

BIER WG
Internet-Draft
Intended status: Standards Track
Expires: September 1, 2016

Zheng. Zhang
Cui. Wang
Ran. Chen
Fangwei. Hu
ZTE Corporation
February 29, 2016

BIER TE YANG module
draft-zhang-bier-te-yang-01

Abstract

This document defines a YANG data model for 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 <http://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 September 1, 2016.

Copyright Notice

Copyright (c) 2016 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 (<http://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.

Internet-Draft

BIER TE YANG module

February 2016

Table of Contents

1.	Introduction	2
2.	Design of the Data Model	2
3.	BIER-TE configuration	4
4.	Notifications	5
5.	BIER TE YANG module	5
6.	Normative References	12
	Authors' Addresses	13

[1.](#) Introduction

[I-D.eckert-bier-te-arch] introduces an architecture for BIER-TE: Traffic Engineering for Bit Index Explicit Replication (BIER). This document defines a YANG data model for BIER TE. The content is in keeping with the TE architecture draft.

[2.](#) Design of the Data Model

Instead of using respective sub-domain-id, si and bsl information like in BIER yang draft [\[I-D.chh-bier-bier-yang\]](#), this document tries to group these sub-domain-id, si and bsl information in a new bier-common grouping to simplify the reference. Later yang modules may import the common grouping easily. Further, if this optimization is recognized, then BIER yang draft [\[I-D.chh-bier-bier-yang\]](#) will be updated to group these sub-domain-id, si and bsl information as well.

```

module: ietf-bier-te
augment /rt:routing:
  +--rw bier-te-config
    +--rw te-subdomain* [subdomain-id]
      +--rw subdomain-id      sub-domain-id
      +--rw adj-id* [adjID]
        | +--rw adjID      adjid
        | +--rw adj-if      uint32
        | +--rw (te-adjID-type)
        |   +--:(p2p)
        |   +--:(bfer)
        |   +--:(leaf-bfer)
        |   +--:(lan)
        |   +--:(spoke)
        |   +--:(ring-clockwise)
        |   +--:(ring-counterclockwise)

```

```

|      +---:(ecmp)
|      +---:(virtual-link)
|      +---:(other)
+---rw te-bsl* [fwd-bsl]
|   +---rw fwd-bsl      uint16

```

```

|   +---rw te-si* [si]
|       +---rw si          si
|       +---rw te-f-index* [te-f-index]
|           +---rw te-f-index      bit-string
|           +---rw (te-adj-type)
|               | +---:(connected)
|               | +---:(routed)
|               | +---:(local-decap)
|               | +---:(ecmp)
|               | +---:(other)
|           +---rw f-bm          bit-string
|           +---rw f-intf       uint32
|           +---rw ecmp?        boolean
|           +---rw frr?         boolean
+---rw ecmp-path* [index]
|   +---rw index      uint32
|   +---rw number* [number]
|       +---rw number      uint16
|       +---rw out-if      uint32
+---rw btaft* [adj-index]
|   +---rw adj-index      uint32
|   +---rw bitposition    bit-string
|   +---rw resetbitmask   bit-string
|   +---rw addbitmask     bit-string
augment /rt:routing:
+---ro bier-te-state
+---ro te-subdomain* [subdomain-id]
|   +---ro subdomain-id    sub-domain-id
+---ro adj-id* [adjID]
|   +---ro adjID          adjid
|   +---ro adj-if          uint32
|   +---ro (te-adjID-type)
|       +---:(p2p)
|       +---:(bfer)
|       +---:(leaf-bfer)
|       +---:(lan)

```

```

|      +---:(spoke)
|      +---:(ring-clockwise)
|      +---:(ring-counterclockwise)
|      +---:(ecmp)
|      +---:(virtual-link)
|      +---:(other)
+---ro te-bsl* [fwd-bsl]
|   +---ro fwd-bsl      uint16
|   +---ro te-si* [si]
|       +---ro si              si
|       +---ro te-f-index* [te-f-index]
|           +---ro te-f-index      bit-string

```

```

|      +---ro (te-adj-type)
|      |   +---:(connected)
|      |   +---:(routed)
|      |   +---:(local-decap)
|      |   +---:(ecmp)
|      |   +---:(other)
|      +---ro f-bm              bit-string
|      +---ro f-intf            uint32
|      +---ro ecmp?             boolean
|      +---ro frr?              boolean
+---ro ecmp-path* [index]
|   +---ro index      uint32
|   +---ro number* [number]
|       +---ro number      uint16
|       +---ro out-if      uint32
+---ro btaft* [adj-index]
|   +---ro adj-index      uint32
|   +---ro bitposition     bit-string
|   +---ro resetbitmask    bit-string
|   +---ro addbitmask      bit-string
notifications:
+---n bier-te-notification
+---ro adjID-is-zero* [if-index]
+---ro if-index      uint32
+---ro (te-adjID-type)
|   +---:(p2p)
|   +---:(bfer)
|   +---:(leaf-bfer)
|   +---:(lan)

```

```

+--:(spoke)
+--:(ring-clockwise)
+--:(ring-counterclockwise)
+--:(ecmp)
+--:(virtual-link)
+--:(other)

```

[3.](#) BIER-TE configuration

The BIER-TE information is indexed by the sub-domain ID. Maybe there are some global BIER-TE information, it should be added in later version.

One interface can be used in different sub-domain, so the BIER TE adjacency information is managed by BIER TE other than by interface.

Because the BIER-TE is controlled by controller now, the information about IGP is not defined. If in the future the IGP is used to carry

the information about BIER-TE, the IGP extension will be added in this document.

[4.](#) Notifications

If the adjacency id of one adjacency is set to zero, the value is invalid. The notification should be sent to controller and network manager.

[5.](#) BIER TE YANG module

```

<CODE BEGINS> file "ietf-bier-te.yang"
module ietf-bier-te {

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

    prefix bier-te;

    import ietf-routing {
        prefix "rt";
    }
    /* import bier-common {

```

```

    prefix "bier-common";
} */

organization " IETF BIER(Bit Indexed Explicit Replication ) Working Group";
contact
    "WG List:  <mailto:bier@ietf.org>
    WG Chair: Tony Przygienda
                <mailto:tonysietf@gmail.com>
    WG Chair: Greg Shepherd
                <mailto:gjshep@gmail.com>

    Editor:   Zheng Zhang
                <mailto:zhang.zheng@zte.com.cn>
    Editor:   Cui Wang
                <mailto:wang.cui1@zte.com.cn>
    Editor:   Ran Chen
                <mailto:chen.ran@zte.com.cn>
    Editor:   Fangwei Hu
                <mailto:hu.fangwei@zte.com.cn>
    ";

description
    "This module contains a collection of YANG definitions for
    managing BIER TE information.";

revision 2016-03-01 {

```

```

    description
    "Initial version.";
    reference "https://tools.ietf.org/html/draft-zhang-bier-te-yang";
}

grouping te-adj-type {
    description "The collection of all possible adjacency type.";
    choice te-adj-type {
        mandatory true;
        case connected {
            description "The type of adjacency is connected. Mostly connect
            }
        case routed {
            description "The type of adjacency is routed. Mostly not connect
            }
    }
}

```

```

        case local-decap {
            description "Means that the packet should be decapsulated and f
        }
        case ecmp {
            description "There is more than one path in the adjacency with
        }
        case other {
            description "Means that the packet should be discarded.";
        }
        description "The collection of all possible adjacency type.";
    }
}

```

```

grouping te-adjID-type {
    description "The collection of all possible adjacency type.";
    choice te-adjID-type {
        mandatory true;
        case p2p {
            description "Describes p2p adjacency.";
        }
        case bfer {
            description "Describes bfer adjacency.";
        }
        case leaf-bfer {
            description "Describes leaf-bfer adjacency. There is no next BF
        }
        case lan {
            description "Describes lan adjacency..";
        }
        case spoke {
            description "Describes spoke adjacency of hub-and-spoke.";
        }
        case ring-clockwise {

```

```

        description "Describes clockwise adjacency in ring.";
    }
    case ring-counterclockwise {
        description "Describes counterclockwise adjacency in ring.";
    }
    case ecmp {
        description "Describes ecmp adjacency.";
    }
}

```

```

        case virtual-link {
            description "Describes virtual adjacency between two indirect c
        }
        case other {
            description "Describes other id type of adjacency.";
        }
        description "The collection of all possible adjacency type.";
    }
}

typedef bsl {
    type enumeration{
        enum 64-bit{
            description "bitstringlength is 64";
        }
        enum 128-bit{
            description "bitstringlength is 128";
        }
        enum 256-bit{
            description "bitstringlength is 256";
        }
        enum 512-bit{
            description "bitstringlength is 512";
        }
        enum 1024-bit{
            description "bitstringlength is 1024";
        }
        enum 2048-bit{
            description "bitstringlength is 2048";
        }
        enum 4096-bit{
            description "bitstringlength is 4096";
        }
    }
    description "The bitstringlength type for imposition mode.";
}

typedef adjid {
    type uint32;
    description "The type for adjacency ID.";
}

```

}


```

/* The definition of si/sub-domain-id/bit-string will be deleted later. */
typedef si {
    type uint16;
    description "The type for set identifier";
}

typedef sub-domain-id {
    type uint16;
    description "The type for sub-domain-id";
}

typedef bit-string {
    type uint16;
    description "The bit mask of one bitstring.";
}

/* The bier-common grouping will be moved to BIER yang. */
grouping bier-common {
    description "Common information in BIER";
    leaf subdomain-id {
        type sub-domain-id;
        description "ID of the sub domain.";
    }
    leaf si {
        type si;
        description "The value of the set identifier.";
    }
    leaf bsl {
        type bsl;
        description "The BitStringLength supported by this node.";
    }
    leaf bit-string {
        type bit-string;
        description "The bit-string of BIER forwarding.";
    }
}

grouping te-adjID {
    list adj-id {
        key "adjID";
        description "This ID information of one adjacency.";
        leaf adjID {
            type adjid;
            mandatory true;
            description "The adjacency id.";
        }
    }
}

```

```
        leaf adj-if {
            /* type if:if-index; */
            type uint32; /* for compilation */
            mandatory true;
            description "The corresponding interface of this adjacency.";
        }
        uses te-adjID-type;
    }
    description "This group presents adjacency ID information for BIER TE."
}

grouping te-ecmp {
    description "The ecmp information.";
    list ecmp-path {
        key "index";
        description "The index of the ecmp paths.";
        leaf index {
            type uint32;
            mandatory true;
            description "The ecmp index.";
        }
        list number {
            key "number";
            description "The list of the ecmp paths.";
            leaf number {
                type uint16;
                mandatory true;
                description "The number of the ecmp paths.";
            }
            leaf out-if {
                /* type if:if-index; */
                type uint32; /* for compilation */
                mandatory true;
                description "The outgoing interface.";
            }
        }
    }
}

grouping te-frr {
    description "The TE fast reroute information.";
    list btaft {
        key "adj-index";
        description "The adjacency index of the frr paths.";
        leaf adj-index {
            type uint32;
        }
    }
}
```

```
mandatory true;
description "The frr adjacency index.;"
```

```
    }
    leaf bitposition {
        type bit-string;
        mandatory true;
        description "The bitposition information.;"
    }
    leaf resetbitmask {
        type bit-string;
        mandatory true;
        description "The deleting bitmask of the forwarding item.;"
    }
    leaf addbitmask {
        type bit-string;
        mandatory true;
        description "The adding bitmask of the forwarding item.;"
    }
}

grouping te-items {
    description "The BIER TE forwarding items collection.;"
    uses te-adj-type;
    leaf f-bm {
        type bit-string;
        mandatory true;
        description "The bitmask of the forwarding item.;"
    }
    leaf f-intf {
        /* type if:if-index; */
        type uint32; /* for compilation */
        mandatory true;
        description "The out interface of this forwarding item.;"
    }
    leaf ecmp {
        type boolean;
        description "The capability of ecmp paths.;"
    }
    leaf frr {
        type boolean;
```

```

        description "The capability of fast re-route.";
    }
}

grouping te-fwd-item {
    list te-si {
        key "si";
        description "The forwarding items of one set identifier.";
        leaf si{

```

```

        type si;
        mandatory true;
        description "The set identifier of this forwarding item.";
    }
    list te-f-index {
        key "te-f-index";
        description "The forwarding information of one BIER TE item.";
        leaf te-f-index {
            type bit-string;
            mandatory true;
            description "The bit index of BIER TE forwarding item.";
        }
        uses te-items;
    }
}
description "The forwarding items in one set identifier.";
}

grouping te-info {
    description "The BIER TE forwarding information.";
    list te-subdomain {
        key "subdomain-id";
        description "The forwarding items of one sub-domain.";
        leaf subdomain-id {
            type sub-domain-id;
            description "The sub-domain-id of this sub-domain.";
        }
        uses te-adjID;

        list te-bsl {
            key "fwd-bsl";
            description "The forwarding items in one bitstringlength.";

```

```

        leaf fwd-bsl {
            type uint16;
            description "The value of bitstringlength.";
        }
        uses te-fwd-item;
    }
    uses te-ecmp;
    uses te-frr;
}

augment "/rt:routing" {
    description "The BIER TE information.";
    container bier-te-config {
        description "The BIER TE information container.";
        uses te-info;
    }
}

```

```

    }
}

augment "/rt:routing" {
    description "The read-only BIER TE information.";
    container bier-te-state {
        config false;
        description "The BIER TE information in nodes.";
        uses te-info;
    }
}

notification bier-te-notification {
    description
        "This notification is sent when a condition changes in BIER TE.";
    list adjID-is-zero {
        key "if-index";
        description "The adjacency id is zero.";
        leaf if-index {
            type uint32;
            description "The adjacency id of this interface is zero.";
        }
        uses te-adjID-type;
    }
}
}

```

```
}  
<CODE ENDS>
```

6. Normative References

[I-D.chh-bier-bier-yang]

Chen, R., hu, f., Zhang, Z., dai.xianxian@zte.com.cn, d., and M. Sivakumar, "YANG Data Model for BIER Protocol", [draft-chh-bier-bier-yang-02](#) (work in progress), November 2015.

[I-D.eckert-bier-te-arch]

Eckert, T. and G. Cauchie, "Traffic Engineering for Bit Index Explicit Replication BIER-TE", [draft-eckert-bier-te-arch-02](#) (work in progress), October 2015.

[I-D.ietf-bier-architecture]

Wijnands, I., Rosen, E., Dolganow, A., P, T., and S. Aldrin, "Multicast using Bit Index Explicit Replication", [draft-ietf-bier-architecture-03](#) (work in progress), January 2016.

Zhang, et al.

Expires September 1, 2016

[Page 12]

Internet-Draft

BIER TE YANG module

February 2016

[RFC6020] Bjorklund, M., Ed., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", [RFC 6020](#), DOI 10.17487/RFC6020, October 2010, <<http://www.rfc-editor.org/info/rfc6020>>.

[RFC6087] Bierman, A., "Guidelines for Authors and Reviewers of YANG Data Model Documents", [RFC 6087](#), DOI 10.17487/RFC6087, January 2011, <<http://www.rfc-editor.org/info/rfc6087>>.

[RFC7223] Bjorklund, M., "A YANG Data Model for Interface Management", [RFC 7223](#), DOI 10.17487/RFC7223, May 2014, <<http://www.rfc-editor.org/info/rfc7223>>.

Authors' Addresses

Zheng(Sandy) Zhang
ZTE Corporation
No. 50 Software Ave, Yuhuatai Distinct

Nanjing
China

Email: zhang.zheng@zte.com.cn

Cui(Linda) Wang
ZTE Corporation
No. 50 Software Ave, Yuhuatai Distinct
Nanjing
China

Email: wang.cui1@zte.com.cn

Ran Chen
ZTE Corporation
No. 50 Software Ave, Yuhuatai Distinct
Nanjing
China

Email: chen.ran@zte.com.cn

Fangwei Hu
ZTE Corporation
No.889 Bibo Rd
Shanghai
China

Email: hu.fangwei@zte.com.cn

