.
                     This item should be merged into the packet's
                     bit-string.";
                  uses bit-string;
                }
              } // btaft
            } // te-frr-items
          } // subdomain
        } // te-fwd
      } // bier-te
    }

    /*
    * Notifications
    */
    notification bier-te-notification {
        description
          "The notification is sent when a condition changes.";
        list bp-is-zero {
            key "if-index";
            description "The adjacency id is zero. It is invalid.";
            leaf if-index {
                type if:interface-ref;
                description "The adjacency id is zero.";
            }
            uses adj-type;
        }
    }
  }
  <CODE ENDS>
```

8. 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].

9. IANA Considerations

The IANA is requested to assign two new URIs from the IETF XML registry ([RFC3688]). Authors are suggesting the following URI:

URI: urn:ietf:params:xml:ns:yang:ietf-bier-te

Registrant Contact: BIER WG

XML: N/A, the requested URI is an XML namespace

This document also requests one new YANG module name in the YANG Module Names registry ([RFC6020]) with the following suggestion:

name: ietf-bier-te

namespace: urn:ietf:params:xml:ns:yang:ietf-bier-te

prefix: bier-te

reference: RFC XXXX

10. Acknowledgement

The authors would like to thank Min Gu (gumin20181129@163.com) for her testing, verification and valuable suggestion. And the authors would like to thank Benjamin R and Benchong Xu for their valuable comments.

11. References

11.1. Normative References

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

[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>.

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

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

[RFC8279] Wijnands, IJ., Ed., Rosen, E., Ed., Dolganow, A., Przygienda, T., and S. Aldrin, "Multicast Using Bit Index Explicit Replication (BIER)", RFC 8279, DOI 10.17487/RFC8279, November 2017, <https://www.rfc-editor.org/info/rfc8279>.

[RFC8296] Wijnands, IJ., Ed., Rosen, E., Ed., Dolganow, A., Tantsura, J., Aldrin, S., and I. Meilik, "Encapsulation for Bit Index Explicit Replication (BIER) in MPLS and Non-MPLS Networks", RFC 8296, DOI 10.17487/RFC8296, January 2018, <https://www.rfc-editor.org/info/rfc8296>.

[RFC8343] Bjorklund, M., "A YANG Data Model for Interface Management", RFC 8343, DOI 10.17487/RFC8343, March 2018, <https://www.rfc-editor.org/info/rfc8343>.

[RFC8349] Lhotka, L., Lindem, A., and Y. Qu, "A YANG Data Model for Routing Management (NMDA Version)", RFC 8349, DOI 10.17487/RFC8349, March 2018, <https://www.rfc-editor.org/info/rfc8349>.

[RFC8407] Bierman, A., "Guidelines for Authors and Reviewers of Documents Containing YANG Data Models", BCP 216, RFC 8407, DOI 10.17487/RFC8407, October 2018, <https://www.rfc-editor.org/info/rfc8407>.

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

[RFC9262] Eckert, T., Ed., Menth, M., and G. Cauchie, "Tree Engineering for Bit Index Explicit Replication (BIER-TE)", RFC 9262, DOI 10.17487/RFC9262, October 2022, <https://www.rfc-editor.org/info/rfc9262>.

11.2. Informative References

[I-D.eckert-bier-rbs]
Eckert, T. T., Menth, M., Geng, X., Zheng, X., Meng, R., and F. Li, "Recursive BitString Structure (RBS) Addresses for BIER and MSR6", Work in Progress, Internet-Draft, draft-eckert-bier-rbs-00, 24 October 2022, <https://datatracker.ietf.org/doc/html/draft-eckert-bier-rbs-00>.

[I-D.eckert-bier-te-frr]
Eckert, T. T., Cauchie, G., Braun, W., and M. Menth, "Protection Methods for BIER-TE", Work in Progress, Internet-Draft, draft-eckert-bier-te-frr-03, 5 March 2018, <https://datatracker.ietf.org/doc/html/draft-eckert-bier-te-frr-03>.

[I-D.ietf-bier-bier-yang]
Chen, R., hu, F., Zhang, Z., dai.xianxian@zte.com.cn, and M. Sivakumar, "YANG Data Model for BIER Protocol", Work in Progress, Internet-Draft, draft-ietf-bier-bier-yang-10, 11 February 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-bier-bier-yang-10>.

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

Authors' Addresses

Zheng(Sandy) Zhang
ZTE Corporation
Nanjing
China
Email: zhang.zheng@zte.com.cn

Cui(Linda) Wang
Individual

Email: lindawangjoy@gmail.com

Ran Chen
ZTE Corporation
Nanjing
China
Email: chen.ran@zte.com.cn

Fangwei Hu
Individual
Shanghai
China
Email: hufwei@gmail.com

Mahesh Sivakumar
Juniper networks
1133 Innovation Way
Sunnyvale, CALIFORNIA 94089 ,
United States
Email: sivakumar.mahesh@gmail.com

Huanan Chen
China Telecom
China
Email: chenhuanan@189.cn