Abstract

This document defines a YANG data model for BIER RPCs.

Status of This Memo

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

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

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

This Internet-Draft will expire on March 1, 2019.

Copyright Notice

Copyright (c) 2018 IETF Trust and the persons identified as the document authors. All rights reserved.

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

This document defines a YANG data model for BIER RPCs.

2. Design of the Data Model

module: ietf-bier-rpcs
augment /coam-methods:continuity-check/coam-methods:input/coam-methods:destination-tp:
  +---- bier-address
  |     +---- (bier-address)?
  |     |     +--:(bfrs)
  |     |     |     +---- bier-bsl?           bier:bsl
  |     |     |     +---- bier-subdomainid?   bier:sub-domain-id
  |     |     |     +---- bfir?               bier:bfr-id
  |     |     |     +---- bier-bfers* [bier-bfrid]
  |     |     |     |     +---- bier-bfrid    bier:bfr-id
  |     |     +--:(m-addr)
  |     |     |     +---- vpn-id?             uint32
  |     |     |     +---- source-address?     inet:ip-address
  |     |     |     +---- source-wildcard?    uint8
  |     |     |     +---- group-address?      inet:ip-address
  |     |     |     +---- group-wildcard?    uint8
  |     |     |     +---- target-bfers* [bier-bfrid]
  |     |     |     |     +---- bier-bfrid    bier:bfr-id
  |     |     |     +---- reply-mode?         reply-mode
  |     |     +---- bier-te-address
  |     |     |     +---- bier-te-subdomainid?   bier:sub-domain-id
  |     |     |     +---- bier-te-bp-info* [bier-te-bsl]
  |     |     |     |     +---- bier-te-bsl       bier:bsl
  |     |     |     |     |     +---- bitstring* [si]
  |     |     |     |     |     |     +---- si              bier:si
  |     |     |     |     |     |     |     +---- bitpositions* [bitposition]
  |     |     |     |     |     |     |     +---- bitposition    bier-te:bit-string
  |     |     |     |     +---- fec-stack-type* [si]
  |     |     |     |     |     +---- si              bier:si
  |     |     |     |     |     +---- fec-stack-info* [bitposition]
augment /coam-methods:continuity-check/coam-methods:output/coam-methods:src-test-point:
++++ bier-address
++++ (bier-address)?
+++: (bfrs)
++++ bier-bsl? bier:bsl
++++ bier-subdomainid? bier:sub-domain-id
++++ bfr? bier:bfr-id
++++ bier-bfers* [bier-bfrid]
++++ bier-bfrid bier:bfr-id
+++: (m-addr)
++++ vpn-id? uint32
++++ source-address? inet:ip-address
++++ source-wildcard? uint8
++++ group-address? inet:ip-address
++++ group-wildcard? uint8
++++ target-bfers* [bier-bfrid]
++++ bier-bfrid bier:bfr-id
++++ reply-mode? reply-mode
++++ bier-te-address
++++ bier-te-subdomainid? bier:sub-domain-id
++++ bier-te-bp-info* [bier-te-bsl]
++++ bier-te-bsl bier:bsl
++++ bitstring* [si]
++++ si bier:si
++++ bitpositions* [bitposition]
++++ bitposition bier-te:bit-string
++++ fec-stack-type* [si]
++++ si bier:si
++++ fec-stack-info* [bitposition]
++++ bitposition bier-te:bit-string
++++ (fec-stack-type)
+++: (connected)
++++ local-bfr? inet:ip-address
++++ local-interface? if:interface-ref
+++: (routed)
++++ routed-bfr? inet:ip-address
+++: (local-decap)
++++ bfr? inet:ip-address
reply-mode-te
augment /coam-methods:continuity-check/coam-methods:output/coam-methods:dest-test-point:
  +----- bier-address
      |  +---- (bier-address)?
      |      |  +---- bier-bsl?    bier:bsl
      |      |  +---- bier-subdomainid?    bier:sub-domain-id
      |      |  +---- bfir?    bier:bfr-id
      |      |  +---- bier-bfers* [bier-bfrid]
      |      |      |  +---- bier-bfrid    bier:bfr-id
      |      |      +--:(m-addr)
      |      |      |  +---- vpn-id?    uint32
      |      |      |  +---- source-address?    inet:ip-address
      |      |      |  +---- source-wildcard?    uint8
      |      |      |  +---- group-address?    inet:ip-address
      |      |      |  +---- group-wildcard?    uint8
      |      |  +---- target-bfers* [bier-bfrid]
      |      |      |  +---- bier-bfrid    bier:bfr-id
      |      +---- reply-mode?    reply-mode
  +---- bier-te-address
  +---- bier-te-subdomainid?    bier:sub-domain-id
  +---- bier-te-bp-info* [bier-te-bsl]
      +---- bier-te-bsl    bier:bsl
      +---- bitstring* [si]
          |  +---- si    bier:si
          |      +---- bitpositions* [bitposition]
          |      |  +---- bitposition    bier-te:bit-string
          |      +---- fec-stack-type* [si]
          |      |  +---- si    bier:si
          |      |      +---- fec-stack-info* [bitposition]
          |      |      +---- bitposition    bier-te:bit-string
          |      |      +---- (fec-stack-type)
          |      |          |  +---- local-bfr?    inet:ip-address
          |      |          |  +---- local-interface?    if:interface-ref
          |      +---- (routed)
          |      |  +---- routed-bfr?    inet:ip-address
          |      +---- (local-decap)
          |      |  +---- bfer?    inet:ip-address
          +---- reply-mode-te?    reply-mode
augment /coam-methods:path-discovery/coam-methods:input/coam-methods:destination-tp:
  +----- bier-address
      |  +---- (bier-address)?
      |      |  +---- bier-bsl?    bier:bsl
      |      |  +---- bier-subdomainid?    bier:sub-domain-id
      |      |  +---- bfir?    bier:bfr-id
      |      |  +---- bier-bfers* [bier-bfrid]
--- reply-mode?          reply-mode
  +---- bier-te-address
    +---- bier-te-subdomainid?  bier:sub-domain-id
    +---- bier-te-bp-info* [bier-te-bsl]
      +---- bier-te-bsl  bier:bsl
        +---- bitstring* [si]
          +---- si  bier:si
          +---- bitpositions* [bitposition]
            +---- bitposition  bier-te:bit-string
        +---- fec-stack-type* [si]
          +---- si  bier:si
          +---- fec-stack-info* [bitposition]
            +---- bitposition  bier-te:bit-string
        +---- (fec-stack-type)
          +--:(connected)
            +---- local-bfr?  inet:ip-address
            +--:(routed)
              +---- routed-bfr?  inet:ip-address
              +--:(local-decap)
                +---- bfer?  inet:ip-address
      +---- reply-mode-te?          reply-mode
augment /coam-methods:path-discovery/coam-methods:output/coam-methods:dest-test-point:
  +---- bier-address
    +---- (bier-address)?
      +--:(bfrs)
        +---- bier-bsl?  bier:bsl
        +---- bier-subdomainid?  bier:sub-domain-id
        +---- bfrid?  bier:bfr-id
        +---- bier-bfers* [bier-bfrid]
          +---- bier-bfrid  bier:bfr-id
      +--:(m-addr)
        +---- vpn-id?  uint32
        +---- source-address?  inet:ip-address
        +---- source-wildcard?  uint8
        +---- group-address?  inet:ip-address
        +---- group-wildcard?  uint8
        +---- target-bfers* [bier-bfrid]
          +---- bier-bfrid  bier:bfr-id
      +---- reply-mode?          reply-mode
    +---- bier-te-address
      +---- bier-te-subdomainid?  bier:sub-domain-id
      +---- bier-te-bp-info* [bier-te-bsl]
        +---- bier-te-bsl  bier:bsl
        +---- bitstring* [si]
          +---- si  bier:si
          +---- bitpositions* [bitposition]
            +---- bitposition  bier-te:bit-string
Internet-Draft               BIER RPCs YANG                  August 2018

|  +---- fec-stack-type* [si]  bier:si
|  +---- si  bier:si
|  +---- fec-stack-info* [bitposition]
|  +---- bitposition  bier-te:bit-string
|  +---- {fec-stack-type}
|  +---- connected  inet:ip-address
|  +---- local-bfr?  inet:ip-address
|  +---- local-interface?  if:interface-ref
|  +---- routed  inet:ip-address
|  +---- routed-bfr?  inet:ip-address
|  +---- (local-decap)
|  +---- bfer?  inet:ip-address
|  +---- reply-mode-te?  reply-mode
|  +---- af
|  +---- reply-mode-te?  reply-mode
|  +---- af
|  +---- af

augment /coam-methods:path-discovery/coam-methods:output/coam-methods:path-trace-info/coam-methods:path-trace-info-list:
|  +---- af
|  +---- af
|  +---- af
|  +---- af
|  +---- af

Chen & Gu                 Expires March 1, 2019                 [Page 7]
3. Overview the bier rpcs model

There are seven parts of the bier rpcs data model.

- This Module augments the "coam-methods:continuity-check/coam-methods:input/coam-methods:destination-tp:" with a bier-address container and bier-te address container. The Containers define all the input parameters related to 'continuity-check' operation.

- This Module augments the "coam-methods:continuity-check/coam-methods:output/coam-methods:src-test-point:" with a bier-address container and bier-te address container. The Containers define all the output src-test-point parameters related to 'continuity-check' operation.

- This Module augments the "coam-methods:continuity-check/coam-methods:output/coam-methods:dest-test-point:" with a bier-address container and bier-te address container. The Containers define all the output dest-test-point parameters related to 'continuity-check' operation.

- This Module augments the "coam-methods:path-discovery/coam-methods:input/coam-methods:destination-tp:" with a bier-address container and bier-te address container. The Containers define all the input parameters related to 'path-discovery' operation.

- This Module augments the "coam-methods:path-discovery/coam-methods:output/coam-methods:src-test-point:" with a bier-address container and bier-te address container. The Containers define all the output src-test-point parameters related to 'path-discovery' operation.

- This Module augments the "coam-methods:path-discovery/coam-methods:output/coam-methods:dest-test-point:" with a bier-address container and bier-te address container. The Containers define all the output dest-test-point parameters related to 'path-discovery' operation.
This Module augments the "coam-methods:path-discovery/coam-methods:output/coam-methods:path-trace-info/coam-methods:path-trace-info-list" with a bier-address container, bier-te address container and bier-response container. The Containers define all the output path trace information list related to 'path-discovery' operation.

4. BIER YANG Data Model

<CODE BEGINS> file "ietf-bier-rpcs@2018-02-08.yang"
module ietf-bier-rpcs {
    prefix bier-rpcs;

    import ietf-bier {
        prefix bier;
    }

    import ietf-bier-te{
        prefix bier-te;
    }

    import ietf-connectionless-oam-methods {
        prefix coam-methods;
    }

    import ietf-inet-types {
        prefix "inet";
    }

    import ietf-interfaces {
        prefix if;
    }

    import ietf-multicast-information {
        prefix multicast-info;
    }

    organization
    "IETF BIER(Bit Indexed Explicit Replication ) Working Group";

    contact
    "WG List: <mailto:bier@ietf.org>
    WG Chair: Tony Przygienda
    <mailto:tonysietf@gmail.com>"
typedef reply-mode {
    type enumeration {
        enum "do not Reply" {
            value 1;
            description "reply mode is do not reply";
        }
        enum "reply via IPv4/IPv6 UDP packet" {
            value 2;
            description "reply mode is via IPv4/IPv6 UDP packet";
        }
        enum "reply via BIER-TE packet" {
            value 3;
            description "reply mode is via BIER-TE packet";
        }
    }
} 

description "Reply mode";

grouping general-multicast {
    description "The general multicast address information";
}
leaf source-address {
    type inet:ip-address;
    description "The address of multicast source. The value set to zero means that the receiver interests in all source that relevant to one group.";
}
leaf source-wildcard {
    type uint8;
    description "The wildcard information of source.";
}
leaf group-address {
    type inet:ip-address;
    description "The address of multicast group.";
}
leaf group-wildcard {
    type uint8;
    description "The wildcard information of group.";
}

grouping m-addr {
    description "The vpn multicast information.";
    leaf vpn-id {
        type uint32;
        description "The vpn-id of the multicast flow. If there is global instance, the vpnid value should be zero.";
    }
    uses general-multicast;
}

grouping bier-address {
    description "bier test point address.";
    choice bier-address {
        case bfrs {
            description "use bfrs as address.";
            leaf bier-bsl {
                type bier:bsl;
                description "bier bitstringlength.";
            }
            leaf bier-subdomainid {
                type bier:sub-domain-id;
                description "bier sub-domain id.";
            }
        }
    }
}
leaf bfir {
    type bier:bfr-id;
    description "bier bfir id.";
}
list bier-bfers {
    key bier-bfrid;
    leaf bier-bfrid {
        type bier:bfr-id;
        description "bier bfr identifier.";
    }
    description "bier BFERs.";
}
}
case m-addr {
description "use multicast flow address as address.";
uses multicast-info:m-addr;
}
description "BIER address." ;
}
list target-bfers {
    key bier-bfrid;
    leaf bier-bfrid {
        type bier:bfr-id;
        description "bier bfr identifier.";
    }
    description "target BFERs.";
}
leaf reply-mode{
    type reply-mode;
    description "reply mode.";
}
}
grouping bier-te-address{
description "bier-te test point address.";
leaf bier-te-subdomainid {
    type bier:sub-domain-id;
    description
"bier-te sub-domain id.";
}
list bier-te-bp-info {
    key "bier-te-bsl";
    description "bier-te bitpositions information.";
    leaf bier-te-bsl {
        type bier:bsl;
        description "bier bitstringlength.";
    }
}
list bitstring {
    key "si";
    description "The bitstring of BIER-TE path.";
    leaf si {
        type bier:si;
        description "The set identifier of this forwarding item.";
    }
}
list bitpositions {
    key "bitposition";
    description "the list of bitposition information.";
    leaf bitposition {
        type bier-te:bit-string;
        description "the bitposition information.";
    }
}
list fec-stack-type {
    key "si";
    description "The FEC stack info.";
    leaf si {
        type bier:si;
        description "The set identifier of this forwarding item.";
    }
}
list fec-stack-info {
    key "bitposition";
    description "the choice of fec stack type.";
    leaf bitposition {
        type bier-te:bit-string;
        description "the bitposition information.";
    }
}
choice fec-stack-type{
  mandatory true;
  case connected {
    description "The type of adjacency is connected. Mostly connected interfaces.";
    leaf local-bfr{
      type inet:ip-address;
      description "the local bfr identifier.";
    }
    leaf local-interface{
      type if:interface-ref;
      description "the local interface.";
    }
  }
  case routed {
    description "The type of adjacency is routed. Mostly not connected interfaces.";
    leaf routed-bfr{
      type inet:ip-address;
      description "the loopback address of bfr.";
    }
  }
  case local-decap {
    description "Means that the packet should be decapsulated and forward out BIER domain.";
    leaf bfer{
      type inet:ip-address;
      description "bfer identifier.";
    }
  }
}

leaf reply-mode-te{
  type reply-mode;
  description "the reply mode.";
}
identity tp-address-technology-type {
  description
  "Test point address type";
}

identity bier-address-type {
  base tp-address-technology-type;
  description
  "bier address address type";
}

identity bier-te-address-type {
  base tp-address-technology-type;
  description
  "bier te address address type";
}

augment /coam-methods:continuity-check/coam-methods:input/coam-methods:destination-tp{
  description
  "augment the connectionless oam methods yang for input CC destination test point address";
  container bier-address {
    uses bier-address;
    description
    "bier Address.";
  }
}

container bier-te-address {
  uses bier-te-address;
  description
  "bier te Address.";
}

augment /coam-methods:continuity-check/coam-methods:output/coam-methods:src-test-point{
  description
  "augment the connectionless oam methods yang for output CC source test point address";
  container bier-address {
    uses bier-address;
    description
    "bier Address.";
  }
}

container bier-te-address {
  uses bier-te-address;
}
augment /coam-methods:continuity-check/coam-methods:output/coam-methods:dest-test-point{
    description
    "augment the connectionless oam methods yang for output CC destination test point address";
    container bier-address {
        uses bier-address;
        description
        "bier Address.";
    }
    container bier-te-address {
        uses bier-te-address;
        description
        "bier te Address.";
    }
}
}

augment /coam-methods:path-discovery/coam-methods:input/coam-methods:destination-tp {
    description
    "augment the connectionless oam methods yang for input path-discovery destination test point address";
    container bier-address {
        uses bier-address;
        description
        "bier Address.";
    }
    container bier-te-address {
        uses bier-te-address;
        description
        "bier te Address.";
    }
}

augment /coam-methods:path-discovery/coam-methods:output/coam-methods:src-test-point{
    description
    "augment the connectionless oam methods yang for output path-discovery src test point address";
    container bier-address {
        uses bier-address;
        description
        "bier Address.";
    }
    container bier-te-address {
        uses bier-te-address;
        description
        "bier te Address.";
    }
}
"augment "/coam-methods:path-discovery/coam-methods:output/coam-methods:dest-test-point"{

description
"augment the connectionless oam methods yang for output path-discovery destination test point address";
container bier-address {
  uses bier-address;
  description
  "bier Address.";
}
}

"augment /coam-methods:path-discovery/coam-methods:output/coam-methods:path-trace-info/coam-methods:path-trace-info-list{

description
"augment the connectionless oam methods yang for output path trace info.";
container bier-address {
  uses bier-address;
  description
  "bier Address.";
}
}

container bier-te-address {
  uses bier-te-address;
  description
  "bier te Address.";
}

container bier-te-address {
  uses bier-te-address;
  description
  "bier te Address.";
}

container bier-te-address {
  uses bier-te-address;
  description
  "bier te Address.";
}

container bier-response{ 
  leaf ttl{ 
    type int16; 
    description
    "TTL is ";
  }
  leaf responder-bfr{ 
    type inet:ip-address; 
  }
}
5. Security Considerations

TBD.

6. Acknowledgements

We would like to thank IJsbrand Wijnands, Reshad Rahman and Giles Heron for their comments and support of this work.

7. IANA Considerations

This document requires no IANA Actions. Please remove this section before RFC publication.

8. Normative references

[I-D.ietf-bier-architecture]

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

[I-D.ietf-bier-te-arch]
[I-D.ietf-lime-yang-connectionless-oam-methods]
Kumar, D., Wang, Z., Wu, Q., Rahman, R., and S. Raghavan,
"Retrieval Methods YANG Data Model for the Management of
Operations, Administration, and Maintenance (OAM)
Protocols that use Connectionless Communications", draft-
ietf-lime-yang-connectionless-oam-methods-13 (work in
progress), November 2017.

the Network Configuration Protocol (NETCONF)", RFC 6020,
DOI 10.17487/RFC6020, October 2010,

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

RFC 6991, DOI 10.17487/RFC6991, July 2013,

[RFC7223] Bjorklund, M., "A YANG Data Model for Interface
Management", RFC 7223, DOI 10.17487/RFC7223, May 2014,

Authors’ Addresses

Ran Chen
ZTE Corporation
No.50 Software Avenue, Yuhuatai District
Nanjing, Jiangsu Province  210012
China

Phone: +86 025 88014636
Email: chen.ran@zte.com.cn

Min Gu
ZTE Corporation
No.50 Software Avenue, Yuhuatai District
Nanjing, Jiangsu Province  210012
China