Abstract

Bit Index Explicit Replication IPv6 encapsulation (BIERv6) introduces an approach to use IPv6 extension header to carry BIER header with IPv6 unicast address as destination address. It provides the ability to replicate a packet from one router to another router in a different domain as well as in the same domain. This document introduces the techniques for multicast deployment across multiple domains using BIERv6, and demonstrate how BIERv6 is beneficial for such deployment.

Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119] and [RFC8174].

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 January 5, 2020.
1. Introduction

Bit Index Explicit Replication [RFC8296] IPv6 encapsulation (BIERv6) described in [I-D.xie-bier-ipv6-encapsulation] introduces an approach to use IPv6 extension header to carry BIER header. One BIERv6 option, using IPv6 unicast address as destination address provides the ability to replicate a packet from one router to another router in a different domain as well as in the same domain. This document introduces the techniques for multicast deployment across multiple domains using BIERv6, and demonstrates how BIERv6 is beneficial for such deployment.

2. Terminology

Readers of this document are assumed to be familiar with the terminology and concepts of the documents listed as Normative References.
3. Inter-domain Multicast Overview

It is common to deploy multicast services across multiple domains.

One typical scenario for this type of deployment is in a service-provider network for MVPN service as described in [I-D.ietf-bier-ipv6-requirements]. Service provider network tends to be very heterogeneous with full-mesh backbone network, and metro networks with fabric for dense area coverage or ring-shaped for sparse area coverage. The backbone network and metro networks are autonomous systems interconnected by border routers (BRs). Multicast-based delivery of video need to be set up from a source router on the backbone to each of the boundary routers of each metro network.

This scenario may have some variant. For example, multicast source router is a Top of Rack (TOR) switch in a service provider data center (SPDC) connected to backbone with data center gateway(s) (DC-GW), and multicast receiver is the home broadband subscribers connected to boundary routers (e.g. BNG) of each metro network. Operators may want to set up multicast-based delivery from TOR to BNGs seamlessly without segmentation or stitching on DC-GW(s) or BR(s).

It is described as hierarchical multicast in this document.

Another typical scenario for inter-domain multicast deployment is in peering network as described in [RFC8313] to set up multicast-based delivery of content across inter-domain peering points.

This scenario may have some variant. For example, interconnected content delivery networks (CDNs) (described in [RFC6770]) owned by Network Service Providers (NSPs) or Enterprise Service Providers may need to deliver multicast from one to others.

It is described as peering multicast in this document.

4. Inter-domain Multicast Deployment using BIERv6

4.1. Hierarchical Multicast

Following is an example of hierarchical deployment of multicast.
Figure 1: Inter-Domain Hierarchical Multicast

Multicast source is connected to PE1x, and multicast receivers are connected to PE2x and PE3x.

PE1x, PE2x, PE3x is located in Backbone (AS 64001), Metro 2 (AS 64002), and Metro 3 (AS 64003) respectively, and BR1, BR2, BR3 is boarder of these three domains. They belong to a single administrative domain.

IGP underlay for BIERv6 is deployed in Metro2, Metro3 respectively. The bfr-ids in Metro2 and Metro 3 should be divided rationally.

PE1x, PE2x, PE3x uses 2001::E1, 2001::E2, 2001::E3 as IPv6 BFR-prefix (and End.BIER function) respectively.

BR1, BR2, BR3 uses 2001::B1, 2001::B2, 2001::B3 as IPv6 BFR-prefix (and End.BIER function) respectively.

All of them use the Non-MPLS static BSL-SD-SI BIFT encoding method described in [I-D.ietf-bier-non-mpls-bift-encoding] as the auto-generation method.
On BR1, static configuration can be used to construct inter-domain BIERv6 forwarding table.

```
bier sub-domain 6 ipv6-underlay
  bfr-prefix 2001::B1
  bfr-id 0
  encapsulation ipv6 bsl 256 max-si 2
  static-bift
    nexthop 2001::B2 bfr-id 1 to 256
    nexthop 2001::B3 bfr-id 257 to 512
```

Accordingly, the following BIFTs will be constructed:

- BIFT correspond to SD<6>/BSL<256>/SI<0> (neighbor = 2001::B2, F-BM = ffff....ffff)
- BIFT correspond to SD<6>/BSL<256>/SI<1> (neighbor = 2001::B3, F-BM = ffff....ffff)

On PE1x, static configuration can be used to construct inter-domain BIERv6 forwarding table.

```
bier sub-domain 6 ipv6-underlay
  bfr-prefix 2001::E1
  bfr-id 0
  encapsulation ipv6 bsl 256 max-si 2
  static-bift
    nexthop 2001::B1 bfr-id 1 to 512
```

Accordingly, the following BIFTs will be constructed:

- BIFT correspond to SD<6>/BSL<256>/SI<0> (neighbor = 2001::B1, F-BM = ffff....ffff)
- BIFT correspond to SD<6>/BSL<256>/SI<1> (neighbor = 2001::B1, F-BM = ffff....ffff)

Use of BGP as inter-domain underlay protocol to advertise the BIER information from BR2 or BR2 to BR1, or from BR1 to PE1x is outside the scope of this document.

On each domain, two redundant border routers may be deployed, and anycase IPv6 address can be used on each pair of BRs as BFR-prefix. Inter-Domain BIER will converge normally when unicast converge and the BIFT will be reconstructed accordingly.

For multicast overlay layer, there are no extensions needed. MVPN is deployed on PE1x, PE2x and PE3x using sub-domain 6 and bsl 256 without segmentation on border router(s).
Note: Use of the IPv6 address configured on PE1 to identify an MVPN instance can eliminate the need for BFR-id configuration on PE1x, which otherwise has to be configured from the space of a sub-domain.

4.2. Peering Multicast

Following is an example of peering deployment of multicast.

AD = Administrative Domain (independent autonomous system)
BR = Border Router
SRC = Multicast Source
RCV = Multicast Receiver

Figure 2: Inter-Domain Peering Multicast

Each Administrative Domain AD-1, AD-2 or AD-3 is configured a unique color. Color 1, 2, 3 are used in this example.

For routing underlay layer, the ingress router uses IGP protocol (IS-IS as example in this document) for the domain it belongs to, and uses static configuration for the domain it doesn’t belong to.

Below is an example of routing underlay configuration on PE1x:
# PE1x routing underlay layer configuration
bier sub-domain 6 ipv6-underlay
  bfr-prefix 2001::E1
  bfr-id 1
  encapsulation ipv6 bsl 256 max-si 1
color 1 protocol isis
color 2 static-bift
  next-hop 2001::B2 bfr-id 1 to 512
color 3 static-bift
  next-hop 2001::B3 bfr-id 1 to 256

The following lists the BIFT that will be constructed on PE1x:

BIFT corresponding to SD<6>/BSL<256>/SI<0> for color 1 ;;Ref1
BIFT corresponding to SD<6>/BSL<256>/SI<0> for color 2 ;;Ref2
BIFT corresponding to SD<6>/BSL<256>/SI<1> for color 2 ;;Ref3
BIFT corresponding to SD<6>/BSL<256>/SI<0> for color 3 ;;Ref4

Ref1: BIFT constructed using IGP.
Ref2: BIFT constructed using static configuration, with BR2 a multi-hop BFR neighbor of PE1x.
Ref3: BIFT constructed using static configuration, with BR2 a multi-hop BFR neighbor of PE1x.
Ref3: BIFT constructed using static configuration, with BR3 a multi-hop BFR neighbor of PE1x.

For multicast overlay layer, the color extended community defined in [RFC5512] is carried in Leaf A-D route together with the PTA attribute.

(1) PE in each domain gets the color it belongs to. This can be done by configuration on each PE in each domain.

(2) PE carries a color attribute in BGP-MVPN Leaf A-D route when advertising to Ingress PE as response to explicit-tracking initiated by the Ingress PE. This can be done by configuration on MVPN deployment. Refer to [I-D.xie-bier-ipv6-mvpn] for other attributes needed to be used.

(3) The Ingress PE gets the Leaf A-D route, learns the BFERs of a color (representing a domain) interested in a multicast flow, and constructs the overlay forwarding table. Below is an example of the overlay forwarding table on PE1x:
Ref1: packet will be replicated according to the BitString<0001> and the BIFT constructed using the IGP for SD<6>/BSL<256>/SI<0> for color 1.

Ref2: packet will be replicated according to the BitString<0001> and the BIFT constructed using the static-bift configuration for SD<6>/BSL<256>/SI<0> for color 2.

Ref3: packet will be replicated according to the BitString<0001> and the BIFT constructed using the static-bift configuration for SD<6>/BSL<256>/SI<1> for color 2.

Ref3: packet will be replicated according to the BitString<0001> and the BIFT constructed using the static-bift configuration for SD<6>/BSL<256>/SI<1> for color 3.

Note: BFR-id configuration on PE1x is only necessary when PE1x will act as BFER, for example, there is multicast packet from PE2x to PE1x. The BFR-ids in color 1, 2, 3 is independent on each other.

5. Security Considerations

The procedures of this document do not, in themselves, provide privacy, integrity, or authentication for the control plane or the data plane.

6. IANA Considerations

No IANA Allocation is required in this document.

7. Acknowledgements

TBD.

8. References

8.1. Normative References

[I-D.ietf-bier-ipv6-requirements]
8.2. Informative References


Authors’ Addresses

Liang Geng
China Mobile
Beijing 10053

Email: gengliang@chinamobile.com

Jingrong Xie
Huawei Technologies

Email: xiejingrong@huawei.com

Mike McBride
Futurewei

Email: mmcbride7@gmail.com

Gang Yan
Huawei Technologies

Email: yangang@huawei.com
Abstract

Bit Index Explicit Replication (BIER) is an architecture that provides multicast forwarding through a "BIER domain" without requiring intermediate routers to maintain multicast related per-flow state. Neither does BIER require an explicit tree-building protocol for its operation. A multicast data packet enters a BIER domain at a "Bit-Forwarding Ingress Router" (BFIR), and leaves the BIER domain at one or more "Bit-Forwarding Egress Routers" (BFERs). The BFIR router adds a BIER header to the packet. Such header contains a bit-string in which each bit represents exactly one BFER to forward the packet to. The set of BFERs to which the multicast packet needs to be forwarded is expressed by the according set of bits switched on in BIER packet header.

This document describes the procedure needed for mLDP tunnels to be signaled over and stitched through a BIER core, allowing LDP routers to run traditional Multipoint LDP services through a BIER core.

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), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.
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." The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html

This Internet-Draft will expire on October 8, 2017.

Copyright Notice

Copyright (c) 2019 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.

Table of Contents

1. Introduction ........................................ 3
2. Conventions used in this document ................. 3
   2.1. Definitions .................................. 3
3. mLDP Signaling Through BIER domain ............... 4
   3.1. Ingress BBR procedure ........................ 5
   3.1.1. Automatic mLDP session creation ............ 5
   3.2. EBBR procedure method ........................ 6
4. Datapath Forwarding ............................... 6
   4.1. Datapath traffic flow ........................ 6
5. Recursive FEC ..................................... 7
6. IANA Considerations ............................... 7
7. Security Considerations ........................... 7
8. References ....................................... 7
   8.1. Normative References ......................... 7
   8.2. Informative References ....................... 7
7. Acknowledgments ................................... 7
Authors’ Addresses ................................ 7
1. Introduction

Some operators that are using mLDP P2MP LSPs for their multicast transport would like to deploy BIER technology in some segment of their network. This draft explains a method to signal mLDP services and stitch it to a BIER domain, with minimal disruption and operational impact to the mLDP domain.

2. Conventions used in this document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

2.1. Definitions

Some of the terminology specified in [I-D.draft-ietf-bier-architecture-05] is replicated here and extended by necessary definitions:

**BIER:**

Bit Index Explicit Replication (The overall architecture of forwarding multicast using a Bit Position).

**BFR:**

Bit Forwarding Router (A router that participates in Bit Index Multipoint Forwarding). A BFR is identified by a unique BFR-prefix in a BIER domain.

**BFIR:**

Bit Forwarding Ingress Router (The ingress border router that inserts the Bit Map into the packet). Each BFIR must have a valid BFR-id assigned. BFIR is term used for dataplain packet forwarding.

**BFER:**

Bit Forwarding Egress Router. A router that participates in Bit Index Forwarding as leaf. Each BFER must be a BFR. Each BFER must have a valid BFR-id assigned. BFIR is term used for dataplain packet forwarding.

**BBR:**

BIER Boundary router. The router between the LDP domain and
BIER domain.

IBBR:
Ingress BIER Boundary Router. The ingress router from signaling point of view. It maintains mLDP adjacency toward the LDP domain and determines if the mLDP FEC needs to be signaled across the BIER domain via targeted ldp.

EBBR:
Egress BIER Boundary Router. The egress router in BIER domain from signaling point of view. It terminates the targeted ldp signaling through BIER domain. It also keeps track of all IBBRs that are part of this p2mp tree

BFT:
Bit Forwarding Tree used to reach all BFERs in a domain.

BIFT:
Bit Index Forwarding Table.

BIER sub-domain:
A further distinction within a BIER domain identified by its unique sub-domain identifier. A BIER sub-domain can support multiple BitString Lengths.

BFR-id:
An optional, unique identifier for a BFR within a BIER sub-domain.

3. mLDP Signaling Through BIER domain
As per figure 1, point-to-multipoint and multipoint-to-multipoint LSPs established via mLDP [RFC6388] can be signaled through a BIER domain via targeted LDP sessions. This procedure is explained in [RFC7060] (Using LDP Multipoint Extension on Targeted LDP Sessions).

This document provides some details and defines some needed procedures.

3.1 Ingress BBR procedure

The Ingress BBR (IBBR) is connected to the mLDP on one side and a BIER domain on the other side. To connect the LDP domains via BIER domain IBBR needs to be establish targeted LDP session with EBBR closest to the root of the P2mp or mp2mp LSP. To do so IBBR will follow procedures in [RFC7060] in particular the section "6. targeted mLDP with Multicast Tunneling".

The target LDP session can be established manually via configuration or via automated mechanism.

3.1.1 Automatic tLDP session creation

A tLDP session can be generated automatically from every IBBR to EBBR when a mLDP FEC arrives on the IBBR. The Root node address in the mLDP FEC can be used to find the EBBR. To identify the EBBR same procedures as [RFC7060] section 2.1 can be used or the procedures as explained in the [draft-ietf-bier-pim-signaling] appendix A. After finding the IBBR the tLDP session can be initiated from the IBBR to EBBR.
3.2. EBBR procedure method

The Egress BBR (EBBR) is connected to the mLDP domain which the root of the P2MP or MP2MP LSP resides on. The EBBR should accept the tLDP session and assign a upstream assigned label for arriving FEC.

The EBBR should follow the [RFC7060] procedures with following modifications:

- The label assigned by EBBR cannot be Implicit Null. This is to ensure that identity of each p2mp and/or mp2mp tunnel in BIER domain is uniquely distinguished.

- The label can be assigned from a domain-wide Common Block (DCB) [I-D.zhang-bess-mvpn-evpn-aggregation-label], as well as upstream assigned.

- The Interface ID TLV [RFC6389] includes a new BIER sub-domain sub-tlv (type TBD)

With same token the EBBR should track all the arriving FECs and the IBBRs that are generating these FECs. EBBR will use this information to build the bier header for each set of common FEC arriving from the IBBRs.

4. Datapath Forwarding

4.1. Datapath traffic flow

On BFIR when the MPLS label for P2MP/MP2MP LSP arrives a lookup in ILM table is done and the label is swapped with tLDP upstream assigned label. The BFIR will note all the BFERs that are interested in specific p2mp/mp2mp LSP (as per section 3.2). BFIR will put the corresponding BIER header with bit index set for all IBBRs interested in this P2MP LSP. BFIR will set the BIERHeader.Proto = MPLS and will forward the BIER packet into BIER domain.

In the BIER domain normal BIER forwarding procedure will be done, as per [RFC 8279]

The IBBRs will receive the BIER packet, will look at the protocol of BIER header (MPLS). BFER will remove the BIER header and will do a lookup in the ILM table for the upstream assigned label and perform its corresponding action.

It should be noted that these procedures are valid if BFIR is the ILER and/or BFER is the ELER as per [RFC 7060]
5. Recursive FEC

The above procedures also will work with a mLDP recursive FEC. The root used to determine the EBBR is the outer root of the FEC. The entire recursive FEC needs to be preserve when it is forwarded via tLDP and the label request.

6. IANA Considerations

This document contains no actions for IANA.

7. Security Considerations

TBD

8. References

8.1. Normative References


8.2. Informative References


7. Acknowledgments <Add any acknowledgements>

Authors’ Addresses

Hooman Bidgoli (editor)
Nokia
600 March Rd.
Ottawa, Ontario K2K 2E6
Canada
Abstract

Point to multipoint (P2MP) BFD is designed to verify multipoint connectivity. This document specifies the application of P2MP BFD in BIER network.

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 January 4, 2020.

Copyright Notice

Copyright (c) 2019 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

Quan Xiong, et al. Expires January 4, 2020 [Page 1]
1. Introduction

Bit Index Explicit Replication (BIER) [RFC8279] provides forwarding of multicast data packets through a multicast domain. It does so without requiring any explicit tree-building protocol and without requiring intermediate nodes to maintain any per-flow state.

[ RFC8562] defines a method of using Bidirectional Forwarding Detection (BFD) to monitor and detect unicast failures between the sender (head) and one or more receivers (tails) in multipoint or multicast networks. [RFC8563] describes active tail extensions to the BFD protocol for multipoint networks.

This document describes the procedures for using such mode of BFD protocol to monitor connectivity between a multipoint sender, Bit-Forwarding Ingress Router (BFIR), and a set of one or more multipoint receivers, Bit-Forwarding Egress Routers (BFERs). The BIER BFD only supports the unidirectional multicast. This document defines the use of P2MP BFD as per [RFC8562], and active tail as per [RFC8563] for BIER-specific domain.
2. Conventions used in this document

2.1. Terminology

This document uses the acronyms defined in [RFC8279] along with the following:

BFD: Bidirectional Forwarding Detection.

OAM: Operations, Administration, and Maintenance.

P2MP: Point to Multi-Point.

2.2. 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.

3. BIER BFD Encapsulation

BIER BFD encapsulation uses the BIER OAM packet format defined in [I-D.ietf-bier-ping]. The value of the Msg Type field MUST be set to BIER BFD (TBD1 by IANA). BFD Control Packet, defined in Section 4 [RFC5880] immediately follows the BIER OAM header. The operation of Multipoint BFD with the BFD Control Packet is described in [RFC8562].

4. BIER BFD Session Bootstrapping

As defined in [RFC8562], BIER BFD session MAY be established to monitor the state of the multipoint path. The BIER BFD session could be created for each multipoint path and the set of BFERs over which the BFIR wishes to run BIER BFD. The BFIR MUST advertise the BFD Discriminator along with the corresponding multipoint path to the set of BFERs. Bootstrapping a BIER BFD session MAY use BIER OAM message section 4.1 or the control plane section 4.2.

The BIER BFD bootstrapping MUST be repeated when the value of this discriminator being changed.

4.1. BIER OAM Bootstrapping

The BIER OAM could be used for bootstrapping the BIER BFD session. The BFIR sends the BIER OAM Echo request message carrying a BFD discriminator TLV which immediately follows the Target SI-Bitstring TLV (section 3.3.2 [I-D.ietf-bier-ping]).
The Target SI-Bitstring TLV MUST be used to carry the set of BFER information (including Sub-domain-id, Set ID, BS Len, Bitstring) for the purpose of session establishment.

The BFD discriminator TLV is a new TLV for BIER OAM TLV with the type (TBD2 by IANA) and the length of 4. The value contains the 4-byte local discriminator generated by BFIR for this session. This discriminator MUST subsequently be used as the My Discriminator field in the BIER BFD session packets sent by BFIR. The format is as follows.

```
0                   1                   2                   3
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          Type=TBD2            |             Length            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        My Discriminator                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 1: BFD discriminator TLV

4.2. IGP protocol Bootstrapping

An alternative option to bootstrap the BIER BFD is to advertise the BFD information in control plane. This document defines a new BIER BFD Sub-sub-TLV carried in IS-IS and OSPF protocol.

The BFIR generates the My Discriminator value for each multicast flow and advertises it to the expecting BFERs which is indicated by the Bitstring which is carried in BIER BFD sub-sub-TLV. The corresponding BFERs SHOULD store the My Discriminator value for packet Demultiplexing.

4.2.1. IS-IS extension for BIER BFD

The new BIER BFD Sub-sub-TLV is carried within the BIER Info sub-TLV defined in [RFC8401]. The format is as follows.
Type: TBD3 by IANA.

Length: Length of the BIER BFD Sub-sub-TLV for IS-IS extension, in bytes.

My Discriminator: A unique, nonzero discriminator value generated by BFIR for each multipoint path.

The BitString field carries the set of BFR-IDs of BFER(s) that the BFIR expects to establish BIER BFD session.

The BIFT-id represents a particular Bit Index Forwarding Table (BIFT) as per [RFC8279].

4.2.2. OSPF extension for BIER BFD

The new BIER BFD Sub-TLV is a sub-TLV of the BIER Sub-TLV defined in [RFC8444]. The format is as follows.
Figure 3: BIER BFD Sub-TLV for OSPF extension

Type: TBD4 by IANA.

Length: Length of the BIER BFD Sub-TLV for OSPF extension, in bytes.

Other fields in BIER BFD Sub-TLV is the same with section 4.2.1.

5. Discriminators and Packet Demultiplexing

As defined in [RFC8562], the BFIR sends BFD Control packets over the multipoint path via the BIER BFD session with My Discriminator set to the value assigned by the BFIR and the value of the Your Discriminator set to zero. The set of BFERs MUST demultiplex BFD packets based on a combination of the source address, My Discriminator value. The source address is BFIR-id and BIER MPLS Label (MPLS network) or BFIR-id and BIFT-id (Non-MPLS network) for BIER BFD. The My Discriminator value is advertised in BIER BFD bootstrapping using one of options described in section 4.

6. Active Tail in BIER BFD

[RFC8563] defined an extension for Multipoint BFD, which allows the head to discover the state of a multicast distribution tree for any sub-set of tails. For BIER BFD in active tail mode, the BFIR may learn the state and connectivity of the BFERs. As per [RFC8563], the BFIR uses a combination of multicast Poll sequence messages and unicast Poll messages. The unicast messages must be sent over the path which is disjoint from the multicast distribution tree.
7. Security Considerations

For BIER OAM packet processing security considerations, see [I-D.ietf-bier-ping].

For general multipoint BFD security considerations, see [RFC8562].

No additional security issues are raised in this document beyond those that exist in the referenced BFD documents.

8. Acknowledgements

Authors would like to thank the comments and suggestions from Sandy Zhang, Jeffrey (Zhaohui) Zhang, Donald Eastlake 3rd.

9. IANA Considerations

IANA is requested to assign new type from the BIER OAM Message Type registry as follows:

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>TBD1</td>
<td>BIER BFD</td>
<td>[this document]</td>
</tr>
<tr>
<td>TBD2</td>
<td>BFD discriminator TLV</td>
<td>[this document]</td>
</tr>
<tr>
<td>TBD3</td>
<td>BIER BFD Sub-sub-TLV for IS-IS</td>
<td>[this document]</td>
</tr>
<tr>
<td>TBD4</td>
<td>BIER BFD Sub-TLV for OSPF</td>
<td>[this document]</td>
</tr>
</tbody>
</table>

Table 1

10. References

10.1. Normative References

[I-D.ietf-bier-ping]


Quan Xiong, et al. Expires January 4, 2020
10.2. Informative References


Authors' Addresses

Quan Xiong, et al. Expires January 4, 2020 [Page 8]
BIER IPv6 Requirements
draft-ietf-bier-ipv6-requirements-01

Abstract

The BIER WG has a charter item to work on mechanisms which use BIER natively in IPv6. This document is intended to help the WG with this effort by specifying requirements for transporting packets, with Bit Index Explicit Replication (BIER) headers, in an IPv6 environment. There will be a need to send IPv6 payloads, to multiple IPv6 destinations, using BIER. There have been several proposed solutions in this area. But there hasn’t been a document which describes the problem and lists the requirements. The goal of this document is to describe the BIER IPv6 requirements and summarize the pro’s and con’s of the proposed solutions.

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 January 2, 2020.

Copyright Notice

Copyright (c) 2019 IETF Trust and the persons identified as the document authors. All rights reserved.
1. Introduction

Bit Index Explicit Replication (BIER) [RFC8279] is an architecture that provides optimal multicast forwarding, without requiring intermediate routers to maintain per-flow state, through the use of a
multicast-specific BIER header. [RFC8296] defines two types of BIER encapsulation to run on physical links: one is BIER MPLS encapsulation to run on various physical links that support MPLS, the other is non-MPLS BIER Ethernet encapsulation to run on ethernet links, with an ethertype 0xab37. This document describes using BIER in non-MPLS IPv6 environments. We explain the requirements of transporting IPv4/IPv6 multicast payloads, from an IPv6 router (BFIR) to multicast IPv6 destinations (BFERs), using BIER. This can include native IPv6 encapsulation and generic tunneling. The goal of this document is to help the BIER WG evaluate the BIER v6 requirements and solutions.

1.1. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

1.2. Terminology

- BIER: Bit Index Explicit Replication. Provides optimal multicast forwarding through adding a BIER header and removing state in intermediate routers.

- BUM: Broadcast, Unknown Unicast, Multicast. Term used to describe the three types of Ethernet modes that will be forwarded to multiple destinations

2. Problem Statement

The problem is the ability of the network to transport BUM packets, with BIER headers, in an IPv6 environment. In an IPv6 network, many deployments consider using a non-MPLS encapsulation for unicast as the data-plane. In such case, it may be expected to have a BIER IPv6 encapsulation which is compliant with various kinds of physical links, perhaps in a hop-by-hop manner, and maintain the benefit of "fast reroute" of an IPv6 tunnel. Evaluating the BIER IPv6 requirements will help determine the best solutions to address these problems.

3. BIER IPv6 Scenario’s
This basic scenario depicts the need to replicate bier packets from a BFIR to BFERs across an IPv6 core. The IPv6 environment may include a variety of link types, may be entirely IPv6, may be dual stack or any type of combination which includes IPv6. Regardless of the environment, there are times when a BIER header, including the BIER bitstring used to determine the set of BIER forwarding egress routers, will need to traverse a IPv6 domain. The ways in which BIER will function in an IPv6 environment is the problem that needs to be solved. [RFC8354] lists some good IPv6 related use cases which we will similarly reference in this document.

3.1. BIERv6 for Access Network

Access networks deliver a variety of types of multicast video traffic from the service provider’s network to the home (or Enterprise) environment and from the home towards the service provider’s network.

There will be a need to send traffic from the IPv4 access towards the service provider’s IPv6 network and vice versa. A packet could be mapped into a providers IPv6 network through the use of a BIERv6 header. The access devices would not need to know specific details about the packet to perform this mapping; instead the access device would only need to know how to process a BIER header unless there is end to end IPv6.

3.2. BIERv6 for Data Center

Some Data Center operators are transitioning their Data Center infrastructure from IPv4 to native IPv6 only, in order to cope with IPv4 address depletion and to achieve larger scale. In such environment, BIERv6, can be used to natively steer multicast data across an IPv6 data center.
3.3. BIERv6 for Core Networks

While the overall amount of traffic offered to the network continues to grow and considering that multiple types of traffic with different characteristics and requirements are quickly converging over single network architecture, the network operators are starting to face new challenges.

Some operators are currently building, or plan to build in the near future, an IPv6 only native infrastructure for their core network. Having a native BIERv6 infrastructure will help maintain simplicity of the network and reduce state versus traditional IP Multicast.

3.4. Implications for BIER in SRv6

The Source Packet Routing in Networking (SPRING) architecture describes how Segment Routing can be used to steer packets through an IPv6 or MPLS network using the source routing paradigm. [RFC8354] focuses on use cases for Segment Routing in an IPv6 only environment, something which is equally important for BIER in an IPv6 only environment.

4. Requirements

There have been several suggested requirements, on the BIER email list, which we will use to form the BIER IPv6 requirements and to help evaluate the proposed solutions:

4.1. L2 Agnostic

The solution should be agnostic to the underlying L2 data link type.

4.2. Hop by hop DA modification

The solution should not require hop-by-hop modification of the IP destination address field.

A multicast packet whose DA is multicast address does not require DA modification hop by hop when replicating the packet to the nexthop BFR.

An anycast packet whose DA is an anycast address configured on each BFRs in the domain may be another option does not require DA modification when replicating the packet to the nexthop BFR.

It is common to get the impression that BIERv6 could use multicast address, as BIER is kind of one-hop replication on each BFR in normal cases. However, as described in section 6.9 of [RFC8279], it is
useful to support Non-BIER routers within a BIER domain. From the
discussion about this document on IETF104, focus is on the advantages
of using unicast address that otherwise could not possible by using
multicast address or anycast address for the two cases: replication
from a BFR to other BFR(s) connected by Layer-3 Non-BFR router(s)
without using tunneling techniques, and replication from a BFR to
other BFR(s) connected by Layer-2 switch(es) without broadcasting or
snooping on Layer-2 switch(es) in between. Based on the natural
reachability of an IPv6 unicast address, it can support the multi-hop
replication cases as well as the one-hop replication case.

This requirement may be deprecated if unicast address is prefered as
a solution for both multi-hop replication and one-hop replication
without using two different encapsulations.

4.3. L4 Inspection

The solution should not require the BFRs to inspect layer 4 or
require any changes to layer 4.

4.4. Multicast address in SA field

The solution should not allow a multicast address to be put in the IP
source address field.

4.5. Incorrect bits

The solution should not assume that bits never get set incorrectly.

If a packet with incorrect bits set, it should not damage the
functions like Unicast Reverse Path Forwarding (URPF), or cause loops
or duplicates as described in section 6.8 of [RFC8279].

4.6. SA filtering

The solution should not require changes in source address filtering
procedures.

4.7. BIER architecture support

The solution should be possible to be used to support the entire BIER
architecture.

Multiple sub-domains bound to one or many topologies or algorithms,
multiple sets for more BFERs, multiple BIFIs for ECMP should be
supported.
4.8. Keep it simple

The solution should avoid having to use different encapsulation types, or use complex tunneling techniques, to support BIER as a E2E multicast transport.

A single encapsulation should support Layer-2 switch within BFRs, or non-BFR within a BIER domain, or inter-domain deployment of BIER.

4.9. Hardware fast path

The solution should enable the processing and forwarding of BIER packets in hardware fast path.

5. Solutions Evaluation

The following are solutions that have been proposed to solve BIER in IPv6 environments.

As illustrated in these examples, the BIER header, or the BitString, may appear in the IPv6 Header, IPv6 Extension Header, IPv6 Payload, or IPv6 Tunnel Packet:

5.1. BIER-ETH encapsulation in IPv6 networks

```
+---------------+-----------------+-------------------+
|   Ethernet    |   BIER header   | payload           |
|  (ethType =   | (BIFT-id, ...)  |                   |
|    0xAB37)    |     )           |                   |
|                |  Next Header    |                   |
+---------------+-----------------+-------------------+
```

BIER-ETH encapsulation (BIER header for Non-MPLS networks as defined in [RFC8296]) can be used to transport the multicast data in the IPv6 network by encapsulating the multicast user data payload within the BIER-ETH header. However, using BIER-ETH in IPv6 networks is not considered to be a native IPv6 solution which utilizes the IPv6 header to forward the packet. Below listed are some of the properties of BIER-ETH encapsulation which could be seen as the reasons for the same:

- BIER-ETH is not agnostic to the underlying (L2) data link type. It can be deployed only in the networks with Ethernet data link and cannot be deployed in an network which deploys any other data link types. Use of BIER-ETH in IPv6 networks might also result in using different BIER encapsulations, when BIER is used as a E2E multicast transport across a larger heterogeneous IPv6 networks.
with different data link types used in different layers of the network.

- BIER-ETH in IPv6 networks is considered similar to 6PE solution where-in the multicast user data packet is encapsulated within the BIER-MPLS header.

  * It is worth noting that the only major difference between BIER-MPLS and BIER-Non-MPLS header is that BIER-MPLS uses downstream assigned MPLS label while BIER-Non-MPLS header uses a domain-wide-unique BIFT-id. While the use of domain-wide-unique BIFT-id in BIER-ETH header takes away the complexity of allocation and state maintenance from the network, it still requires some sort of ID (similar to label) to identify the application context after the decapsulation of BIER header (example: MVPN VRF Label). Encoding of such an ID/LABEL before encapsulating the multicast user data payload with BIER-ETH header cannot be avoided.

  * The absence of an IPv6 header, and the optional IPv6 extension headers, deprives BIER of some of the useful cases (ex: Use of IPv6 address for identification of network function or service mapping) that is otherwise possible in native IPv6 encapsulation which utilizes a IPv6 header.

  * Tunneling of BIER packets is one common technique used for FRR, to tunnel over BIER incapable nodes etc. While it is possible for the BIER-ETH encapsulated packet to be further encapsulated within a GRE6 or SRv6, etc tunnel, it might not be possible to parse and decapsulate different types of tunnel headers and forward the BIER packet completely in hardware fast path similar to the label stack processing in BIER-MPLS networks. It would be useful to select an encapsulation which could help in processing the tunnel and BIER header and make the forwarding decision completely in hardware fast path, which is lacking in BIER-ETH encapsulation if chosen to be deployed in pure IPv6 networks.

5.2. Encode Bitstring in IPv6 destination address

```
+-------------------+-------------------
|  IPv6 header      | payload           |
| (BitString in     |                  |
| DA lower bits)    |                  |
| Next Header       |                  |
+-------------------+-------------------
```
As described in [I-D.pfister-bier-over-ipv6], the information required by BIER is stored in the destination IPv6 address. The BIER BitString is encoded in the low-order bits of the IPv6 destination address of each packet. The high-order bits of the IPv6 destination address are used by intermediate routers for unicast forwarding, deciding whether a packet is a BIER packet, and if so, to identify the BIER Sub-Domain, Set Identifier and BitString length. No additional extension or encapsulation header is required. Instead of encapsulating the packet in IPv6, the payload is attached to the BIER IPv6 header and the IPv6 protocol number is set to the type of the payload. If the payload is UDP, the UDP checksum needs to change when the BitString in the IPv6 destination address changes.

5.3. Add BIER header into IPv6 Extension Header

<table>
<thead>
<tr>
<th>IPv6 header</th>
<th>IPv6 Ext header</th>
<th>payload</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(BIER header in TLV Type = X)</td>
<td></td>
</tr>
</tbody>
</table>

According to [RFC8200] In IPv6, optional internet-layer information is encoded in separate headers that may be placed between the IPv6 header and the upper-layer header in a packet. There is a small number of such extension headers, each one identified by a distinct Next Header value. An IPv6 packet may carry zero, one, or more extension headers, each identified by the Next Header field of the preceding header. Extension headers (except for the Hop-by-Hop Options header) are not processed, inserted, or deleted by any node along a packet’s delivery path, until the packet reaches the node (or each of the set of nodes, in the case of multicast) identified in the Destination Address field of the IPv6 header. The Hop-by-Hop Options header is not inserted or deleted, but may be examined or processed by any node along a packet’s delivery path, until the packet reaches the node (or each of the set of nodes, in the case of multicast) identified in the Destination Address field of the IPv6 header. The Hop-by-Hop Options header, when present, must immediately follow the IPv6 header. Its presence is indicated by the value zero in the Next Header field of the IPv6 header.

Two of the currently-defined extension headers are the Hop-by-Hop Options header and the Destination Options header which carry a variable number of type-length-value (TLV) encoded "options".

In [I-D.xie-bier-ipv6-encapsulation] an IPv6 BIER Destination Option is carried by the IPv6 Destination Option Header (indicated by a Next Header value 60). It is initialized in a packet sent by an IPv6 BFIR
router to inform the following BFR routers in an IPv6 BIER domain to replicate to destination BFER routers hop-by-hop. BIER is generally a hop-by-hop and one-to-many architecture and it is required for a BIER IPv6 encapsulation to include the BIER Header ([RFC8296]) as an IPv6 Extension Header, to pilot the hop-by-hop BIER replication.

Hop by hop Options Headers may be considered. The Hop-by-Hop Options header is used to carry optional information that may be examined and processed by every node along a packet’s delivery path. The Hop-by-Hop Options header is identified by a Next Header value of 0 in the IPv6 header.

Defining New Extension Headers and Options may also be considered, if the IPv6 Destination Option Header is not good enough and new extension headers can solve the problem better.

Such proposals may include requests to IANA to allocate a "BIER Option" code from "Destination Options and Hop-by-Hop Options", and/or a "BIER Option Header" code from "IPv6 Extension Header Types".

5.4. Transport BIER as IPv6 payload

+---------------------------------------------------------------+
| IPv6 header | IPv6 Ext header (optional) | BIER Hdr + payload as IPv6 payload |
| Next Header | Next Header = X             | +----------------------------------|
+---------------------------------------------------------------+

There is a proposal for a transport-independent BIER encapsulation header which is applicable regardless of the underlying transport technology. As described in [I-D.xu-bier-encapsulation] and [I-D.zhang-bier-bierin6], the BIER header, and the payload following it, can be combined as an IPv6 payload, and be indicated by a new Upper-layer IPv6 Next-Header value. A unicast IPv6 destination address is used for the replication and changes when replicating a packet out to a neighbor.

Such proposals may include a request to IANA to allocate an IPv6 Next-Header code from "Assigned Internet Protocol Numbers".

5.5. Tunneling BIER in a IPv6 tunnel
A generic IPv6 Tunnel could be used to encapsulate the bier packet within an IPv6 domain.

GRE is a mechanism by which any ethernet payload can be carried by an IP GRE tunnel due to the 16-bits ‘Protocol Type’ field. Both IPv4 and IPv6 can be used to carry GRE. The Ethernet type codepoint 0xAB37, defined for BIER, can be used in a GRE header to indicate the subsequent BIER header and payload in an IPv6 network.

UDP-based tunneling is another mechanism which uses a specific UDP port to indicate a UDP payload format. Both IPv4 and IPv6 can support UDP. Such UDP-based tunnels can be used for BIER in an IPv6 network by defining a new UDP port to indicate the BIER header and payload.

6. IANA Considerations

Some BIERv6 encapsulation proposals do not require any action from IANA while other proposals require new BIER Destination Option codepoints from IPv6 sub-registries, new "Next header" values, or require new IP Protocol codes. This document, however, does not require anything from IANA.

7. Security Considerations

There are no security issues introduced by this draft.

8. Acknowledgement

Thank you to Eric Rosen for his listed set of requirements on the bier wg list.
9. Normative References

[I-D.pfister-bier-over-ipv6]

[I-D.xie-bier-ipv6-encapsulation]

[I-D.xu-bier-encapsulation]

[I-D.zhang-bier-bierin6]


Authors’ Addresses

Mike McBride
Futurewei
Email: michael.mcbride@futurewei.com

Jingrong Xie
Huawei
Email: xiejingrong@huawei.com

Senthil Dhanaraj
Huawei
Email: senthil.dhanaraj@huawei.com

Rajiv Asati
Cisco
Email: rajiva@cisco.com
Abstract

This document specifies the ingress part of a multicast flow overlay for BIER networks. Using existing multicast listener discovery protocols, it enables multicast membership information sharing from egress routers, acting as listeners, toward ingress routers, acting as queriers. Ingress routers keep per-egress-router state, used to construct the BIER bit mask associated with IP multicast packets entering the BIER domain.

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 January 9, 2020.

Copyright Notice

Copyright (c) 2019 IETF Trust and the persons identified as the document authors. All rights reserved.
1. Introduction

The Bit Index Explicit Replication (BIER - [RFC8279]) forwarding technique enables IP multicast transport across a BIER domain. When receiving or originating a packet, ingress routers have to construct a bit mask indicating which BIER egress routers located within the same BIER domain will receive the packet. A stateless approach would consist of forwarding all incoming packets toward all egress routers, which would in turn make a forwarding decision based on local information. But any more efficient approach would require ingress routers to keep some state about egress routers multicast membership.
information, hence requiring state sharing from egress routers toward ingress routers.

This document specifies how to use the Multicast Listener Discovery protocol version 2 [RFC3810] (resp. the Internet Group Management protocol version 3 [RFC3376]) as the ingress part of a BIER multicast flow overlay (BIER layering is described in [RFC8279]) for IPv6 (resp. IPv4). It enables multicast membership information sharing from egress routers, acting as listeners, toward ingress routers, acting as queriers. Ingress routers keep per-egress-router state, used to construct the BIER bit mask associated with IP multicast packets entering the BIER domain.

This specification is applicable to both IP version 4 and version 6. It therefore specifies two separate mechanisms operating independently. For the sake of simplicity, the rest of this document uses IPv6 terminology. It can be applied to IPv4 by replacing 'MLDv2' with 'IGMPv3', and following specific requirements when explicitly stated.

2. Terminology

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.

The terms "Bit-Forwarding Router" (BFR), "Bit-Forwarding Egress Router" (BFER), "Bit-Forwarding Ingress Router" (BFIR), "BFR-id" and "BFR-Prefix" are to be interpreted as described in [RFC8279].

Additionally, the following definitions are used:

BIER Multicast Listener Discovery (BMLD): The modified version of MLD specified in this document.

BMLD Querier: A BFR implementing the Querier part of this specification. A BMLD Node MAY be both a Querier and a Listener.

BMLD Listener: A BFR implementing the Listener part of this specification. A BMLD Node MAY be both a Querier and a Listener.

3. Overview

This document proposes to use the mechanisms described in MLDv2 in order to enable multicast membership information sharing from BFERs toward BFIRs within a given BIER domain. BMLD queries (resp.
reports) are sent over BIER toward all BMLD Nodes (resp. BMLD Queriers) using modified MLDv2 messages which IP destination is set to a configured ‘all BMLD Nodes’ (resp. ‘all BMLD Queriers’) IP multicast address.

By running MLDv2 instances with per-listener explicit tracking, BMLD Queriers are able to map BMLD Listeners with MLDv2 membership states. This state is then used to construct the set of BFERs associated with each incoming IP multicast data packet.

4. Applicability Statement

BMLD runs on top of a BIER Layer and provides the ingress part of a BIER multicast flow overlay, i.e., it specifies how BFIRs construct the set of BFERs for each ingress IP multicast data packet. The BFER part of the Multicast Flow Overlay is out of scope of this document.

The BIER Layer MUST be able to transport BMLD messages toward all BMLD Queriers and Listeners. Such packets are IP multicast packets with a BFR-Prefix as source address, a multicast destination address, and containing a MLDv2 message.

BMLD only requires state to be kept by Queriers, and is therefore more scalable than PIMv2 [RFC7761] in terms of overall state, but is also likely to be less scalable than PIMv2 in terms of the amount of control traffic and the size of the state that is kept by individual routers.

This specification is applicable to both IP version 4 and version 6. It therefore specifies two separate mechanisms operating independently. For the sake of simplicity, this document uses IPv6 terminology. It can be applied to IPv4 by replacing ‘MLDv2’ with ‘IGMPv3’, and following specific requirements when explicitly stated.

5. Querier and Listener Specifications

Routers desiring to receive IP multicast traffic (e.g., for their own use, or for forwarding) MUST behave as BMLD Listeners. Routers receiving IP multicast traffic from outside the BIER domain, or originating multicast traffic, MUST behave as BMLD Queriers.

BMLD Queriers (resp. BMLD Listeners) MUST act as MLDv2 Queriers (resp. MLDv2 Listeners) as specified in [RFC3810] unless stated otherwise in this section.
5.1. Configuration Parameters

Both Queriers and Listeners MUST operate as BFIRs and BFERs within the BIER domain in order to send and receive BMLD messages. They MUST therefore be configured accordingly, as specified in [RFC8279].

All Listeners MUST be configured with an ‘all BMLD Queriers’ multicast address and the BFR-ids of all the BMLD Queriers. This is used by Listeners to send BMLD reports over BIER toward all Queriers. All Queriers MUST be configured to accept BMLD reports sent to this address.

All Queriers MUST be configured with an ‘all BMLD Nodes’ multicast address and the BFR-ids of all the Queriers and Listeners. This information is used by Queriers to send BMLD queries over BIER toward all BMLD Nodes. All BMLD Nodes MUST be configured to accept BMLD queries sent to this address.

It may be cumbersome to configure the exact set of BFR-ids for Queriers and Listeners. One MAY configure the set of BFR-ids to contain any potentially used BFR-id, perhaps having all bit positions set. There is no harm in configuring unused BFR-ids. Configuring the BFR-ids of additional routers would in most cases cause no harm, as a router would drop the BMLD message unless it is configured as a Querier or a Listener.

Note that BMLD (unlike MLDv2) makes use of per-instance configured multicast group addresses rather than well-known addresses so that multiple instances of BMLD (using different group addresses) can be run simultaneously within the same BIER domain. Configured group addresses MAY be obtained from allocated IP prefixes using [RFC3306]. One MAY choose to use the well-known MLDv2 addresses in one instance, but different instances MUST use different addresses.

IP packets coming from outside of the BIER domain and having a destination address set to the configured ‘all BMLD Queriers’ or the ‘all BMLD Nodes’ group address MUST be dropped. It is RECOMMENDED that these configured addresses have a limited scope, enforcing this behavior by scope-based filtering on BIER domain’s egress interfaces.

5.2. MLDv2 instances.

BMLD Queriers MUST run a MLDv2 Querier instance with per-host tracking, which means they keep track of the MLDv2 state associated with each BMLD Listener. For that purpose, Listeners are identified by their respective BFR-Prefix, used as IP source address in all BMLD reports.
BMLD Listeners MUST run a MLDv2 Listener instance expressing their interest in the multicast traffic they are supposed to receive for local use or forwarding.

BMLD Listeners and Queriers MUST NOT run the MLDv1 (IGMPv2 and IGMPv1 for IPv4) backward compatibility procedures.

5.2.1. Sending Queries

BMLD Queries are IP packets sent over BIER by BMLD Queriers:

- Toward all BMLD Nodes (i.e., providing to the BIER Layer the BFR-ids of all BMLD Nodes).
- With the IP destination address set to the ‘all BMLD Nodes’ group address.
- With the IP source address set to the BFR-Prefix of the sender.
- With a TTL value large enough such that the packet can be received by all BMLD Nodes, depending on the underlying BIER layer (whether it decrements the IP TTL or not) and the size of the network. The default value is 64.

5.2.2. Sending Reports

BMLD Reports are IP packets sent over BIER by BMLD Listeners:

- Toward all BMLD Queriers (i.e., providing to the BIER layer the BFR-ids of all BMLD Queriers).
- With the IP destination address set to the ‘all BMLD Queriers’ group address.
- With the IP source address set to the BFR-Prefix of the sender.
- With a TTL value large enough such that the packet can be received by all BMLD Queriers, depending on the underlying BIER layer (whether it decrements the IP TTL or not) and the size of the network. The default value is 64.

Since the reports may contain a large number of records, they may become larger than the maximum BIER payload that can be delivered to all the BMLD Queriers. Hence an implementation will need to either use a small default maximum size, allow configuration of a maximum size, or rely on MTU discovery. MTU discovery may be done for a sub-domain using BIER MTU Discovery [I-D.venaas-bier-mtud]) or for the set of BMLD Queriers using Path MTU Discovery [I-D.ietf-bier-path-mtu-discovery]).

5.2.3. Receiving Queries

BMLD Queriers and Listeners MUST check the destination address of all the IP packets that are received or forwarded over BIER whenever their own BIER bit is set in the packet. If the destination address is equal to the 'all BMLD Nodes' group address the packet is processed as specified in this section.

If the IPv6 (resp. IPv4) packet contains an ICMPv6 (resp. IGMP) message of type 'Multicast Listener Query' (resp. of type 'Membership Query'), it is processed by the MLDv2 (resp. IGMPv3) instance run by the BMLD Querier. It MUST be dropped otherwise.

During the MLDv2 processing, the packet MUST NOT be checked against the MLDv2 consistency conditions (i.e., the presence of the router alert option, the TTL equaling 1 and, for IPv6 only, the source address being link-local).

5.2.4. Receiving Reports

BMLD Queriers MUST check the destination address of all the IP packets that are received or forwarded over BIER whenever their own BIER bit is set. If the destination address is equal to the 'all BMLD Queriers' the packet is processed as specified in this section.

If the IPv6 (resp. IPv4) packet contains an ICMPv6 (resp. IGMP) message of type 'Multicast Listener Report Message v2' (resp. 'Version 3 Membership Report'), it is processed by the MLDv2 (resp. IGMPv3) instance run by the BMLD Querier. It MUST be dropped otherwise.

During the MLDv2 processing, the packet MUST NOT be checked against the MLDv2 consistency conditions (i.e., the presence of the router alert option, the TTL equaling 1 and, for IPv6 only, the source address being link-local).
5.3. Packet Forwarding

BMLD Queriers configure the BIER Layer using the information obtained using BMLD, which associates BMLD Listeners (identified by their BFR-Prefixes) with their respective MLDv2 membership state.

More specifically, the MLDv2 state associated with each BMLD Listener is provided to the BIER layer such that whenever a multicast packet enters the BIER domain, if that packet matches the membership information from a BMLD Listener, its BFR-id is added to the set of BFR-ids the packet should be forwarded to by the BIER-Layer.

6. Security Considerations

BMLD makes use of IP MLDv2 messages transported over BIER in order to configure the BIER Layer of BFIRs. BMLD messages MUST be secured, either by relying on physical or link-layer security, by securing the IP packets (e.g., using IPSec [RFC4301]), or by relying on security features provided by the BIER Layer.

Whenever an attacker would be able to spoof the identity of a router, it could:

- Redirect undesired traffic toward the spoofed router by subscribing to undesired multicast traffic.
- Prevent desired multicast traffic from reaching the spoofed router by unsubscribing to some desired multicast traffic.

7. IANA Considerations

This specification does not require any action from IANA.

8. Acknowledgements

Comments concerning this document are very welcome.

9. References

9.1. Normative References

[I-D.ietf-bier-path-mtu-discovery]
9.2. Informative References


Appendix A.  BIER Use Case in Data Centers

In current data center virtualization, virtual eXtensible Local Area Network (VXLAN) [RFC7348] is a kind of network virtualization overlay technology which is overlaid between NVEs and is intended for multi-tenancy data center networks, whose reference architecture is illustrated as per Figure 1.
And there are two kinds of most common methods about how to forward BUM packets in this virtualization overlay network. One is using PIM as underlay multicast routing protocol to build explicit multicast distribution tree, such as PIM-SM [RFC7761] or PIM-BIDIR [RFC5015] multicast routing protocol. Then, when BUM packets arrive at NVE, it requires NVE to have a mapping between the VXLAN Network Identifier and the IP multicast group. According to the mapping, NVE can encapsulate BUM packets in a multicast packet which group address is the mapping IP multicast group address and steer them through explicit multicast distribution tree to the destination NVEs. This method has two serious drawbacks. It need the underlay network supports complicated multicast routing protocol and maintains multicast related per-flow state in every transit nodes. What is more, how to configure the ratio of the mapping between VNI and IP multicast group is also an issue. If the ratio is 1:1, there should be 16M multicast groups in the underlay network at maximum to map to the 16M VNIs, which is really a significant challenge for the data center devices. If the ratio is n:1, it would result in inefficiency bandwidth utilization which is not optimal in data center networks.

The other method is using ingress replication to require each NVE to create a mapping between the VXLAN Network Identifier and the remote addresses of NVEs which belong to the same virtual network. When NVE receives BUM traffic from the attached tenant, NVE can encapsulate these BUM packets in unicast packets and replicate them and tunnel them to different remote NVEs respectively. Although this method can eliminate the burden of running multicast protocol in the underlay network, it has a significant disadvantage: large waste of bandwidth, especially in big-sized data center where there are many receivers.
BIER [RFC8279] is an architecture that provides optimal multicast forwarding through a "BIER domain" without requiring intermediate routers to maintain any multicast related per-flow state. BIER also does not require any explicit tree-building protocol for its operation. A multicast data packet enters a BIER domain at a "Bit-Forwarding Ingress Router" (BFIR), and leaves the BIER domain at one or more "Bit-Forwarding Egress Routers" (BFERs). The BFIR router adds a BIER header to the packet. The BIER header contains a bit-string in which each bit represents exactly one BFER to forward the packet to. The set of BFERs to which the multicast packet needs to be forwarded is expressed by setting the bits that correspond to those routers in the BIER header. Specifically, for BIER-TE, the BIER header may also contain a bit-string in which each bit indicates the link the flow passes through.

The following sub-sections try to propose how to take full advantage of overlay multicast protocol to carry virtual network information, and create a mapping between the virtual network information and the bit-string to implement BUM services in data centers.

A.1. Convention and Terminology

The terms about NVO3 are defined in [RFC7365]. The most common terminology used in this appendix is listed below.

NVE: Network Virtualization Edge, which is the entity that implements the overlay functionality. An NVE resides at the boundary between a Tenant System and the overlay network.

VXLAN: Virtual eXtensible Local Area Network

VNI: VXLAN Network Identifier

Virtual Network Context Identifier: Field in an overlay encapsulation header that identifies the specific VN the packet belongs to.

A.2. BIER in data centers

This section tries to describe how to use BIER as an optimal scheme to forward the broadcast, unknown and multicast (BUM) packets when they arrive at the ingress NVE in data centers.

The principle of using BIER to forward BUM traffic is that: firstly, it requires each ingress NVE to have a mapping between the Virtual Network Context Identifier and the bit-string in which each bit represents exactly one egress NVE to forward the packet to. And then, when receiving the BUM traffic, the BFIR/Ingress NVE maps the receiving BUM traffic to the mapping bit-string, encapsulates the
BIER header, and forwards the encapsulated BUM traffic into the BIER domain to the other BFERs/Egress NVEs indicated by the bit-string.

Furthermore, as for how each ingress NVE knows the other egress NVEs that belong to the same virtual network and creates the mapping is the main issue discussed below. Basically, BIER Multicast Listener Discovery is an overlay solution to support ingress routers to keep per-egress-router state to construct the BIER bit-string associated with IP multicast packets entering the BIER domain. The following section tries to extend BIER MLD to carry virtual network information(such as Virtual Network Context identifier), and advertise them between NVEs. When each NVE receive these information, they create the mapping between the virtual network information and the bit-string representing the other NVEs belonged to the same virtual network.

A.3. A BIER MLD solution for Virtual Network information

The BIER MLD solution allows having multiple MLD instances by having unique pairs of BMLD Nodes and BMLD Querier addresses for each instance. Assume for now that we have a unique instance per VNI and that all BMLD routers are using the same mapping between VNIs and BMLD address pairs. Also for each VNI there is a multicast group used for encapsulation of BUM traffic over BIER. This group may potentially be shared by some or all of the VNIs.

Each NVE acquires the Virtual Network information, and advertises this Virtual Network information to other NVEs through the MLD messages. For a given VNI it sends BMLD reports to the BMLD nodes address used for that VNI, for the group used for delivering BUM traffic for that VNI. This allows all NVE routers to know which other NVE routers have interest in BUM traffic for a particular VNI. If one attached virtual network is migrated, the NVE will withdraw the Virtual Network information by sending an unsolicited BMLD report. Note that NVEs also respond to periodic queries to BMLD Nodes addresses corresponding to VNIs for which they have interest.

When ingress NVE receives the Virtual Network information advertisement message, it builds a mapping between the receiving Virtual Network Context Identifier in this message and the bit-string in which each bit represents one egress NVE who sends the same Virtual Network information. Subsequently, once this ingress NVE receives some other MLD advertisements which include the same Virtual Network information from some other NVEs, it updates the bit-string in the mapping and adds the corresponding sending NVE to the updated bit-string. Once the ingress NVE removes one virtual network, it will delete the mapping corresponding to this virtual network as well as send withdraw message to other NVEs.
After finishing the above interaction of MLD messages, each ingress NVE knows where the other egress NVEs are in the same virtual network. When receiving BUM traffic from the attached virtual network, each ingress NVE knows exactly how to encapsulate this traffic and where to forward them to.

This can be used in both IPv4 network and IPv6 network. In IPv4, IGMP protocol does the similar extension for carrying Virtual Network information TLV in Version 2 membership report message.

Note that it is possible to have multiple VNIs map to the same pair of BMLD addresses. Provided VNIs that map to the same BMLD address uses different multicast groups for encapsulation, this is not a problem, because each instance is tracking interest for each multicast group separately. If multiple VNIs map to the same pair and the multicast group used is not unique, some NVEs may receive BUM traffic for which they are not interested. An NVE would drop packets for an unknown VNI, but it means wasting some bandwidth and processing. This is similar to the non-BIER case where there is not a unique multicast group for encapsulation. The improvement offered by using BMLD is by using multiple instance, hence reducing the problems caused by using the same transport group for multiple VNIs.

Authors’ Addresses

Pierre Pfister
Cisco Systems
Paris
France

Email: pierre pfister@darou.fr

IJsbrand Wijnands
Cisco Systems
De Kleetlaan 6a
Diegem 1831
Belgium

Email: ice@cisco.com
Applicability of BIER Multicast Overlay for Adaptive Streaming Services
draft-ietf-bier-multicast-http-response-01

Abstract

HTTP Level multicast, using BIER, is described as a use case in the BIER Use cases document. HTTP Level Multicast is used in today's video streaming and delivery services such as HLS, AR/VR etc., generally realized over IP Multicast as well as other use cases such as software update delivery. A realization of "HTTP Multicast" over "IP Multicast" is described for the video delivery use case. IP multicast is commonly used for IPTV services. DVB and BBF is also developing a reference architecture for IP Multicast service. A few problems with IPMC, such as waste of transmission bandwidth, increase in signaling when there are few users are described. Realization over BIER, through a BIER Multicast Overlay Layer, is described as an alternative. How BIER Multicast Overlay operation improves over IP Multicast, such as reduction in signaling, dynamic creation of multicast groups to reduce signaling and bandwidth wastage is described. We conclude with few next steps.

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 December 30, 2019.
1. Introduction

The BIER Use Cases document [I-D.ietf-bier-use-cases] describes an "HTTP Level Multicast" scenario, where HTTP Responses are carried over a BIER multicast infrastructure to multiple clients. Especially rate-adaptive HTTP solutions can benefit from the dynamic multicast group membership changes enabled by BIER. For this, the "server side NAP (Network Attachment Point), creates a list of outstanding client side NAP (Network Attachment Point) requests for the same HTTP
resource. When the response is available, the list of NAPs with outstanding client requests are converted into the BIER or BIER-TE bitstring and used to send the HTTP response.

In this draft, we describe use cases for such HTTP response multicast capability. Specifically for HTTP-based video streaming, we describe how this can be realized over IP Multicast and how the operation of the video delivery use case can be improved if realized over BIER. The realization over BIER is achieved through what is called "BIER Multicast overlay" layer, i.e., the methods by which the sending BIER router knows how to send other application packets. The requirements for BIER Multicast overlay layer is described in this document. It also describes the necessary functions that form the BIER multicast overlay and the operations that enable the desired "HTTP Level Multicast" behavior. One such operation is generating the PATH ID (represents the path between BFIR and BFER) based on named service relationship and translating it to appropriate BIER header. We describe a list of protocols needed for the realization of the individual operations.

1.1. Reference Deployment

Let us formulate the architecture of the BIER multicast overlay for the scenario outlined in [I-D.ietf-bier-use-cases]. This overlay is shown in Figure 1 below.
Figure 1: Deployment over BIER

The multicast overlay is formed by the BFIR and BFER of the BIER layer and the additional SH (Service Handler) and PCE (Path Computation Element) elements shown in the figure. When interconnecting with a non-BIER enabled IP routed peering network, a special SH, such as Border Gateway may be used.

The Service Handler and BFER can be assumed to be collocated and can be viewed as Client Network Attachment Point (CNAP). Clients sends and receives HTTP transactions through CNAP.

On the server side, the Service handling function can be part of the Server Network Attachment Point (SNAP). It includes the BFIR function and SH. SNAP is responsible for aggregating the relevant
HTTP Requests and sending one or more BIER Multicast HTTP response to multiple clients who requested the same content.

The SH function is assumed to be collocated with BFIR / BFER. The BFIR and BFER is assumed to be normal router boxes in the network. If the additional function of SH cannot be added to normal routers, then SH can be deployed as a separate function outside the routers. In such scenario an interface between SH and BFIR or BFER needs to be defined.

As part of the POINT/RIFE/FLAME EU Horizon 2020 projects, HTTP Level Multicast use case has been executed on SDN based and ICN based underlay network, as described in the [I-D.irtf-icnrg-deployment-guidelines].

"HTTP multicast" demonstrated benefits in HTTP-level streaming video delivery, when deployed on a POINT test bed with 80+ nodes. This draft [I-D.irtf-icnrg-deployment-guidelines] also describes protocol requirements to enable HTTP multicast to work on ICN underlay.

2. Conventions used in this document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

3. Use Cases

With the extensive use of "web technology", "distributed services" and availability of heterogeneous network, HTTP has effectively transitioned into the common transport or session layer for E2E and multi-hop communication across the web that is also called Service signaling. Multi-hop when using a sequence of HTTP instance such as HTTP caches. The draft "On the use of HTTP as a Substrate" [I-D.ietf-httpbis-bcp56bis], describes how HTTP is commonly used among service instances to communicate with each other, thus abstracting the lower layer details to application developers.

For example, HTTP provides a common transport to support application layer streaming (Section 3.1) for not only conventional TV broadcasting, but also emerging Virtual Reality (VR) applications like VR-based tourist guide. HTTP can also be leveraged to support wide-area large-scale software updates (Section 3.2) such as for Vehicle-to-Everything (V2X) or Internet of vehicles use case. In the following, we present how such HTTP transport capability can be extended with multicast delivery for HTTP responses in certain use cases.
3.1. HTTP-based Streaming

Referring to the BIER Use Cases [I-D.ietf-bier-use-cases], multicast is used to scale out HLS (HTTP live streaming) to a large number of receivers that use HTTP. This is used today in solutions like DOCSIS hybrid streaming [TR_IPMC_ABR]. Multicast can speed up both live and high-demand VoD streaming. Adaptive Bit Rate IPMC [TR_IPMC_ABR] describes use of IP multicast towards the CMTS or a box beside it, where the content is converted to HTTP/TCP to stream to the receivers (e.g., homes). A server hosting the HLS content is shown as "NAP Server". The gateways acting as receivers for the multicast from the server are shown as "Client-NAP" (CNAP). Each CNAP can serve multiple clients.

Dynamic Adaptive Streaming (DASH) [ISO_DASH] over HTTP is another HTTP-based streaming approach. In DASH, each media is described by a Media Presentation Description (MPD) file, through which a DASH client (e.g., a media player) is instructed how to download, interpret and play the media. The media content is encoded into fragments or chunks at different bit rates. Both the MPD and media fragments are stored at a server. The DASH client first needs to retrieve the MPD file from the server; then it can start to retrieve media fragments encoded at different bit rates from the server. DASH players may use rate adaptation, i.e., switching the retrieval from one rate chunks to another rate. Usually this rate adaptation is utilizing delay measurements, resulting in TCP like behavior in terms of backoff in case of increasing delay. DASH has been designed to reuse most of existing Internet infrastructure and protocols and can run over different underlying transports including HTTP. For example, two major media service providers Netflix and Youtube use DASH over HTTP as their streaming technology.

HTTP request and response used in media streaming services like HLS and DASH over HTTP, use HTTP responses for delivery of content, i.e., each chunk is returned as an HTTP response to the requesting client. In such scenarios, where semi-synchronous access to the same resource occurs (such as watching prominent videos over Netflix or similar platforms or live TV over HTTP), traffic grows linearly with the number of viewers since the HTTP-based server will provide an HTTP response to each individual viewer. This poses a significant burden on operators in terms of costs and on users in terms of likely degradation of quality.

The use of HTTP-based streaming of video content is not limited to traditional TV broadcasting. Consider a virtual reality use case where several users are joining a VR session at the same time, e.g., centered around a joint event. Hence, due to the temporal correlation of the VR sessions, we can assume that multiple requests

are sent for the same content at any point, particularly when viewing angles of VR clients are similar or the same. Due to availability of virtual functions and cloud technology, the actual end point from where content is delivered may change.

3.2. HTTP-based Software Updates

Various new types of devices such as vehicles and robots are being connected to Internet. They could be physically located at or moving between different places and connect to Internet via different telecom operators. Software updates for these devices become important and introduce point-to-multipoint traffic from a software server to devices. Using V2X as an example, the software server could be a part of telecom operators or maintained by car manufacturers. In either case, the software server keeps vehicle software or firmware images, which will be transmitted to many vehicles across the global Internet, based on a pull or push model. HTTP is commonly used for those software updates to provide an E2E transport between the software server and each vehicle requesting software updates. As a result, the traffic from the software server to vehicles increases linearly with the number of connected vehicles since each vehicle will establish a HTTP connection with the software server.

4. Requirements

A realization for the "HTTP multicast" use case may have the following requirements:

- MUST support multiple FQDN-based service endpoints to exist in the overlay to allow for utilizing several service endpoints for delivery and would therefore enable localization of content delivery.

- MUST send FQDN-based service requests at the network level to a suitable FQDN-based service endpoint via policy-based selection of appropriate path information.

- MUST allow for multicast delivery of HTTP response to same HTTP request URI.

- MUST provide direct path mobility, where the path between the egress and ingress Service Routers(SR) can be determined as being optimal (e.g., shortest path or direct path to a selected instance), is needed to avoid the use of anchor points and further reduce service-level latency.
5. Realization over IP Multicast

We now discuss the realization of chunk-based delivery over IP multicast delivery methods. We focus our presentation here on the video streaming use case in Section 3.1.

IPTV or Internet video distribution in CDNs, uses HTTP Level Multicast and realized over IP Multicast (IPMC). Many features of the IPTV service uses IPMC Group dependent state. Besides popular features like PIM, Mldp, in a variable bit rate encoded content source, content consumption also depends on group state.

DVB released reference architecture [DVB_REF_ARCH] for an end-to-end system to deliver linear content over IP networks in a scalable and standards-compliant manner. It focuses on delivering Adaptive Bit Rate unicast content over a IP multicast network.

A Multicast gateway is deployed in a CPE, Upstream Network Edge device or Terminal and provides multicast to unicast conversion facilities for several homes. All in-scope traffic on the access network between the Multicast Gateway (e.g. network edge device) and the Terminal or home gateway device is unicast. The individual media files are encapsulated into other protocols, so that they can be recovered as discrete files, when they exit the multicast pipe, which is terminated at Multicast Gateway. Interface "L" between Multicast server and Content playback supports fetching of all specified types of Content, Conditional request, Range request, Caching etc. BBF also started similar work in October 2016, called WT-399. This work is now coordinated with DVB. BBF focuses on developing the device management model.

Assume clients that are consuming the same content (such as a TV program) and that this content has for each block (typically segments worth 2 seconds of content) a set of outstanding requests from its clients. When IP Multicast is used in the domain, such as in aforementioned pre-existing solutions like in Cablelabs/DOCSIS [TR_IPMC_ABR], all possible blocks of the content have to be mapped to some IP multicast group, and the CNAP will need to know the mapping of block to groups. For example, a live stream may have 11 different bitrates available. In the most simple Block to IP multicast group mapping scheme, there could be 11 multicast groups, one for all the blocks of one bitrate (note that this is not necessarily done in deployments of this solution, but we consider it here for the purpose of explanation).

If the multicast domain and especially the links into the CNAP has enough bandwidth, this solution work well with IP multicast. As soon as there is at least one Client connected to a CNAP for one
particular content, the CNAP would join all 11 multicast groups for this content.

5.1. Mapping to Requirements

To realize "HTTP Level Multicast" over "IP Multicast", some additional functions needs to be supported in an intermediate (overlay) layer.

Support of mapping between FQDN based end points, Multicast Address. Creating multicast group from FQDN based end points.

Control mechanism related to time when to start sending response as the multicast group is created. It is required that the source should not send response immediately to the Multicast address. Wait for some time to build the group sufficiently and then send response.

Support of IGMP signaling between User device, NAPs and Multicast Router.

5.2. Problems

If the number of clients on a CNAP for a particular program is large, the approach will work fairly well, because the likelihood that each of the 11 bitrates of a content is necessary for at least one Client is then fairly high.

When the number of receivers is not very large, IP multicast runs into two issues. If all the bitrates for the content are sent across the same group, then many of the bitrates may not be required and would have to be received unnecessarily and dropped by the CNAP. If each bitrate was sent on a different IP multicast group, the CNAP could dynamically join/leave each multicast group based on the known receivers, but that would create an extremely high and undesirable amount of IP multicast signaling protocol activity (PIM/IGMP) that is easily overloading the network.

For efficiency reasons, the CNAP would need to dynamically join to only those bitrate steams where it does have outstanding requests, therefore achieving the best efficiency. This would mean in the worst case that a CNAP would need to send for each new block, aka.: every two second for every client one IGMP/PIM leave and one IGMP/PIM join towards the upstream router to get a block for an appropriate bitrate (or changed content) whenever bitrate or content on a client have changed. This high rate of control-plane signaling between CNAP and routers, and even between routers inside the multicast Domain is a major pain point and may easily prohibit deployment of these solutions because in many network devices, the performance of PIM/
IGMP is not scaled for continuous change in forwarding. Even worse, the limit may not simply be the CPU performance of the routers control plane, but a limitation in the number of changes in forwarding that the forwarding plane units (NPU/ASICs) can support.

6. Realization over BIER

6.1. Description of a "BIER Multicast Overlay" to support HTTP Multicast

The Service Handler (as in Figure 1) in BIER Multicast Overlay, process the FQDN in the service request. At the service level, e.g. HTTP service, the fixed relationship among consumer and providers may be abstracted using "Service Names", and the changing relationship at the Service execution endpoints can be managed at the "multicast overlay" level, handing out the exact locations where service request or response needs to be sent to BIER layer.

```
+-------------+        +-----------+       +-----------+
|             |        |           |       | PATH ID   |
| Service name|        | Multicast |       | translates|
| [producer,   |------->| Overlay   |------>| to BIER   |
|  consumer]   |        | Layer     |       | header    |
|             |        |           |       |           |
+-------------+        +-----------+       +-----------+
```

Figure 2: Service name to Path ID translation

We illustrate this using HTTP URI as service names. It should be noted, other identifiers can also be used as service name, such as an IP address. In the example illustration, other layers such as TCP, IP has been terminated at the egress point. Outside BIER domain we terminate TCP/IP session to extract the URI. The URI is processed by the "multicast overlay" layer to generate PATH IDENTIFIER , which is used as BIER header.

Path Identifier or PATH ID, is used in path-based approach, which utilizes path information provided by the source of the packet for forwarding said packet in the network. This is similar to segment routing albeit differing in the type of information provided for such source-based forwarding.

Once the BIER header is determined and added at the BFIR, the rest of the transport layers is assumed to be any underlay technology as supported by BIER. We assume TCP friendly transport, which can assure reliable delivery.
6.1.1. BIER Multicast Overlay Components

With reference to Figure 1, the following components are part of BIER Multicast Overlay Layer.

- **Service Handler (SH):** The Service handler terminates transport level protocols, such as TCP, and extracts the URI. It processes the URI in order to determine the PATH ID by contacting the PCE for a suitable path resolution, which in turn is used to send the HTTP Request.

- **Optional PCE:** Path Computation Element keeps track of all service execution end points through a registration process. SH interacts with the PCE to obtain PATH information by resolving the FQDN from the incoming URI at the ingress SH to a suitable PATH ID.

- **Interface functions to BFIR where the PATH ID is mapped to BIER header.** An Interface to the BFER is likely not required because the BFER will only receive the traffic that they need and should be able to derive from the BIER payload which subset of its receivers need to get an HTTP encapsulated version of a particular reply.

6.1.2. BIER Multicast Overlay Operations

As shown in Figure 3, the "Multicast overlay function" includes a function called PCE (Path Computation Element function), which is responsible for selecting the correct multicast end point and possibly realizing path policy enforcement. The result of the selection is a BIER path identifier, which is delivered to the SH upon initial path computation request (or provided to the ingress router BFIR to be added as BIER header) (i.e., when sending a request to or response for a specific URL for the first time). The path identifier is utilized for any future request for a given URL-based request.

All service end points indicate availability to the PCE through a registration procedure, the PCE will instruct all SHs to invalidate previous path identifiers to the specific URL that might exist. This may result in an a renewed path computation request at the next service request forwarding. Through this, the newly registered service endpoint might be utilized if the policy-governed path computation selects said service instance. Otherwise, a previously resolved PATH ID for the URI determined at the ingress SH is being used instead, removing any resolution latency to an SH-local lookup of the PATH ID.
In the diagram shown above, an HTTP request is sent by an IP-based device towards the FQDN of the server defined in the HTTP request.

At the client facing SH, the HTTP request is terminated at the TCP level at a local HTTP proxy. The server side SH at the egress terminates any transport protocol on the outgoing (server) side. These terminating functions are assumed to be part of the client/server SH. As a consequence, the SH obtains the destination "Service Name" from the received HTTP request.

If no local BIER forwarding information exists at the client side SH, the path computation entity (PCE) is consulted, which calculates a unicast path from the BFIR to which the client SH is connected to the BFER to which the server SH is connected. The PCE provides the forwarding information (Path ID) to the client SH, which in turn caches the result. The Client SH may forward the Path ID to BFIR, which creates the BIER header.

Ultimately, the "HTTP Request" encapsulated by BIER header, as shown in above diagram, is forwarded by the client SH towards the server-facing SH via the local BFIR. We assume a (TCP-friendly) transport protocol being used for the transmission between client and server SH. The possibility of sending one HTTP response to several CNAPs makes this a reliable multicast transport protocol. The exact nature
of this transport protocol is left for further studies. A suitable transport or Layer 2 encapsulation, as supported by BIER layer, is added to the above payload.

```
+-------------+-------------+--------------+
|             |             |              |
| Transport L2| BIER HEADER | HTTP REQUEST |
|   HEADER    |             | [ENCODED IN  |
|             |             | TEXT]        |
+-------------+-------------+--------------+
```

Figure 5: Transport Encapsulation of BIER payload

Upon arrival of an HTTP request at the server SH, it forwards the HTTP request as a well-formed HTTP request locally to the server, awaiting an HTTP response for the reverse direction.

If no BIER forwarding information exists for the reverse direction towards the requesting client SH, this information is requested from the PCE, similar to the operation in forward direction.

6.2. Achieving Multicast Responses

Upon arrival of any further client SH request at the server SH to an HTTP request whose response is still outstanding, the client SR is added to an internal request table. Optionally, the request is suppressed from being sent to the server.

Upon arrival of an HTTP response at the server SH, the server SH consults its internal request table for any outstanding HTTP requests to the same request. The server SH retrieves the stored BIER forwarding information for the reverse direction for all outstanding HTTP requests and determines the path information to all client SHs through a binary OR over all BIER forwarding identifiers with the same SI field. This newly formed joint BIER multicast response identifier is used to send the HTTP response across the network.

BIER makes the solution scalable. Instead of IP multicast with IGMP/PIM, BIER is being used between Server NAP (SNAP) and CNAP, the SNAP simply coalesces the forwarded HTTP requests from the CNAP, and determines for every requested block the set of CNAPs requesting it. A set of CNAPs corresponds to a set of bits in the BIER-bitstring, one bit per CNAP. The SNAP then sends the block into BIER with the appropriate bitstring set.
This completely eliminates any dynamic multicast signaling between CNAP and SNAP. It also avoids sending of any unnecessary data block, which in the IP multicast solution is pretty much unavoidable.

Furthermore, using the approach with BIER, the SNAP can also easily control how long to delay sending of blocks. For example, it may wait for some percentage of the time of a block (e.g., $50\% = 1$ second), therefore ensuring that it is coalescing as many requests into one BIER multicast answer as possible.

6.3. BIER multicast Overlay Traffic Management

BIER-TE (BIER Traffic Engineering [I-D.ietf-bier-te-arch]) forwards and replicates packets like BIER based on a BitString in the packet header. Where BIER forwards and replicates its packets on shortest paths towards BFER, BIER-TE allows (and requires) to also use bits in the bitstring to indicate the paths in the BIER domain across which the BIER-TE packets are to be sent. This is done to support Traffic Engineering for BIER packets via explicit hop-by-hop and/or loose hop forwarding of BIER-TE packets. A BIER-TE controller calculates explicit paths for this packet forwarding.

The Multicast Flow Overlay operates as in BIER. Instead of interacting with the BIER layer, it interacts with the BIER-TE Controller.

In this draft, "Name-based" service forwarding over BIER, is described to handle changes in service execution end points and manage adhoc relationship in a multicast group. BIER-TE is another way of doing this, while integrated with BIER architecture. The PCE function described earlier in the BIER Multicast Overlay, may become part of BIER-TE Controller. The SH function in the CNAP and SNAP communicates with BIER TE controller. SH sends the service name to the controller, which process the request using the PCE function and returns the "bitstring" to be used as BIER header for delivery of the HTTP response to multiple clients.

7. IANA Considerations

This document requests no IANA actions.

8. Security Considerations

The operations in Section 6 consider the forwarding of HTTP packets between ingress and egress points based on information derived from the HTTP request. The support for HTTPS is foreseen to ensure suitable encryption capability of such exchanges. For this to happen, we expect certificate sharing agreements to exist between the
content provider and the BIER overlay provider, ensuring the 
extraction of the suitable request parameters while allowing for the 
re-encryption of the content for an encrypted delivery over the BIER 
network. Since we liken the relationship between content and BIER 
overlay provider to that between content and CDN provider, the 
existence of certificate sharing agreements is similar to the 
practice used for CDNs.

9. Informative References

[DVB_REF_ARCH]
DVB, "Adaptive media streaming over IP multicast", DVB 
Document A176, March 2018, 
<https://www.dvb.org/resources/public/standards/a176_adaptive_media_streaming_over_ip_multicast_2018-02-16_draft_blu
ebook.pdf>.

[I-D.ietf-bier-te-arch]
Eckert, T., Cauchie, G., Braun, W., and M. Menth, "Traffic 
Engineering for Bit Index Explicit Replication (BIER-TE)", 
draft-ietf-bier-te-arch-00 (work in progress), January 
2018.

[I-D.ietf-bier-use-cases]
Kumar, N., Asati, R., Chen, M., Xu, X., Dolganow, A., 
Przygienda, T., Gulko, A., Robinson, D., Arya, V., and C. 
Bestler, "BIER Use Cases", draft-ietf-bier-use-cases-09 
(work in progress), January 2019.

[I-D.ietf-httpbis-bcp56bis]
Nottingham, M., "On the use of HTTP as a Substrate", 
draft-ietf-httpbis-bcp56bis-05 (work in progress), May 
2018.

[I-D.irtf-icnrg-deployment-guidelines]
Rahman, A., Trossen, D., Kutscher, D., and R. Ravindran, 
"Deployment Considerations for Information-Centric 
Networking (ICN)"", draft-irtf-icnrg-deployment-
guidelines-06 (work in progress), May 2019.

[ISO_DASH]
ISO, "Information technology -- Dynamic adaptive streaming 
over HTTP (DASH) -- Part 1: Media presentation description 
and segment formats", ISO/IEC 23009-1:2014, May 2014, 
<http://standards.iso.org/ittf/PubliclyAvailableStandards/
index.html>.
Authors’ Addresses

Dirk Trossen
InterDigital Europe, Ltd
64 Great Eastern Street, 1st Floor
London EC2A 3QR
United Kingdom

Email: Dirk.Trossen@InterDigital.com

Akbar Rahman
InterDigital Communications, LLC
1000 Sherbrooke Street West
Montreal H3A 3G4
Canada

Email: Akbar.Rahman@InterDigital.com

Chonggang Wang
InterDigital Communications, LLC
1001 E Hector St
Conshohocken 19428
USA

Email: Chonggang.Wang@InterDigital.com

Toerless Eckert
Futurewei Technologies Inc.
2330 Central Expy
Santa Clara 95050
USA

Email: tte+ietf@cs.fau.de
Performance Measurement (PM) with Marking Method in Bit Index Explicit Replication (BIER) Layer
draft-ietf-bier-pmmm-oam-06

Abstract

This document describes a hybrid performance measurement method for multicast service through a Bit Index Explicit Replication domain.

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 January 2, 2020.

Copyright Notice

Copyright (c) 2019 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
1. Introduction

[RFC8279] introduces and explains the Bit Index Explicit Replication (BIER) architecture and how it supports the forwarding of multicast data packets. [RFC8296] specified that in the case of BIER encapsulation in an MPLS network, a BIER-MPLS label, the label that is at the bottom of the label stack, uniquely identifies the multicast flow. [RFC8321] describes a hybrid performance measurement method, per RFC7799’s classification of measurement methods [RFC7799]. The method, called Packet Network Performance Monitoring (PNPM), can be used to measure packet loss, latency, and jitter on live traffic complies with requirements #5 and #12 listed in [I-D.ietf-bier-oam-requirements]. Because this method is based on marking consecutive batches of packets, the method is often referred to as a marking method.

This document defines how the marking method can be used on the BIER layer to measure packet loss and delay metrics of a multicast flow in an MPLS network.

2. Conventions used in this document
2.1. Terminology

BFR: Bit-Forwarding Router
BFER: Bit-Forwarding Egress Router
BFIR: Bit-Forwarding Ingress Router
BIER: Bit Index Explicit Replication
OAM: Operations, Administration and Maintenance

2.2. 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.

3. OAM Field in BIER Header

[RFC8296] defined the two-bits long field, referred to as OAM. The OAM field can be used for the marking performance measurement method. Because the setting of the field to any value does not affect forwarding and/or quality of service treatment of a packet, using the OAM field for PNPM in BIER layer can be viewed as the example of the hybrid performance measurement method.

Figure 1 displays the interpretation of the OAM field defined in this specification for the use by PNPM method.

```
 0
 0 1
+-+-+-+-+
| S | D |
+-+-+-+-+
```

Figure 1: OAM field of BIER Header format

where:

- S - Single-Marking flag;
- D - Double-Marking flag.
4. Theory of Operation

The marking method can be used in the multicast environment supported by BIER layer. Without limiting any generality consider multicast network presented in Figure 2. Any combination of markings can be applied to a multicast flow by the Bit Forwarding Ingress Router (BFIR) at either ingress or egress point to perform node, link, segment or end-to-end measurement to detect performance degradation defect and localize it efficiently.

```
-----
  \   
   --| D |
       
-----
  \   
   --| B |--
       
-----
  \   
   --|-- E |
       
-----
  \   
   | A |
-----
  \   
   --|-- F |
       
-----
  \   
   --| C |--
       
-----
  \   
   --|-- G |
```

Figure 2: Multicast network

Using the marking method, a BFIR creates distinct sub-flows in the particular multicast traffic over BIER layer. Each sub-flow consists of consecutive blocks of identically marked packets. For example, a block of N packets, with each packet being marked as X, is followed by the block of M packets with each packet being marked as Y. These blocks are unambiguously recognizable by a monitoring point at any Bit Forwarding Router (BFR) and can be measured to calculate packet loss and/or packet delay metrics. It is expected that the marking values be set and cleared at the edge of BIER domain. Thus for the scenario presented in Figure 2 if the operator initially monitors the A-C-G and A-B-D segments he may enable measurements on segments C-F and B-E at any time.


As explained in [RFC8321], marking can be applied to delineate blocks of packets based either on the equal number of packets in a block or based on the equal time interval. The latter method offers better control as it allows a better account for capabilities of downstream
nodes to report statistics related to batches of packets and, at the same time, time resolution that affects defect detection interval.

If the Single-Marking measurement is used to measure packet loss, then the D flag MUST be set to zero on transmit and ignored by the monitoring point.

The S flag is used to create sub-flows to measure the packet loss by switching the value of the S flag every N-th packet or at certain time intervals. Delay metrics MAY be calculated with the sub-flow using any of the following methods:

- First/Last Packet Delay calculation: whenever the marking, i.e., the value of S flag changes, a BFR can store the timestamp of the first/last packet of the block. The timestamp can be compared with the timestamp of the packet that arrived in the same order through a monitoring point at a downstream BFR to compute packet delay. Because timestamps collected based on the order of arrival this method is sensitive to packet loss and re-ordering of packets (see Section 4.3 for more details).

- Average Packet Delay calculation: an average delay is calculated by considering the average arrival time of the packets within a single block. A BFR may collect timestamps for each packet received within a single block. Average of the timestamp is the sum of all the timestamps divided by the total number of packets received. Then the difference between the average packet arrival time calculated for the downstream monitoring point and the same metric but calculated at the upstream monitoring point is the average packet delay on the segment between these two points. This method is robust to out of order packets and also to packet loss on the segment between the measurement points (packet loss may cause a minor loss of accuracy in the calculated metric because the number of packets used is different at each measurement point). This method only provides a single metric for the duration of the block, and it doesn’t give the minimum and maximum delay values. This limitation of producing only the single metric could be overcome by reducing the duration of the block. As a result, the calculated value of the average delay will better reflect the minimum and maximum delay values of the block’s duration time.

4.2. Double-Marking Enabled Measurement

Double-Marking method allows measurement of minimum and maximum delays for the monitored flow, but it requires more nodal and network resources. If the Double-Marking method used, then the S flag is used to create the sub-flow, i.e., mark blocks of packets. The D
flag is used to mark single packets within a block to measure delay and jitter.

The first marking (S flag alternation) is needed for packet loss and also for average delay measurement. The second marking (D flag is put to one) creates a new set of marked packets that are fully identified over the BIER network, so that a BFR can store the timestamps of these packets; these timestamps can be compared with the timestamps of the same packets on a second BFR to compute packet delay values for each packet. The number of measurements can be easily increased by changing the frequency of the second marking. On the other hand, the higher frequency of the second marking will cause a higher volume of the measurement data being transported through the BIER domain. An operator should consider and balance both effects. This method is useful to measure not only the average delay but also the minimum and maximum delay values and, in wider terms, to know more about the statistic distribution of delay values.

4.3. Operational Considerations

For the ease of operational procedures, the initial marking of a multicast flow is performed at BFIR and cleared, by way of removing BIER encapsulation form a payload packet, at the edge of the BIER domain by BFERs.

Since at the time of writing this specification, there are no proposals to using auto-discovery or signaling mechanism to inform downstream nodes what methodology is used each monitoring point MUST be configured beforehand.

Section 4.3 [RFC8321] provides a detailed analysis of how packet re-ordering and the duration of the block in the Single-Marking mode of the marking method impact the accuracy of the packet loss measurement. Re-ordering of packets in the Single-Marking mode will be noticeable only at the edge of a block of packets (re-ordering within the block cannot be detected in the Single-Marking mode). If the extra delay for some packets is much smaller than half of the duration of a block, then it should be easier to attribute re-ordered packets to the proper block and thus maintain the accuracy of the packet loss measurement.

5. IANA Considerations

This document requests IANA to register format of the OAM field of BIER Header as the following:
6. Security Considerations

Regarding using the marking method, [RFC8321] stressed two types of security concerns. First, the potential harm caused by the measurements, is a lesser threat as [RFC8296] defines OAM field used by the marking method so that the value of "two bits have no effect on the path taken by a BIER packet and have no effect on the quality of service applied to a BIER packet." Second security concern, potential harm to the measurements can be mitigated by using policy, suggested in [RFC8296], to accept BIER packets only from trusted routers, not from customer-facing interfaces.

All the security considerations for BIER discussed in [RFC8296] are inherited by this document.

7. Acknowledgement

TBD

8. References

8.1. Normative References


8.2. Informative References

[I-D.ietf-bier-oam-requirements]


Authors' Addresses

Greg Mirsky
ZTE Corp.
Email: gregimirsky@gmail.com

Lianshu Zheng
Huawei Technologies
Email: vero.zheng@huawei.com

Mach Chen
Huawei Technologies
Email: mach.chen@huawei.com
Traffic Engineering for Bit Index Explicit Replication (BIER-TE)  
draft-ietf-bier-te-arch-03

Abstract

This memo introduces per-packet stateless strict and loose path engineered replication and forwarding for Bit Index Explicit Replication packets ([RFC8279]). This is called BIER-TE.

BIER-TE leverages the BIER architecture ([RFC8279]) and extends it with a new semantic for bits in the bitstring. BIER-TE can leverage BIER forwarding engines with little or no changes.

In BIER, the BitPositions (BP) of the packets bitstring indicate BIER Forwarding Egress Routers (BFER), and hop-by-hop forwarding uses a Routing Underlay such as an IGP.

In BIER-TE, BitPositions indicate adjacencies. The BIFT of each BFR are only populated with BPs that are adjacent to the BFR in the BIER-TE topology. The BIER-TE topology can consist of layer 2 or remote (route) adjacencies. The BFR then replicates and forwards BIER packets to those adjacencies. This results in the aforementioned strict and loose path forwarding.

BIER-TE can co-exist with BIER forwarding in the same domain, for example by using separate sub-domains. In the absence of routed adjacencies, BIER-TE does not require a BIER routing underlay, and can then be operated without requiring an IGP routing protocol.

BIER-TE operates without explicit in-network tree-building and carries the multicast distribution tree in the packet header. It can therefore be a good fit to support multicast path steering in Segment Routing (SR) networks.

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 January 9, 2020.

Copyright Notice

Copyright (c) 2019 IETF Trust and the persons identified as the document authors. All rights reserved.

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

Table of Contents

1. Introduction .............................................. 3
   1.1. Basic Examples ...................................... 4
   1.2. BIER-TE Topology and adjacencies ...................... 7
   1.3. Comparison with BIER .................................. 8
   1.4. Requirements Language ................................ 8
2. Layering .................................................. 8
   2.1. The Multicast Flow Overlay ........................... 9
   2.2. The BIER-TE Controller Host .......................... 9
   2.2.1. Assignment of BitPositions to adjacencies of the network topology .......... 10
   2.2.2. Changes in the network topology .................... 10
   2.2.3. Set up per-multicast flow BIER-TE state .......... 10
   2.2.4. Link/Node Failures and Recovery ................. 10
   2.3. The BIER-TE Forwarding Layer ........................ 11
   2.4. The Routing Underlay ................................ 11
3. BIER-TE Forwarding ....................................... 11
   3.1. The Bit Index Forwarding Table (BIFT) ................ 11
   3.2. Adjacency Types ...................................... 12
   3.2.1. Forward Connected .................................. 12

1. Introduction

BIER-TE shares architecture, terminology and packet formats with BIER as described in [RFC8279] and [RFC8296]. This document describes BIER-TE in the expectation that the reader is familiar with these two documents.
In BIER-TE, BitPositions (BP) indicate adjacencies. The BIFT of each BFR is only populated with BP that are adjacent to the BFR in the BIER-TE Topology. Other BPs are left without adjacency. The BFR replicate and forwards BIER packets to adjacent BPs that are set in the packet. BPs are normally also reset upon forwarding to avoid duplicates and loops. This is detailed further below.

1.1. Basic Examples

BIER-TE forwarding is best introduced with simple examples.
BIER-TE Topology:

Diagram:

```
p5    p6
    --- BFR3 ---
p3/    p13     \p7
BFR1 ---- BFR2                                BFR5 ----- BFR6
p1    p2    p4\    \p14     /p10    p11    p12
    --- BFR4 ---
p8    p9
```

(simplified) BIER-TE Bit Index Forwarding Tables (BIFT):

- **BFR1**: p1 -> local_decap  
  p2 -> forward_connected to BFR2

- **BFR2**: p1 -> forward_connected to BFR1  
  p5 -> forward_connected to BFR3  
  p8 -> forward_connected to BFR4

- **BFR3**: p3 -> forward_connected to BFR2  
  p7 -> forward_connected to BFR5  
  p13 -> local_decap

- **BFR4**: p4 -> forward_connected to BFR2  
  p10 -> forward_connected to BFR5  
  p14 -> local_decap

- **BFR5**: p6 -> forward_connected to BFR3  
  p9 -> forward_connected to BFR4  
  p12 -> forward_connected to BFR6

- **BFR6**: p11 -> forward_connected to BFR5  
  p12 -> local_decap

Figure 1: BIER-TE basic example

Consider the simple network in the above BIER-TE overview example picture with 6 BFR. p1...p14 are the BitPositions (BP) used. All BFR can act as ingress BFR (BFIR), BFR1, BFR3, BFR4 and BFR6 can also be egress BFR (BFER). Forward_connected is the name for adjacencies that are representing subnet adjacencies of the network. Local_decap is the name of the adjacency to decapsulate BIER-TE packets and pass their payload to higher layer processing.
Assume a packet from BFR1 should be sent via BFR4 to BFR6. This requires a bitstring \((p2,p8,p10,p12)\). When this packet is examined by BIER-TE on BFR1, the only BitPosition from the bitstring that is also set in the BIFT is \(p2\). This will cause BFR1 to send the only copy of the packet to BFR2. Similarly, BFR2 will forward to BFR4 because of \(p8\), BFR4 to BFR5 because of \(p10\) and BFR5 to BFR6 because of \(p12\). \(p12\) also makes BFR6 receive and decapsulate the packet.

To send in addition to BFR6 via BFR4 also a copy to BFR3, the bitstring needs to be \((p2,p5,p8,p10,p12,p13)\). When this packet is examined by BFR2, \(p5\) causes one copy to be sent to BFR3 and \(p8\) one copy to BFR4. When BFR3 receives the packet, \(p13\) will cause it to receive and decapsulate the packet.

If instead the bitstring was \((p2,p6,p8,p10,p12,p13)\), the packet would be copied by BFR5 towards BFR3 because \(p6\) instead of BFR2 to BFR5 because of \(p6\) in the prior case. This is showing the ability of the shown BIER-TE Topology to make the traffic pass across any possible path and be replicated where desired.

BIER-TE has various options to minimize BP assignments, many of which are based on assumptions about the required multicast traffic paths and bandwidth consumption in the network.

The following picture shows a modified example, in which Rtr2 and Rtr5 are assumed not to support BIER-TE, so traffic has to be unicast encapsulated across them. Unicast tunneling of BIER-TE packets can leverage any feasible mechanism such as MPLS or IP, these encapsulations are out of scope of this document. To emphasize non-native forwarding of BIER-TE packets, these adjacencies are called "forward_routed", but otherwise there is no difference in their processing over the aforementioned "forward_connected" adjacencies.

In addition, bits are saved in the following example by assuming that BFR1 only needs to be BFIR but not BFER or transit BFR.
BIER-TE Topology:

Diagram:

```
p1  p3  p7
....> BFR3 <....  p5
........       ........>
BFR1       (Rtr2)          (Rtr5)      BFR6
........       ........>
....> BFR4 <....  p6
    p2  p4  p8
```

(simplified) BIER-TE Bit Index Forwarding Tables (BIFT):

- **BFR1**: p1 -> forward_routed to BFR3
  p2 -> forward_routed to BFR4
- **BFR3**: p3 -> local_decap
  p5 -> forward_routed to BFR6
- **BFR4**: p4 -> local_decap
  p6 -> forward_routed to BFR6
- **BFR6**: p5 -> local_decap
  p6 -> local_decap
  p7 -> forward_routed to BFR3
  p8 -> forward_routed to BFR4

Figure 2: BIER-TE basic overlay example

To send a BIER-TE packet from BFR1 via BFR3 to BFR6, the bitstring is (p1,p5). From BFR1 via BFR4 to BFR6 it is (p2,p6). A packet from BFR1 to BFR3,BFR4 and BFR6 can use (p1,p2,p3,p4,p5) or (p1,p2,p3,p4,p6), or via BFR6 (p2,p3,p4,p6,p7) or (p1,p3,p4,p5,p8).

1.2. BIER-TE Topology and adjacencies

The key new component in BIER-TE to control where replication can or should happens and how to minimize the required BP for segments is - as shown in these two examples - the BIER-TE topology.

The BIER-TE Topology effectively consists of the BIFT of all the the BFR and can also be expressed in a diagram as a graph where the edges are the adjacencies between the BFR. Adjacencies are naturally unidirectional. BP can be reused across multiple adjacencies as long as this does not lead to undesired duplicates or loops as explained further down in the text.
If the BIER-TE topology represents the underlying (layer 2) topology of the network, this is called "native" BIER-TE as shown in the first example. This can be freely mixed with "overlay" BIER-TE, in "forward_routed" adjacencies are used.

1.3. Comparison with BIER

The key differences over BIER are:

- BIER-TE replaces in-network autonomous path calculation by explicit paths calculated offpath by the BIER-TE controller host.

- In BIER-TE every BitPosition of the BitString of a BIER-TE packet indicates one or more adjacencies - instead of a BFER as in BIER.

- BIER-TE in each BFR has no routing table but only a BIER-TE Forwarding Table (BIFT) indexed by SI:BitPosition and populated with only those adjacencies to which the BFR should replicate packets to.

BIER-TE headers use the same format as BIER headers.

BIER-TE forwarding does not require/use the BFIR-ID. The BFIR-ID can still be useful though for coordinated BFIR/BFER functions, such as the context for upstream assigned labels for MPLS payloads in MVPN over BIER-TE.

If the BIER-TE domain is also running BIER, then the BFIR-ID in BIER-TE packets can be set to the same BFIR-ID as used with BIER packets.

If the BIER-TE domain is not running full BIER or does not want to reduce the need to allocate bits in BIER bitstrings for BFIR-ID values, then the allocation of BFIR-ID values in BIER-TE packets can be done through other mechanisms outside the scope of this document, as long as this is appropriately agreed upon between all BFIR/BFER.

1.4. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

2. Layering

End to end BIER-TE operations consists of four layers: The "Multicast Flow Overlay", the "BIER-TE Controller Host", the "Routing Underlay" and the "BIER-TE forwarding layer". The Bier-TE Controller Host is the new architectural element in BIER-TE compared to BIER.
The Multicast Flow Overlay

The Multicast Flow Overlay operates as in BIER. See [RFC8279]. Instead of interacting with the BIER forwarding layer layer (as in BIER), it interacts with the BIER-TE Controller Host.

2.2. The BIER-TE Controller Host

The BIER-TE controller host is representing the control plane of BIER-TE. It communicates two sets of information with BFRs:

During bring-up or modifications of the network topology, the controller discovers the network topology and creates the BIER-TE topology from it: determine which adjacencies are required/desired and assign BitPositions to them. Then it signals the resulting of BitPositions and their adjacencies to each BFR to set up their BIER-TE BIFTs.

During day-to-day operations of the network, the controller signals to BFIRs what multicast flows are mapped to what BitStrings.

Communications between the BIER-TE controller host to BFRs is ideally via standardized protocols and data-models such as Netconf/Restconf/Yang. This is currently outside the scope of this document. Vendor-
specific CLI on the BFRs is also a possible stopgap option (as in many other SDN solutions lacking definition of standardized data model).

For simplicity, the procedures of the BIER-TE controller host are described in this document as if it is a single, centralized automated entity, such as an SDN controller. It could equally be an operator setting up CLI on the BFRs. Distribution of the functions of the BIER-TE controller host is currently outside the scope of this document.

2.2.1. Assignment of BitPositions to adjacencies of the network topology

The BIER-TE controller host tracks the BFR topology of the BIER-TE domain. It determines what adjacencies require BitPositions so that BIER-TE explicit paths can be built through them as desired by operator policy.

The controller then pushes the BitPositions/adjacencies to the BIFT of the BFRs, populating only those SI:BitPositions to the BIFT of each BFR to which that BFR should be able to send packets to - adjacencies connecting to this BFR.

2.2.2. Changes in the network topology

If the network topology changes (not failure based) so that adjacencies that are assigned to BitPositions are no longer needed, the controller can re-use those BitPositions for new adjacencies. First, these BitPositions need to be removed from any BFIR flow state and BFR BIFT state, then they can be repopulated, first into BIFT and then into the BFIR.

2.2.3. Set up per-multicast flow BIER-TE state

The BIER-TE controller host tracks the multicast flow overlay to determine what multicast flow needs to be sent by a BFIR to which set of BFER. It calculates the desired distribution tree across the BIER-TE domain based on algorithms outside the scope of this document (e.g.: CSFP, Steiner Tree,...). It then pushes the calculated BitString into the BFIR.

2.2.4. Link/Node Failures and Recovery

When link or nodes fail or recover in the topology, BIER-TE can quickly respond with the optional FRR procedures described in [I-D.eckert-bier-te-frr]. It can also more slowly react by recalculating the BitStrings of affected multicast flows. This
reaction is slower than the FRR procedure because the controller needs to receive link/node up/down indications, recalculate the desired BitStrings and push them down into the BFIRs. With FRR, this is all performed locally on a BFR receiving the adjacency up/down notification.

2.3. The BIER-TE Forwarding Layer

When the BIER-TE Forwarding Layer receives a packet, it simply looks up the BitPositions that are set in the BitString of the packet in the Bit Index Forwarding Table (BIFT) that was populated by the BIER-TE controller host. For every BP that is set in the BitString, and that has one or more adjacencies in the BIFT, a copy is made according to the type of adjacencies for that BP in the BIFT. Before sending any copy, the BFR resets all BitPositions in the BitString of the packet to which it can create a copy. This is done to inhibit that packets can loop.

2.4. The Routing Underlay

BIER-TE is sending BIER packets to directly connected BIER-TE neighbors as L2 (unicasted) BIER packets without requiring a routing underlay. BIER-TE forwarding uses the Routing underlay for forward_routed adjacencies which copy BIER-TE packets to not-directly-connected BFRs (see below for adjacency definitions).

If the BFR intends to support FRR for BIER-TE, then the BIER-TE forwarding plane needs to receive fast adjacency up/down notifications: Link up/down or neighbor up/down, eg.: from BFD. Providing these notifications is considered to be part of the routing underlay in this document.

3. BIER-TE Forwarding

3.1. The Bit Index Forwarding Table (BIFT)

The Bit Index Forwarding Table (BIFT) exists in every BFR. For every subdomain in use, it is a table indexed by SI:BitPosition and is populated by the BIER-TE control plane. Each index can be empty or contain a list of one or more adjacencies.

BIER-TE can support multiple subdomains like BIER. Each one with a separate BIFT

In the BIER architecture, indices into the BIFT are explained to be both BFR-id and SI:BitString (BitPosition). This is because there is a 1:1 relationship between BFR-id and SI:BitString - every bit in every SI is/can be assigned to a BFIR/BFER. In BIER-TE there are
more bits used in each BitString than there are BFIR/BFER assigned to
the bitstring. This is because of the bits required to express the
(traffic engineered) path through the topology. The BIER-TE
forwarding definitions do therefore not use the term BFR-id at all.
Instead, BFR-ids are only used as required by routing underlay, flow
overlay of BIER headers. Please refer to Section 7 for explanations
how to deal with SI, subdomains and BFR-id in BIER-TE.

<table>
<thead>
<tr>
<th>Index: SI:BitPosition</th>
<th>Adjacencies: &lt;empty&gt; or one or more per entry</th>
</tr>
</thead>
<tbody>
<tr>
<td>0:1</td>
<td>forward_connected(interface,neighbor,DNR)</td>
</tr>
<tr>
<td>0:2</td>
<td>forward_connected(interface,neighbor,DNR)</td>
</tr>
<tr>
<td>0:3</td>
<td>local_decap({VRF})</td>
</tr>
<tr>
<td>0:4</td>
<td>forward_routed({VRF,}l3-neighbor)</td>
</tr>
<tr>
<td>0:5</td>
<td>&lt;empty&gt;</td>
</tr>
<tr>
<td>0:6</td>
<td>ECMP((adjacency1,...adjacencyN), seed)</td>
</tr>
<tr>
<td>...</td>
<td>BitStringLength</td>
</tr>
</tbody>
</table>

Figure 4: BIFT adjacencies

The BIFT is programmed into the data plane of BFRs by the BIER-TE
controller host and used to forward packets, according to the rules
specified in the BIER-TE Forwarding Procedures.

Adjacencies for the same BP when populated in more than one BFR by
the controller do not have to have the same adjacencies. This is up
to the controller. BPs for p2p links are one case (see below).

3.2. Adjacency Types

3.2.1. Forward Connected

A "forward_connected" adjacency is towards a directly connected BFR
neighbor using an interface address of that BFR on the connecting
interface. A forward_connected adjacency does not route packets but only L2 forwards them to the neighbor.

Packets sent to an adjacency with "DoNotReset" (DNR) set in the BIFT will not have the BitPosition for that adjacency reset when the BFR creates a copy for it. The BitPosition will still be reset for copies of the packet made towards other adjacencies. The can be used for example in ring topologies as explained below.

3.2.2. Forward Routed

A "forward_routed" adjacency is an adjacency towards a BFR that is not a forward_connected adjacency: towards a loopback address of a BFR or towards an interface address that is non-directly connected. Forward_routed packets are forwarded via the Routing Underlay.

If the Routing Underlay has multiple paths for a forward_routed adjacency, it will perform ECMP independent of BIER-TE for packets forwarded across a forward_routed adjacency.

If the Routing Underlay has FRR, it will perform FRR independent of BIER-TE for packets forwarded across a forward_routed adjacency.

3.2.3. ECMP

The ECMP mechanisms in BIER are tied to the BIER BIFT and are therefore not directly useable with BIER-TE. The following procedures describe ECMP for BIER-TE that we consider to be lightweight but also well manageable. It leverages the existing entropy parameter in the BIER header to keep packets of the flows on the same path and it introduces a "seed" parameter to allow engineering traffic to be polarized or randomized across multiple hops.

An "Equal Cost Multipath" (ECMP) adjacency has a list of two or more adjacencies included in it. It copies the BIER-TE to one of those adjacencies based on the ECMP hash calculation. The BIER-TE ECMP hash algorithm must select the same adjacency from that list for all packets with the same "entropy" value in the BIER-TE header if the same number of adjacencies and same seed are given as parameters. Further use of the seed parameter is explained below.

3.2.4. Local Decap

A "local_decap" adjacency passes a copy of the payload of the BIER-TE packet to the packets NextProto within the BFR (IPv4/IPv6, Ethernet,...). A local_decap adjacency turns the BFR into a BFER for matching packets. Local_decap adjacencies require the BFER to
3.3. Encapsulation considerations

Specifications for BIER-TE encapsulation are outside the scope of this document. This section gives explanations and guidelines.

Because a BFR needs to interpret the BitString of a BIER-TE packet differently from a BIER packet, it is necessary to distinguish BIER from BIER-TE packets. This is subject to definitions in BIER encapsulation specifications.

MPLS encapsulation [RFC8296] for example assigns one label by which BFRs recognizes BIER packets for every (SI, subdomain) combination. If it is desirable that every subdomain can forward only BIER or BIER-TE packets, then the label allocation could stay the same, and only the forwarding model (BIER/BIER-TE) would have to be defined per subdomain. If it is desirable to support both BIER and BIER-TE forwarding in the same subdomain, then additional labels would need to be assigned for BIER-TE forwarding.

"forward_routed" requires an encapsulation permitting to unicast BIER-TE packets to a specific interface address on a target BFR. With MPLS encapsulation, this can simply be done via a label stack with that addresses label as the top label - followed by the label assigned to (SI, subdomain) - and if necessary (see above) BIER-TE. With non-MPLS encapsulation, some form of IP tunneling (IP in IP, LISP, GRE) would be required.

The encapsulation used for "forward_routed" adjacencies can equally support existing advanced adjacency information such as "loose source routes" via eg: MPLS label stacks or appropriate header extensions (eg: for IPv6).

3.4. Basic BIER-TE Forwarding Example

Step by step example of basic BIER-TE forwarding. This does not use ECMP or forward_routed adjacencies nor does it try to minimize the number of required BitPositions for the topology.
Figure 5: BIER-TE Forwarding Example

pXX indicate the BitPositions number assigned by the BIER-TE controller host to adjacencies in the BIER-TE topology. For example, p9 is the adjacency towards BFR9 on the LAN connecting to BFER2.

**BIFT BFIR2:**
- p13: local_decap()
- p2: forward_connected(BFR3)

**BIFT BFR3:**
- p1: forward_connected(BFIR2)
- p7: forward_connected(BFER1)
- p8: forward_connected(BFR4)

**BIFT BFER1:**
- p11: local_decap()
- p6: forward_connected(BFR3)
- p8: forward_connected(BFR4)

Figure 6: BIER-TE Forwarding Example Adjacencies

...and so on.

Traffic needs to flow from BFIR2 towards Rcv1, Rcv2. The controller determines it wants it to pass across the following paths:

- BFIR2 -> BFR3
  -> BFR4 -> BFR5 -> BFER2 -> Rcv2

Figure 7: BIER-TE Forwarding Example Paths

These paths equal to the following BitString: p2, p5, p7, p8, p10, p11, p12.

This BitString is set up in BFIR2. Multicast packets arriving at BFIR2 from Src are assigned this BitString.

BFIR2 forwards based on that BitString. It has p2 and p13 populated. Only p13 is in BitString which has an adjacency towards BFR3. BFIR2 resets p2 in BitString and sends a copy towards BFR2.

BFR3 sees a BitString of p5, p7, p8, p10, p11, p12. It is only interested in p1, p7, p8. It creates a copy of the packet to BFER1 (due to p7) and one to BFR4 (due to p8). It resets p7, p8 before sending.

BFER1 sees a BitString of p5, p10, p11, p12. It is only interested in p6, p7, p8, p11 and therefore considers only p11. p11 is a "local_decap" adjacency installed by the BIER-TE controller host because BFER1 should pass packets to IP multicast. The local_decap adjacency instructs BFER1 to create a copy, decapsulate it from the BIER header and pass it on to the NextProtocol, in this example IP multicast. IP multicast will then forward the packet out to LAN2 because it did receive PIM or IGMP joins on LAN2 for the traffic.

Further processing of the packet in BFR4, BFR5 and BFER2 accordingly.

3.5. Forwarding comparison with BIER

Forwarding of BIER-TE is designed to allow common forwarding hardware with BIER. In fact, one of the main goals of this document is to encourage the building of forwarding hardware that can not only support BIER, but also BIER-TE - to allow experimentation with BIER-TE and support building of BIER-TE control plane code.

The pseudocode in Section 6 shows how existing BIER/BIFT forwarding can be amended to support basic BIER-TE forwarding, by using BIER BIFT’s F-BM. Only the masking of bits due to avoid duplicates must be skipped when forwarding is for BIER-TE.
Whether to use BIER or BIER-TE forwarding can simply be a configured choice per subdomain and accordingly be set up by a BIER-TE controller host. The BIER packet encapsulation [RFC8296] too can be reused without changes except that the currently defined BIER-TE ECMP adjacency does not leverage the entropy field so that field would be unused when BIER-TE forwarding is used.

3.6. Requirements

Basic BIER-TE forwarding MUST support to configure Subdomains to use basic BIER-TE forwarding rules (instead of BIER). With basic BIER-TE forwarding, every bit MUST support to have zero or one adjacency. It MUST support the adjacency types forward_connected without DNR flag, forward_routed and local_decap. All other BIER-TE forwarding features are optional. This Basic BIER-TE requirements make BIER-TE forwarding exactly the same as BIER forwarding with the exception of skipping the aforementioned F-BM masking on egres.

BIER-TE forwarding SHOULD support the DNR flag, as this is highly useful to save bits in rings (see Section 4.6).

BIER-TE forwarding MAY support more than one adjacency on a bit and ECMP adjacencies. The importance of ECMP adjacencies is unclear when traffic engineering is used because it may be more desirable to explicitly steer traffic across non-ECMP paths to make per-path traffic calculation easier for controllers. Having more than one adjacency for a bit allows further savings of bits in hub&spoke scenarios, but unlike rings it is less "natural" to flood traffic across multiple links unconditional. Both ECMP and multiple adjacencies are forwarding plane features that should be possible to support later when needed as they do not impact the basic BIER-TE replication loop. This is true because there is no inter-copy dependency through resetting of F-BM as in BIER.

4. BIER-TE Controller Host BitPosition Assignments

This section describes how the BIER-TE controller host can use the different BIER-TE adjacency types to define the BitPositions of a BIER-TE domain.

Because the size of the BitString is limiting the size of the BIER-TE domain, many of the options described exist to support larger topologies with fewer BitPositions (4.1, 4.3, 4.4, 4.5, 4.6, 4.7, 4.8).
4.1. P2P Links

Each P2p link in the BIER-TE domain is assigned one unique BitPosition with a forward_connected adjacency pointing to the neighbor on the p2p link.

4.2. BFER

Every BFER is given a unique BitPosition with a local_decap adjacency.

4.3. Leaf BFERs

Leaf BFERs are BFERs where incoming BIER-TE packets never need to be forwarded to another BFR but are only sent to the BFER to exit the BIER-TE domain. For example, in networks where PEs are spokes connected to P routers, those PEs are Leaf BFERs unless there is a U-turn between two PEs.

All leaf-BFER in a BIER-TE domain can share a single BitPosition. This is possible because the BitPosition for the adjacency to reach the BFER can be used to distinguish whether or not packets should reach the BFER.

This optimization will not work if an upstream interface of the BFER is using a BitPosition optimized as described in the following two sections (LAN, Hub and Spoke).

4.4. LANs

In a LAN, the adjacency to each neighboring BFR on the LAN is given a unique BitPosition. The adjacency of this BitPosition is a forward_connected adjacency towards the BFR and this BitPosition is populated into the BIFT of all the other BFRs on that LAN.

```
  BFR1
    p1
LAN1--------
p3|  p4|  p2|
  BFR3 BFR4 BFR7
```

Figure 8: LAN Example

If Bandwidth on the LAN is not an issue and most BIER-TE traffic should be copied to all neighbors on a LAN, then BitPositions can be saved by assigning just a single BitPosition to the LAN and populating the BitPosition of the BIFTs of each BFRs on the LAN with
a list of forward_connected adjacencies to all other neighbors on the LAN.

This optimization does not work in the face of BFRs redundantly connected to more than one LANs with this optimization because these BFRs would receive duplicates and forward those duplicates into the opposite LANs. Adjacencies of such BFRs into their LANs still need a separate BitPosition.

4.5. Hub and Spoke

In a setup with a hub and multiple spokes connected via separate p2p links to the hub, all p2p links can share the same BitPosition. The BitPosition on the hubs BIFT is set up with a list of forward_connected adjacencies, one for each Spoke.

This option is similar to the BitPosition optimization in LANs: Redundantly connected spokes need their own BitPositions.

4.6. Rings

In L3 rings, instead of assigning a single BitPosition for every p2p link in the ring, it is possible to save BitPositions by setting the "Do Not Reset" (DNR) flag on forward_connected adjacencies.

For the rings shown in the following picture, a single BitPosition will suffice to forward traffic entering the ring at BFRa or BFRb all the way up to BFR1:

On BFRa, BFRb, BFR30,... BFR3, the BitPosition is populated with a forward_connected adjacency pointing to the clockwise neighbor on the ring and with DNR set. On BFR2, the adjacency also points to the clockwise neighbor BFR1, but without DNR set.

Handling DNR this way ensures that copies forwarded from any BFR in the ring to a BFR outside the ring will not have the ring BitPosition set, therefore minimizing the chance to create loops.
Note that this example only permits for packets to enter the ring at BFRa and BFRb, and that packets will always travel clockwise. If packets should be allowed to enter the ring at any ring BFR, then one would have to use two ring BitPositions. One for clockwise, one for counterclockwise.

Both would be set up to stop rotating on the same link, eg: L1. When the ingress ring BFR creates the clockwise copy, it will reset the counterclockwise BitPosition because the DNR bit only applies to the bit for which the replication is done. Likewise for the clockwise BitPosition for the counterclockwise copy. In result, the ring ingress BFR will send a copy in both directions, serving BFRs on either side of the ring up to L1.

4.7. Equal Cost MultiPath (ECMP)

The ECMP adjacency allows to use just one BP per link bundle between two BFRs instead of one BP for each p2p member link of that link bundle. In the following picture, one BP is used across L1,L2,L3 and BFR1/BFR2 have for the BP
---L1-----
BFR1 --L2----- BFR2
---L3-----

BIFT entry in BFR1:
<table>
<thead>
<tr>
<th>Index</th>
<th>Adjacencies</th>
</tr>
</thead>
<tbody>
<tr>
<td>0:6</td>
<td>ECMP((L1-to-BFR2,L2-to-BFR2,L3-to-BFR2), seed)</td>
</tr>
</tbody>
</table>

BIFT entry in BFR2:
<table>
<thead>
<tr>
<th>Index</th>
<th>Adjacencies</th>
</tr>
</thead>
<tbody>
<tr>
<td>0:6</td>
<td>ECMP((L1-to-BFR1,L2-to-BFR1,L3-to-BFR1), seed)</td>
</tr>
</tbody>
</table>

Figure 10: ECMP Example

In the following example, all traffic from BFR1 towards BFR10 is intended to be ECMP load split equally across the topology. This example is not meant as a likely setup, but to illustrate that ECMP can be used to share BPs not only across link bundles, and it explains the use of the seed parameter.
Figure 11: Polarization Example

With the setup of ECMP in above topology, traffic would not be equally load-split. Instead, links L22 and L31 would see no traffic at all: BFR2 will only see traffic from BFR1 for which the ECMP hash in BFR1 selected the first adjacency in a list of 2 adjacencies: link L11-to-BFR2. When forwarding in BFR2 performs again an ECMP with two adjacencies on that subset of traffic, then it will again select the first of its two adjacencies to it: L21-to-BFR4. And therefore L22 and BFR5 sees no traffic.

To resolve this issue, the ECMP adjacency on BFR1 simply needs to be set up with a different seed than the ECMP adjacencies on BFR2/BFR3.

This issue is called polarization. It depends on the ECMP hash. It is possible to build ECMP that does not have polarization, for example by taking entropy from the actual adjacency members into account, but that can make it harder to achieve evenly balanced load-
splitting on all BFR without making the ECMP hash algorithm potentially too complex for fast forwarding in the BFRs.

4.8. Routed adjacencies

4.8.1. Reducing BitPositions

Routed adjacencies can reduce the number of BitPositions required when the traffic engineering requirement is not hop-by-hop explicit path selection, but loose-hop selection.

```
...............             ...............
BFR1----. Redundant ...--L1-- BFR2... Redundant ...---
 \   ... Network ...--L2/--   ... Network ...---
BFR4----. Segment 1 ...--L3-- BFR3... Segment 2 ...---
...............             ...............
```

Figure 12: Routed Adjacencies Example

Assume the requirement in above network is to explicitly engineer paths such that specific traffic flows are passed from segment 1 to segment 2 via link L1 (or via L2 or via L3).

To achieve this, BFR1 and BFR4 are set up with a forward_routed adjacency BitPosition towards an address of BFR2 on link L1 (or link L2 BFR3 via L3).

For paths to be engineered through a specific node BFR2 (or BFR3), BFR1 and BFR4 are set up with a forward_routed adjacency BitPosition towards a loopback address of BFR2 (or BFR3).

4.8.2. Supporting nodes without BIER-TE

Routed adjacencies also enable incremental deployment of BIER-TE. Only the nodes through which BIER-TE traffic needs to be steered - with or without replication - need to support BIER-TE. Where they are not directly connected to each other, forward_routed adjacencies are used to pass over non BIER-TE enabled nodes.

5. Avoiding loops and duplicates

5.1. Loops

Whenever BIER-TE creates a copy of a packet, the BitString of that copy will have all BitPositions cleared that are associated with adjacencies in the BFR. This inhibits looping of packets. The only exception are adjacencies with DNR set.
With DNR set, looping can happen. Consider in the ring picture that
link L4 from BFR3 is plugged into the L1 interface of BFRa. This
creates a loop where the rings clockwise BitPosition is never reset
for copies of the packets traveling clockwise around the ring.

To inhibit looping in the face of such physical misconfiguration,
only forward_connected adjacencies are permitted to have DNR set, and
the link layer destination address of the adjacency (e.g.: MAC
address) protects against closing the loop. Link layers without port
unique link layer addresses should not used with the DNR flag set.

5.2. Duplicates

Duplicates happen when the topology of the BitString is not a tree
but redundantly connecting BFRs with each other. The controller must
therefore ensure to only create BitStrings that are trees in the
topology.

When links are incorrectly physically re-connected before the
controller updates BitStrings in BFIRs, duplicates can happen. Like
loops, these can be inhibited by link layer addressing in
forward_connected adjacencies.

If interface or loopback addresses used in forward_routed adjacencies
are moved from one BFR to another, duplicates can equally happen.
Such re-addressing operations must be coordinated with the
controller.

6. BIER-TE Forwarding Pseudocode

The following simplified pseudocode for BIER-TE forwarding is using
BIER forwarding pseudocode of [RFC8279], section 6.5 with the one
modification necessary to support basic BIER-TE forwarding. Like the
BIER pseudo forwarding code, for simplicity it does hide the details
of the adjacency processing inside PacketSend() which can be
forward_connected, forward_routed or local_decap.
void ForwardBitMaskPacket_withTE (Packet) {
    SI=GetPacketSI(Packet);
    Offset=SI*BitStringLength;
    for (Index = GetFirstBitPosition(Packet->BitString); Index ;
        Index = GetNextBitPosition(Packet->BitString, Index)) {
        F-BM = BIFT[Index+Offset]->F-BM;
        if (!F-BM) continue;
        BFR-NBR = BIFT[Index+Offset]->BFR-NBR;
        PacketCopy = Copy(Packet);
        PacketCopy->BitString &= F-BM;                  \[2\]
        PacketSend(PacketCopy, BFR-NBR);
        // The following must not be done for BIER-TE:
        // Packet->BitString &= ~F-BM;                  \[1\]
    }
}

Figure 13: Simplified BIER-TE Forwarding Pseudocode

The difference is that in BIER-TE, step [1] must not be performed.

In BIER, this step is necessary to avoid duplicates when two or more
BFER are reachable via the same neighbor. The F-BM of all those BFER
bits will indicate each others bits, and step [1] will reset all
these bits on the first copy made for the first of those BFER bits
set in the BitString, hence skipping any further copies to that
neighbor.

Whereas in BIER, the F-BM of bits toward a specific neighbor contain
only the bits of those BFER destined to be forwarded across this
neighbor, in BIER-TE the F-BM for a neighbor needs to have all bits
set except all those bits that are actual (non-empty) adjacencies of
this BFR. Step [2] will reset those adjacency bits to avoid loops,
but all the other bits that are not adjacencies of this BFR need to
stay untouched by [2] so that they can be processed by further BFR
along the path. If [1] was performed as in BIER, then those non-
adjacency bits would erroneously get reset during replication.

To support the DNR (Do Not Reset) flag of forward_connected()
adjacencies, the F-BM must also have its own bit set in the F-BM of
such an adjacency , so that for the packet copy made for this
adjacency the bit stays on, whereas it will not be set in the F-BM of
other bits so that it will be reset for any other packet copy made.

Eliminating the need to perform [1] also makes processing of bits in
the BIER-TE bitstring independent of processing other bits, which may
also simplify forwarding plane implementations.
The following pseudocode is comprehensive:

- This pseudocode eliminates per-bit F-BM, therefore reducing state by BitStringLength^2*SI and eliminating the need for per-packet-copy masking operation except for adjacencies with DNR flag set:
  - AdjacentBits[SI] are bits with a non-empty list of adjacencies. This can be computed whenever the BIER-TE controller host updates the adjacencies.
  - Only the AdjacentBits need to be examined in the loop for packet copies.
  - The packets BitString is masked with those AdjacentBits on ingress to avoid packet loopings.

- The code loops over the adjacencies because there may be more than one adjacency for a bit.

- When an adjacency has the DNR bit, the bit is set in the packet copy (to save bits in rings for example).

- The ECMP adjacency is shown. Its parameters are a ListOfAdjacencies from which one is picked.

- The forward_local, forward_routed, local_decap adjacencies are shown with their parameters.
void ForwardBitMaskPacket_withTE (Packet) {
    SI=GetPacketSI(Packet);
    Offset=SI*BitStringLength;
    AdjacentBitstring = Packet->BitString & ~AdjacentBits[SI];
    Packet->BitString &= AdjacentBits[SI];
    for (Index = GetFirstBitPosition(AdjacentBits); Index ;
        Index = GetNextBitPosition(AdjacentBits, Index)) {
        foreach adjacency BIFT[Index+Offset] {
            if(adjacency == ECMP(ListOfAdjacencies, seed) ) {
                I = ECMP_hash(sizeof(ListOfAdjacencies),
                              Packet->Entropy, seed);
                adjacency = ListOfAdjacencies[I];
            }
            PacketCopy = Copy(Packet);
            switch(adjacency) {
                case forward_connected(interface,neighbor,DNR):
                    if(DNR)
                        PacketCopy->BitString |= 2<<(Index-1);
                        SendToL2Unicast(PacketCopy,interface,neighbor);
                case forward_routed({VRF},neighbor):
                    SendToL3(PacketCopy, {VRF,}l3-neighbor);
                case local_decap({VRF},neighbor):
                    DecapBierHeader(PacketCopy);
                    PassTo(PacketCopy, {VRF,}Packet->NextProto);
            }
        }
    }
}

Figure 14: BIER-TE Forwarding Pseudocode

7. Managing SI, subdomains and BFR-ids

When the number of bits required to represent the necessary hops in
the topology and BFER exceeds the supported bitstring length,
multiple SI and/or subdomains must be used. This section discusses
how.

BIER-TE forwarding does not require the concept of BFR-id, but
routing underlay, flow overlay and BIER headers may. This section
also discusses how BFR-id can be assigned to BFIR/BFER for BIER-TE.
7.1. Why SI and sub-domains

For BIER and BIER-TE forwarding, the most important result of using multiple SI and/or sub-domains is the same: Packets that need to be sent to BFER in different SI or subdomains require different BIER packets: each one with a bitstring for a different (SI,subdomain) bitstring. Each such bitstring uses one bitstring length sized SI block in the BIFT of the subdomain. We call this a BIFT:SI (block).

For BIER and BIER-TE forwarding itself there is also no difference whether different SI and/or sub-domains are chosen, but SI and subdomain have different purposes in the BIER architecture shared by BIER-TE. This impacts how operators are managing them and how especially flow overlays will likely use them.

By default, every possible BFIR/BFER in a BIER network would likely be given a BFR-id in subdomain 0 (unless there are > 64k BFIR/BFER).

If there are different flow services (or service instances) requiring replication to different subsets of BFER, then it will likely not be possible to achieve the best replication efficiency for all of these service instances via subdomain 0. Ideal replication efficiency for N BFER exists in a subdomain if they are split over not more than ceiling(N/bitstring-length) SI.

If service instances justify additional BIER:SI state in the network, additional subdomains will be used: BFIR/BFER are assigned BFIR-id in those subdomains and each service instance is configured to use the most appropriate subdomain. This results in improved replication efficiency for different services.

Even if creation of subdomains and assignment of BFR-id to BFIR/BFER in those subdomains is automated, it is not expected that individual service instances can deal with BFER in different subdomains. A service instance may only support configuration of a single subdomain it should rely on.

To be able to easily reuse (and modify as little as possible) existing BIER procedures including flow-overlay and routing underlay, when BIER-TE forwarding is added, we therefore reuse SI and subdomain logically in the same way as they are used in BIER: All necessary BFIR/BFER for a service use a single BIER-TE BIFT and are split across as many SI as necessary (see below). Different services may use different subdomains that primarily exist to provide more efficient replication (and for BIER-TE desirable traffic engineering) for different subsets of BFIR/BFER.
7.2. Bit assignment comparison BIER and BIER-TE

In BIER, bitstrings only need to carry bits for BFER, which lead to the model that BFR-ids map 1:1 to each bit in a bitstring.

In BIER-TE, bitstrings need to carry bits to indicate not only the receiving BFER but also the intermediate hops/links across which the packet must be sent. The maximum number of BFER that can be supported in a single bitstring or BIFT:SI depends on the number of bits necessary to represent the desired topology between them.

"Desired" topology because it depends on the physical topology, and on the desire of the operator to allow for explicit traffic engineering across every single hop (which requires more bits), or reducing the number of required bits by exploiting optimizations such as unicast (forward_route), ECMP or flood (DNR) over "uninteresting" sub-parts of the topology - e.g.: parts where different trees do not need to take different paths due to traffic-engineering reasons.

The total number of bits to describe the topology in a BIFT:SI can therefore easily be as low as 20% or as high as 80%. The higher the percentage, the higher the likelihood, that those topology bits are not just BIER-TE overhead without additional benefit, but instead they will allow to express the desired traffic-engineering alternatives.

7.3. Using BFR-id with BIER-TE

Because there is no 1:1 mapping between bits in the bitstring and BFER, BIER-TE cannot simply rely on the BIER 1:1 mapping between bits in a bitstring and BFR-id.

In BIER, automatic schemes could assign all possible BFR-ids sequentially to BFERs. This will not work in BIER-TE. In BIER-TE, the operator or BIER-TE controller host has to determine a BFR-id for each BFER in each required subdomain. The BFR-id may or may not have a relationship with a bit in the bitstring. Suggestions are detailed below. Once determined, the BFR-id can then be configured on the BFER and used by flow overlay, routing underlay and the BIER header almost the same as the BFR-id in BIER.

The one exception are application/flow-overlays that automatically calculate the bitstring(s) of BIER packets by converting BFR-id to bits. In BIER-TE, this operation can be done in two ways:

"Independent branches": For a given application or (set of) trees, the branches from a BFIR to every BFER are independent of the
branches to any other BFER. For example, shortest part trees have independent branches.

"Interdependent branches": When a BFER is added or deleted from a particular distribution tree, branches to other BFER still in the tree may need to change. Steiner tree are examples of dependent branch trees.

If "independent branches" are sufficient, the BIER-TE controller host can provide to such applications for every BFR-id a SI:bitstring with the BIER-TE bits for the branch towards that BFER. The application can then independently calculate the SI:bitstring for all desired BFER by OR'ing their bitstrings.

If "interdependent branches" are required, the application could call a BIER-TE controller host API with the list of required BFER-id and get the required bitstring back. Whenever the set of BFER-id changes, this is repeated.

Note that in either case (unlike in BIER), the bits in BIER-TE may need to change upon link/node failure/recovery, network expansion and network load by other traffic (as part of traffic engineering goals). Interactions between such BFIR applications and the BIER-TE controller host do therefore need to support dynamic updates to the bitstrings.

7.4. Assigning BFR-ids for BIER-TE

For non-leaf BFER, there is usually a single bit k for that BFER with a local_decap() adjacency on the BFER. The BFR-id for such a BFER is therefore most easily the one it would have in BIER: SI * bitstring-length + k.

As explained earlier in the document, leaf BFER do not need such a separate bit because the fact alone that the BIER-TE packet is forwarded to the leaf BFER indicates that the BFER should decapsulate it. Such a BFER will have one or more bits for the links leading only to it. The BFR-id could therefore most easily be the BFR-id derived from the lowest bit for those links.

These two rules are only recommendations for the operator or BIER-TE controller assigning the BFR-ids. Any allocation scheme can be used, the BFR-ids just need to be unique across BFRs in each subdomain.

It is not currently determined if a single subdomain could or should be allowed to forward both BIER and BIER-TE packets. If this should be supported, there are two options:
A. BIER and BIER-TE have different BFR-id in the same subdomain. This allows higher replication efficiency for BIER because their BFR-id can be assigned sequentially, while the bitstrings for BIER-TE will have also the additional bits for the topology. There is no relationship between a BFR BIER BFR-id and BIER-TE BFR-id.

B. BIER and BIER-TE share the same BFR-id. The BFR-id are assigned as explained above for BIER-TE and simply reused for BIER. The replication efficiency for BIER will be as low as that for BIER-TE in this approach. Depending on topology, only the same 20%..80% of bits as possible for BIER-TE can be used for BIER.

7.5. Example bit allocations

7.5.1. With BIER

Consider a network setup with a bitstring length of 256 for a network topology as shown in the picture below. The network has 6 areas, each with ca. 180 BFR, connecting via a core with some larger (core) BFR. To address all BFER with BIER, 4 SI are required. To send a BIER packet to all BFER in the network, 4 copies need to be sent by the BFIR. On the BFIR it does not make a difference how the BFR-id are allocated to BFER in the network, but for efficiency further down in the network it does make a difference.

```
area1  area2  area3
BFR1a BFR1b | BFR2a BFR2b | BFR3a BFR3b
 | \   /   /   /   |
 | .    Core .    |
 | \  /   /   /   |
BFR4a BFR4b BFR5a BFR5b BFR6a BFR6b
```

Figure 15: Scaling BIER-TE bits by reuse

With random allocation of BFR-id to BFER, each receiving area would (most likely) have to receive all 4 copies of the BIER packet because there would be BFR-id for each of the 4 SI in each of the areas. Only further towards each BFER would this duplication subside - when each of the 4 trees runs out of branches.

If BFR-id are allocated intelligently, then all the BFER in an area would be given BFR-id with as few as possible different SI. Each area would only have to forward one or two packets instead of 4.
Given how networks can grow over time, replication efficiency in an area will also easily go down over time when BFR-id are network wide allocated sequentially over time. An area that initially only has BFR-id in one SI might end up with many SI over a longer period of growth. Allocating SIs to areas with initially sufficiently many spare bits for growths can help to alleviate this issue. Or renumber BFR-id after network expansion. In this example one may consider to use 6 SI and assign one to each area.

This example shows that intelligent BFR-id allocation within at least subdomain 0 can even be helpful or even necessary in BIER.

7.5.2. With BIER-TE

In BIER-TE one needs to determine a subset of the physical topology and attached BFER so that the "desired" representation of this topology and the BFER fit into a single bitstring. This process needs to be repeated until the whole topology is covered.

Once bits/SIs are assigned to topology and BFER, BFR-id is just a derived set of identifiers from the operator/BIER-TE controller as explained above.

Every time that different sub-topologies have overlap, bits need to be repeated across the bitstrings, increasing the overall amount of bits required across all bitstring/SIs. In the worst case, random subsets of BFER are assigned to different SI. This is much worse than in BIER because it not only reduces replication efficiency with the same number of overall bits, but even further - because more bits are required due to duplication of bits for topology across multiple SI. Intelligent BFER to SI assignment and selecting specific "desired" subtopologies can minimize this problem.

To set up BIER-TE efficiently for above topology, the following bit allocation methods can be used. This method can easily be expanded to other, similarly structured larger topologies.

Each area is allocated one or more SI depending on the number of future expected BFER and number of bits required for the topology in the area. In this example, 6 SI, one per area.

In addition, we use 4 bits in each SI: bia, bib, bea, beb: bit ingress a, bit ingress b, bit egress a, bit egress b. These bits will be used to pass BIER packets from any BFIR via any combination of ingress area a/b BFR and egress area a/b BFR into a specific target area. These bits are then set up with the right forward_routed adjacencies on the BFIR and area edge BFR:
On all BFIR in an area j, bia in each BIFT:SI is populated with the same forward_routed(BFRja), and bib with forward_routed(BFRjb). On all area edge BFR, bea in BIFT:SI=k is populated with forward_routed(BFRka) and beb in BIFT:SI=k with forward_routed(BFRkb).

For BIER-TE forwarding of a packet to some subset of BFER across all areas, a BFIR would create at most 6 copies, with SI=1...SI=6. In each packet, the bits indicate bits for topology and BFER in that topology plus the four bits to indicate whether to pass this packet via the ingress area a or b border BFR and the egress area a or b border BFR, therefore allowing path engineering for those two "unicast" legs: 1) BFIR to ingress area edge and 2) core to egress area edge. Replication only happens inside the egress areas. For BFER in the same area as in the BFIR, these four bits are not used.

7.6. Summary

BIER-TE can like BIER support multiple SI within a sub-domain to allow re-using the concept of BFR-id and therefore minimize BIER-TE specific functions in underlay routing, flow overlay methods and BIER headers.

The number of BFIR/BER possible in a subdomain is smaller than in BIER because BIER-TE uses additional bits for topology.

Subdomains can in BIER-TE be used like in BIER to create more efficient replication to known subsets of BFER.

Assigning bits for BFER intelligently into the right SI is more important in BIER-TE than in BIER because of replication efficiency and overall amount of bits required.

8. BIER-TE and Segment Routing (SR)

Segment Routing (SR ([RFC8402])) aims to enable lightweight path engineering via loose source routing. Compared to its more heavy-weight predecessor RSVP-TE ([RFC3209]), SR does for example not require per-path signaling to each of these hops.

BIER-TE supports the same design philosophy for multicast. Like in SR, it relies on source-routing – via the definition of a BitString. Like SR, it only requires to consider the "hops" on which either replication has to happen, or across which the traffic should be steered (even without replication). Any other hops can be skipped via the use of routed adjacencies.
BIER-TE BitPosition (BP) can be understood as the BIER-TE equivalent of "forwarding segments" in SR, but they have a different scope than SR forwarding segments. Whereas forwarding segments in SR are global or local, BPs in BIER-TE have a scope that is the group of BFR(s) that have adjacencies for this BP in their BIFT. This can be called "adjacency" scoped forwarding segments.

Adjacency scope could be global, but then every BFR would need an adjacency for this BP, for example a forward_routed adjacency with encapsulation to the global SR SID of the destination. Such a BP would always result in ingres replication though. The first BFR encountering this BP would directly replicate to it. Only by using non-global adjacency scope for BPs can traffic be steered and replicated on non-ingres BFR.

SR can naturally be combined with BIER-TE and help to optimize it. For example, instead of defining BitPositions for non-replicating hops, it is equally possible to use segment routing encapsulations (eg: MPLS label stacks) for the encapsulation of "forward_routed" adjacencies.

Note that BIER itself can also be seen to be similar to SR. BIER BPs act as global destination Node-SIDs and the BIER bitstring is simply a highly optimized mechanism to indicate multiple such SIDs and let the network take care of effectively replicating the packet hop-by-hop to each destination Node-SID. What BIER does not allow is to indicate intermediate hops, or terms of SR the ability to indicate a sequence of SID to reach the destination. This is what BIER-TE and its adjacency scoped BP enables.

9. Security Considerations

The security considerations are the same as for BIER with the following differences:

BFR-ids and BFR-prefixes are not used in BIER-TE, nor are procedures for their distribution, so these are not attack vectors against BIER-TE.

10. IANA Considerations

This document requests no action by IANA.

11. Acknowledgements

The authors would like to thank Greg Shepherd, Ijsbrand Wijnands and Neale Ranns for their extensive review and suggestions.
12. Change log [RFC Editor: Please remove]

draft-ietf-bier-te-arch:

03: Last call textual changes by authors to improve readability:
removed Wolfgang Braun as co-authors (as requested).
Improved abstract to be more explanatory. Removed mentioning of
FRR (not conluded on so far).
Added new text into Introduction section because the text was too
difficult to jump into (too many forward pointers). This
primarily consists of examples and the early introduction of the
BIER-TE Topology concept enabled by these examples.
Amended comparison to SR.
Changed syntax from [VRF] to {VRF} to indicate its optional and to
make idnits happy.
Split references into normative / informative, added references.

02: Refresh after IETF104 discussion: changed intended status back
to standard. Reasoning:
Tighter review of standards document == ensures arch will be
better prepared for possible adoption by other WGs (e.g.: DetNet)
or std. bodies.
Requirement against the degree of existing implementations is self
defined by the WG. BIER WG seems to think it is not necessary to
apply multiple interoperating implementions against an
architecture level document at this time to make it qualify to go
to standards track. Also, the levels of support introduced in -01
rev. should allow all BIER forwarding engines to also be able to
support the base level BIER-TE forwarding.

01: Added note comparing BIER and SR to also hopefully clarify
BIER-TE vs. BIER comparison re. SR.
- added requirements section mandating only most basic BIER-TE
forwarding features as MUST.
- reworked comparison with BIER forwarding section to only
summarize and point to pseudocode section.
- reworked pseudocode section to have one pseudocode that mirrors the BIER forwarding pseudocode to make comparison easier and a second pseudocode that shows the complete set of BIER-TE forwarding options and simplification/optimization possible vs. BIER forwarding.

- Added captions to pictures.

00: Changed target state to experimental (WG conclusion), updated references, mod auth association.

- Source now on http://www.github.com/toerless/bier-te-arch

- Please open issues on the github for change/improvement requests to the document - in addition to posting them on the list (bier@ietf.). Thanks!.

draft-eckert-bier-te-arch:

06: Added overview of forwarding differences between BIER, BIER-TE.

05: Author affiliation change only.

04: Added comparison to Live-Live and BFIR to FRR section (Eckert).

04: Removed FRR content into the new FRR draft [I-D.eckert-bier-te-frr] (Braun).

- Linked FRR information to new draft in Overview/Introduction
- Removed BTAFT/FRR from "Changes in the network topology"
- Linked new draft in "Link/Node Failures and Recovery"
- Removed FRR from "The BIER-TE Forwarding Layer"
- Moved FRR section to new draft
- Moved FRR parts of Pseudocode into new draft
- Left only non FRR parts
- removed FrrUpDown(..) and //FRR operations in ForwardBierTePacket(..)
- New draft contains FrrUpDown(..) and ForwardBierTePacket(Packet)
  from bier-arch-03

- Moved "BIER-TE and existing FRR to new draft

- Moved "BIER-TE and Segment Routing" section one level up

- Thus, removed "Further considerations" that only contained this
  section

- Added Changes for version 04

03: Updated the FRR section. Added examples for FRR key concepts.
Added BIER-in-BIER tunneling as option for tunnels in backup
paths. BIFT structure is expanded and contains an additional
match field to support full node protection with BIER-TE FRR.

03: Updated FRR section. Explanation how BIER-in-BIER
encapsulation provides P2MP protection for node failures even
though the routing underlay does not provide P2MP.

02: Changed the definition of BIFT to be more inline with BIER.
In revs. up to -01, the idea was that a BIFT has only entries for
a single bitstring, and every SI and subdomain would be a separate
BIFT. In BIER, each BIFT covers all SI. This is now also how we
define it in BIER-TE.

02: Added Section 7 to explain the use of SI, subdomains and BFR-
id in BIER-TE and to give an example how to efficiently assign
bits for a large topology requiring multiple SI.

02: Added further detailed for rings - how to support input from
all ring nodes.

01: Fixed BFIR -> BFER for section 4.3.

01: Added explanation of SI, difference to BIER ECMP,
consideration for Segment Routing, unicast FRR, considerations for
encapsulation, explanations of BIER-TE controller host and CLI.

00: Initial version.
13. References

13.1. Normative References


13.2. Informative References


Authors’ Addresses

Toerless Eckert (editor)
Huawei USA - Futurewei Technologies Inc.
2330 Central Expy
Santa Clara  95050
USA

Email: tte+ietf@cs.fau.de

Gregory Cauchie
Bouygues Telecom

Email: GCAUCHIE@bouyguestelecom.fr

Eckert, et al. Expires January 9, 2020
Encapsulation for BIER in Non-MPLS IPv6 Networks
draft-xie-bier-ipv6-encapsulation-02

Abstract

This document proposes a BIER IPv6 (BIERv6) encapsulation for Non- MPLS IPv6 Networks using the IPv6 Destination Option extension header.

Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119] and [RFC8174].

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 January 2, 2020.
1. Introduction

Bit Index Explicit Replication (BIER) [RFC8279] is an architecture that provides optimal multicast forwarding without requiring intermediate routers to maintain any per-flow state by using a multicast-specific BIER header.

[RFC8296] defines a common BIER Header format for MPLS and Non-MPLS networks. It has defined two types of encapsulation methods using the common BIER Header, (1) BIER encapsulation in MPLS networks, here-in after referred as MPLS BIER Header in this document and (2) BIER encapsulation in Non-MPLS networks, here-in after referred as Non-MPLS BIER Header in this document. [RFC8296] also assigned
This document proposes a BIER IPv6 encapsulation for Non-MPLS IPv6 Networks, defining a method to carry the standard Non-MPLS BIER header (as defined in [RFC8296]) in the native IPv6 header. A new IPv6 Option type - BIER Option is defined to encode the standard Non-MPLS BIER header and this newly defined BIER Option is carried under the Destination Options header of the native IPv6 Header [RFC8200].

This document details one of the proposed solutions for transporting BIER packets in an IPv6 network. To better understand the overall BIER IPv6 problem space, use cases and proposed solutions, refer to [I-D.ietf-bier-ipv6-requirements].

2. Terminology

Readers of this document are assumed to be familiar with the terminology and concepts of the documents listed as Normative References.

The following new terms are used throughout this document:

- BIERv6 - BIER IPv6.
- BIER Option - An Option type carried in IPv6 Destination Options Header which includes the standard Non-MPLS BIER Header.
- BIERv6 Header - An IPv6 Header with BIER Option.
- BIERv6 Packet - An IPv6 packet with BIERv6 Header. Such an IPv6 packet typically carries the user multicast payload and is forwarded by BFRs in the BIERv6 network towards the multicast receivers.
- BIER Multicast Address - A well-known multicast address used as a Destination Address in the BIERv6 Header to forward the packets to other BFRs in BIERv6 network.

3. BIER IPv6 Encapsulation

3.1. BIER Option in IPv6 Destination Options Header

Destination Options Header and the Options that can be carried under this extension header is defined in [RFC8200]. This document defines a new Option type - BIER Option, to encode the Non-MPLS BIER header. As specified in Section 4.2 [RFC8200], the BIER Option follows type-length-value (TLV) encoding format and the standard Non-MPLS BIER
header [RFC8296] is encoded in the value portion of the BIER Option TLV.

This BIER Option MUST be carried only inside the IPv6 Destination Options header and MUST NOT be carried under the Hop-by-Hop Options header.

Co-existence of Destination Options Header with BIER option TLV and other IPv6 extension headers MUST confirm to the general requirements defined in [RFC8200]. In addition to the requirements defined in [RFC8200], this document requires that the Destination Options Header with a BIER Option TLV MUST appear only after the Routing Header if the Routing Header is present in the IPv6 Header.

The BIER Option is encoded in type-length-value (TLV) format as follows:

```
0                   1                   2                   3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Next Header  |  Hdr Ext Len  |  Option Type  | Option Length |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
Non-MPLS BIER Header (defined in RFC8296)                      |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Next Header  8-bit selector. Identifies the type of header immediately following the Destination Options header.

Hdr Ext Len  8-bit unsigned integer. Length of the Destination Options header in 8-octet units, not including the first 8 octets.

Option Type  To be allocated by IANA. See section 6.

Option Length  8-bit unsigned integer. Length of the option, in octets, excluding the Option Type and Option Length fields.

Non-MPLS BIER Header  The Non-MPLS BIER Header defined in RFC8296. Fields in the Non-MPLS BIER Header MUST be encoded as below.

```
BIFT-id: The BIFT-id is a domain-wide unique value in Non-MPLS IPv6 encapsulation. See Section 2.2 of RFC 8296.

TC: SHOULD be set to binary value 000 upon transmission and MUST be ignored upon. See Section 2.2 of RFC 8296.
```
S bit: SHOULD be set to 1 upon transmission, and MUST be ignored upon reception. See Section 2.2 of RFC 8296.

TTL: MUST be set to 0 upon transmission, and MUST be ignored upon reception. The function of TTL is replaced by the Hop Limit field in IPv6 header.

Nibble: SHOULD be set to 0000 upon transmission, and MUST be ignored upon reception. See Section 2.2 of RFC 8296.

Ver: MUST be set to 0 upon transmission, and MUST be discarded when it is not 0 upon reception. See Section 2.2 of RFC 8296.

BSL: See Section 2.1.2 of RFC 8296.

Entropy: See Section 2.1.2 of RFC 8296.

OAM: See Section 2.1.2 of RFC 8296.

Rsv: See Section 2.1.2 of RFC 8296.

DSCP: SHOULD be set to binary value 000000 upon transmission and MUST be ignored upon reception. In IPv6 BIER encapsulation, uses highest 6-bit of Traffic Class field of IPv6 header to hold a Differentiated Services Codepoint [RFC2474].

Proto: SHOULD be set to 0 upon transmission and MUST be ignored upon reception. In IPv6 BIER encapsulation, the functionality of this 6-bit Proto field is replaced by the Next Header field in Destination Options header, which is the last IPv6 extension header, to indicate the BIER payload, which is also IPv6 payload.

For BIER Proto 1, indicating a Downstream-assigned MPLS payload, use Next Header value 137.

For BIER Proto 2, indicating an Upstream-assigned MPLS payload, there is no Next Header code currently. An upstream-assigned MPLS label within the context of special BFIR router, which in turn is represented by the BFIR-id and the Sub-domain indirectly indicated by the BIFT-id in a BIER-MPLS or BIER-ETH packet, can be replaced by an IPv6 source address in a BIER IPv6 encapsulation packet in a direct manner. In this case, use Next Header value 4 for IPv4 payload, or value 41 for IPv6 payload.
For BIER Proto 3, indicating an Ethernet payload, use Next Header value 97.

For BIER Proto 4, indicating an IPv4 payload, use Next Header value 4.

For BIER Proto 5, indicating a BIER-OAM payload, use Next Header value 58. How the BIER-PING is supported with BIER IPv6 encapsulation is outside the scope of this document.

For BIER Proto 6, indicating an IPv6 payload, use Next Header value 41.

BFIR-id: See Section 2.1.2 of RFC 8296.

BitString: See Section 2.1.2 of RFC 8296.

3.2. Multicast and Unicast Destination Address

BIER is generally a hop-by-hop and one-to-many architecture, and thus the IPv6 Destination Address (DA) being a Multicast Address is a proper approach for both the two paradigms in BIERv6 encapsulation.

This document proposes to use multicast address FF0X::AB37 (to be allocated and reserved by IANA - See Section 6.2) as the IPv6 destination address for the BIERv6 packets to be forwarded in the BIER domain.

All the interfaces of the BFRs supporting the BIERv6 encapsulation defined in this document MUST subscribe and listen to BIER multicast address FF0X::AB37 belong to scopes [1, 2, 3, 4, 5, E] defined in [RFC7346]. However it is RECOMMENDED to use Realm-Local scope (scope value 3), that is FF03:AB37 as a destination address while forwarding the BIERv6 packet, as this scope zone is exactly the BIERv6 Domain. The use of other scopes is outside the scope of this document.

Use of a Unicast Address as a IPv6 Destination Address is permissible and useful in certain cases.

1. Tunneling a BIERv6 packet over a non-BIER capable router.

2. Fast rerouting a BIERv6 packet using a unicast by-pass tunnel.

3. Forwarding a BIERv6 packet to one of the many BFR neighbors connected on a LAN.

4. Connecting BIER domains, for example Data Center domains, in an overlay manner.
The unicast address used in BIERv6 packet targeting a BFR SHOULD be the IPv6 BFR-Prefix advertised from this BFR. When a BFR advertises the BIER information with BIERv6 encapsulation capability, the IPv6 BFR-prefix of this BFR MUST be selected specifically for BIERv6 packet forwarding. Locally this "BIER Specific" IPv6 address is initialized in FIB with a flag of "BIER specific handling", represented as End.BIER function. For convenience, the indication in FIB share the same space as SRv6 Endpoints Behaviors defined in [I-D.ietf-spring-srv6-network-programming]. Apart from this sharing of code space, there is nothing dependent on SRv6. The co-existence of BIERv6 and SRv6 is outside the scope of this document.

BFR Prefix is used only in control plane in BIER MPLS encapsulation but not used in data plane. While in BIERv6, BFR prefix is used in both control plane and data plane. The "BIER Specific" IPv6 address can be used for BIER MPLS in control plane too. So it is RECOMMENDED to use a "BIER specific" IPv6 address as BFR prefix when deploying BIER in IPv6 network from the scratch. One should be careful not use the IPv6 address selected as BFR prefix for other purpose like BGP session until the "BIER specific handling" can do more general process.

The following is an example of configuring a BIER specific IPv6 address and using this address as BFR prefix:

```
# Config a BIER specific IPv6 address with 128-bit mask on loopback0.
interface loopback0
  ipv6 address 2019::AB37 128 End.BIER

# Config the BIER-specific IPv6 address on loopback0 as BFR Prefix.
bier sub-domain 6 ipv6-underlay
  bfr-prefix interface loopback0
```

The address used as "BIER specific" IPv6 address can be from inside the scope of an SRv6 Locator or outside the scope of the SRv6 Locator(s) since it is a host prefix (128-bit prefix-length prefix).

Each "BIER specific" address can be used in one or many sub-domains as BFR-prefix, such that it can be associated with one or many Multi-Topologies (MTs) or algorithms.

More than one "BIER specific" address are also allowed as different BFR-prefix of more than one sub-domain, as described in section 2 of [RFC8279].

The following is an example pseudo-code of the End.BIER function:
1. IF NH = 60 AND HopLimit > 0 ;;Ref1
2. IF (OptType1 = BIER) and (OptLength1 = HdrExtLen*8 + 4) ;;Ref2
3. Lookup the BIER Header inside the BIER option TLV.
4. Forward via the matched entry.
5. ELSE
6. Drop the packet.
7. ELSE IF Last_NH = ICMPv6 ;;Ref3
8. Send to CPU.
9. ELSE
10. Drop the packet.

Ref1: Destination options header follows the IPv6 header directly and HopLimit is bigger than zero.

Ref2: The first TLV is BIER type and is the only one in Destination options header.

Ref3: An ICMPv6 packet using End.BIER as destination address.

3.3. BIERv6 Packet Format

As a multicast packet enters the BIER domain in a Non-MPLS IPv6 network, the multicast packet will be encapsulated with BIERv6 Header.

Typically a BIERv6 header would contain the Destination Options Header as the only Extensions Header besides IPv6 Header. However, it is allowed and possible for other extension headers to appear along with the Destination Options Header as long as the requirements listed in section 3.1 of this document is met.

Format of the multicast packet with BIERv6 encapsulation carrying only the Destination Options header is depicted in the below figure.

```
+---------------+--------------+------------
| IPv6 header   | Dest Options | X type of  |
|               | Header with  | multicast  |
|               | BIER Option  | packet     |
| Next Hdr = 60 | Nxt Hdr = X  |
+---------------+--------------+------------
```

Format of the multicast packet with BIERv6 encapsulation carrying other extension headers along with Destination Options extension header is required to follow general recommendations of [RFC8200] and examples in other RFCs. [RFC6275] introduces how the order should be when other extension headers carries along with Home address option in a destination options header. Similar to this example, this
document requires the Destination Options Header carrying the BIER option MUST be placed as follows:

- After the routing header, if that header is present
- Before the Fragment Header, if that header is present
- Before the AH Header or ESP Header, if either one of those headers is present

Source Address field in the IPv6 header MUST be a routable IPv6 unicast address of the BFIR in any case.

BFIR encodes the Non-MPLS BIER header in the above mentioned encapsulation format and forwards the BIERv6 packet to the next-hop BFR following the local BIFT table.

BFRs in the IPv6 network, processes and replicates the packets towards the BFERs using the local BIFT table. The bit-string field in the Non-MPLS BIER header may be changed by the BFRs as they replicate the packet. BFRs MUST follow the procedures defined in section 3.1 as they modify the other fields in the Non-MPLS BIER header. The source address in the IPv6 header MUST NOT be modified by the BFRs.

4. BIERv6 Packet Processing

There is no BIER-specific processing, and all the 8 steps in section 6.5 of RFC8279 apply to BIERv6 packet processing. However, there are some IPv6-specific processing procedures due to the base and general procedures of IPv6.

On the overlay layer, when a multicast packet enters the BIER domain in a Non-MPLS IPv6 network, the Ingress BFR (BFIR) encapsulates the multicast packet with a BIERv6 Header, transforming it to a BIERv6 packet. The BIERv6 header includes an IPv6 header and IPv6 Destination Options Header within a standard Non-MPLS BIER header. Source Address field in the IPv6 header MUST be set to a routable IPv6 unicast address of the BFIR. Destination Address field in the IPv6 header is set to a BIER multicast address, FF0X::AB37, if the next-hop BFR is directly connected, or MAY be set to a unicast address in case of the scenarios discussed in section 3.2.

On the BIER layer, upon receiving an BIERv6 packet, the BFR processes the IPv6 header first. This is the general procedure of IPv6.

If the IPv6 Destination address is the BIER multicast address, a ‘BIER Specific Handling’ indication will be obtained by the preceding
Multicast DA lookup (MFIB lookup). The BIER option, if exists, will be checked to decide which neighbor(s) to replicate the BIERv6 packet to.

If the IPv6 Destination address is an IPv6 BFR-Prefix unicast address of this BFR, a 'BIER Specific Handling' indication will be obtained by the preceding Unicast DA lookup (FIB lookup). The BIER option, if exists, will be checked to decide which neighbor(s) to replicate the BIERv6 packet to.

It is a local behavior to handle the combination of extension headers, options and the BIER option(s) in destination options header when a 'BIER Specific Handling' indication is got by the preceding MFIB or FIB lookup. Early deployment of BIERv6 may require there is only one BIER option TLV in the destination options header followed the IPv6 header. How other extension headers or more BIER option TLVs in a BIERv6 packet is handled is outside the scope of this document.

A packet having a 'BIER Specific Handling' indication but not having a BIER option MAY be processed normally as normal multicast or unicast forwarding procedures do, or MAY be dropped.

A packet not having a 'BIER Specific Handling' indication but having a BIER option SHOULD be processed normally as normal multicast or unicast forwarding procedures, which may be a behavior of drop, or send to CPU, or other behaviors in existing implementations.

The Destination Address field in the IPv6 Header MUST change to the nexthop BFR's BFR Prefix if Unicast address is used in BIERv6.

The Hop Limit field of IPv6 header MUST decrease by 1 when sending packets to a BFR neighbor, while the TTL in the BIER header MUST be unchanged.

The BitString in the BIER header in the Destination Options Header may change when sending packets to a neighbor. Such change of BitString MUST be aligned with the procedure defined in RFC8279. Because of the requirement to change the content of the option when forwarding BIERv6 packet, the BIER option type should have chg flag 1 per section 4.2 of RFC8200.

The procedures applies normally if a bit corresponding to the self bfr-id is set in the bit-string field of the Non-MPLS BIER header of the BIERv6 packet. The node is considered to be an Egress BFR (BFER) in this case. The BFER removes the BIERv6 header, including the IPv6 header and the Destination Options header, and copies the packet to the multicast flow overlay. The egress VRF of a packet may be
5. Security Considerations

A BIERv6 packet with a special IPv6 Destination Address, either multicast or unicast, would be processed by BIER forwarding procedure only when the ‘BIER valid’ flag has been obtained ahead of time in the normal MFIB or FIB lookup of the IPv6 header. Otherwise the packet with an IPv6 BIER Option will be dropped, as if the Option is not recognize by the node.

An IPv6 packet with BIER multicast address FF0X::AB37 as destination address, but does not carry IPv6 BIER Option will be dropped.

6. IANA Considerations

6.1. BIER Option Type

Allocation is expected from IANA for a BIER Option Type codepoint from the "Destination Options and Hop-by-Hop Options" sub-registry of the "Internet Protocol Version 6 (IPv6) Parameters" registry. The value 0x70 is suggested.

```
+-----------+-----+-----+-------+-------------+------------+
| Hex Value | act | chg | rest | Description | Reference  |
+-----------+-----+-----+-------+-------------+------------+
|    0x70   |  01 |  1  | 10000 | BIER Option | This draft |
```

Figure 1: IPv6 Option Type Suggested

6.2. BIER Multicast Address

Allocation is expected from IANA for a BIER Multicast Address from the "Variable Scope Multicast Addresses" sub-registry of the "IPv6 Multicast Address Space Registry" registry. The address ‘FF0X::AB37’ is suggested.

```
<table>
<thead>
<tr>
<th>Address(es)</th>
<th>Description</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>FF0X:0:0:0:0:0:0:AB37</td>
<td>ALL_BIER_FORWARDERS</td>
<td>This draft</td>
</tr>
</tbody>
</table>
```

Figure 2: Multicast Address Suggested
6.3. End.BIER Function

Allocation is expected from IANA for an End.BIER function codepoint from the "SRv6 Endpoint Behaviors" sub-registry. The value 60 is suggested.

<table>
<thead>
<tr>
<th>Value</th>
<th>Hex</th>
<th>Endpoint function</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>TBD</td>
<td>TBD</td>
<td>End.BIER</td>
<td>This draft</td>
</tr>
</tbody>
</table>

Figure 3: End.BIER Function

7. Acknowledgements

The authors would like to thank Stig Venaas for his valuable comments. Thanks IJsbrand Wijnands, Greg Shepherd, Tony Przygienda, Toerless Eckert, Jeffrey Zhang for the helpful comments to improve this document.

8. References

8.1. Normative References


8.2. Informative References

[I-D.ietf-bier-ipv6-requirements]

[I-D.ietf-spring-srv6-network-programming]


Authors’ Addresses

Jingrong Xie
Huawei Technologies
Email: xiejingrong@huawei.com

Liang Geng
China Mobile
Beijing 10053
Email: gengliang@chinamobile.com

Mike McBride
Futurewei
Email: mmcbride7@gmail.com
Abstract

This document defines IS-IS extensions to support multicast forwarding using the Bit Index Explicit Replication (BIER) with IPv6 encapsulation (BIERv6).

Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119] and [RFC8174].

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 January 2, 2020.

Copyright Notice

Copyright (c) 2019 IETF Trust and the persons identified as the document authors. All rights reserved.
1. Introduction

This document defines IS-IS extensions to support multicast forwarding using the Bit Index Explicit Replication (BIER) with IPv6 encapsulation (BIERV6).

Familiarity with the concept of "BIER specific" IPv6 address introduced in [I-D.xie-bier-ipv6-encapsulation] is necessary to understand the extensions specified in this document.

The [I-D.ietf-spring-srv6-network-programming] describes how a function can be bound to a special "IPv6 Address" within a special "IPv6 Address Block". The function bound to a special "IPv6 Address" can be used to indicate a special forwarding process in data-plane.

The BIER IPv6 encapsulation [I-D.xie-bier-ipv6-encapsulation] uses a "BIER specific" IPv6 unicast address configured locally on a BIER Forwarding Router (BFR) to indicate a "BIER specific handling" in Forwarding Information Base (FIB). This BIER specific IPv6 address is also required to use as the BFR prefix as defined in [RFC8279].
The indication of BFR prefix is a BIER Sub-TLV within the extended IP reachability TLV as specified by in [RFC8401].

The indication of BIER specific function is a "Function Sub-TLV" within the extended IP reachability as specified by in this document.

Note the extended IP reachability only includes the TLV 236 (IPv6 IP Reach TLV) [RFC5308] and TLV 237 (MT IPv6 IP Reach TLVs) [RFC5120] in this document.

The following restrictions defined for BIER Sub-TLV in section 4.2 of [RFC8401] apply equally to Function Sub-TLV:

- Prefix length MUST be 128 for an IPv6 prefix.
- When the Prefix Attributes Flags sub-TLV [RFC7794] is present, the N flag MUST be set and the R flag MUST NOT be set.
- BIER sub-TLVs and Function Sub-TLVs MUST be included when a prefix reachability advertisement is leaked between levels.

2. Terminology

Readers of this document are assumed to be familiar with the terminology and concepts of the documents listed as Normative References.

3. Specification

3.1. Function sub-TLV for BIERv6

The Function sub-TLV is introduced to advertise a specified function bound to an IPv6 prefix with 128 bit prefix length. This new sub-TLV is advertised in the TLV 236 or TLV 237. The sub-TLV has the following format:

```
   0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   Type        |   Length     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   Function    |   Flags      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

Type: 1 octet value indicating "Function Information" this IPv6 prefix bound to. To be assigned by IANA.

Length: 1 octet length in octets. Value 4 is set to this field.
Function: 2 octets value indicating function. A BIER function value called End.BIER defined in [I-D.xie-bier-ipv6-encapsulation] is expected to be the only function in the TLV.

Flags: 1 octet value indicating the Flags for the function preceding this field. No flags are currently defined and 0 should be set for this field.

3.2. Encapsulation sub-sub-TLV for BIERv6

The Encapsulation sub-sub-TLV carries the information for the BIER IPv6 encapsulation of a specific BitString length. It is advertised within the BIER Info sub-TLV defined in [RFC8401] which in-turn is carried within the TLVs 236 or 237. This sub-sub-TLV MAY appear multiple times within a single BIER Info sub-TLV. If the same BitString length is repeated in multiple sub-sub-TLVs inside the same BIER Info sub-TLV, the BIER Info sub-TLV MUST be ignored.

```
| Type | Length |
+-------+--------+
| Max SI | BS Len |
+-------+--------+
```

The Type field is a 1 octet value indicating BIER IPv6 encapsulation. To be assigned by IANA.

The Length field is a 1 octet length in octets. Value 4 is set to this field.

Other fields can be referred to [RFC8401] for MPLS encapsulation, or [I-D.ietf-bier-lsr-ethernet-extensions] for Ethernet encapsulation.

4. Security Considerations

The procedures of this document do not, in themselves, provide privacy, integrity, or authentication for the control plane or the data plane.

5. IANA Considerations

5.1. Function sub-TLV Type Code

Allocation is expected from IANA for a IS-IS Sub-TLV Type codepoint from the "Sub-TLVs for TLVs 135, 235, 236, and 237" sub-registry.
5.2. Encapsulation sub-sub-TLV Type Code

Allocation is expected from IANA for a BIER IPv6 encapsulation sub-sub-TLV codepoint from the "sub-sub-TLVs for BIER Info sub-TLV" sub-registry.

Type: To be assigned by IANA.

Name: BIER IPv6 Encapsulation.

Reference: This document.

6. Acknowledgements

TBD.

7. References

7.1. Normative References

[I-D.ietf-bier-lsr-ethernet-extensions]

[I-D.ietf-spring-srv6-network-programming]

[I-D.xie-bier-ipv6-encapsulation]


7.2. Informative References


Authors' Addresses

Jingrong Xie
Huawei Technologies
Email: xiejingrong@huawei.com

Aijun Wang
China Telecom
Email: wangaj.bri@chinatelecom.cn
Use of BIER IPv6 Encapsulation (BIERv6) for Multicast VPN in IPv6 networks
draft-xie-bier-ipv6-mvpn-01

Abstract

This draft defines the procedures and messages for using Bit Index Explicit Replication (BIER) for Multicast VPN Services in IPv6 networks using the BIER IPv6 encapsulation. It provides a migration path for Multicast VPN service using BIER MPLS encapsulation in MPLS networks to multicast VPN service using BIER IPv6 encapsulation (BIERv6) in IPv6 networks.

Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119] and [RFC8174].

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 January 2, 2020.
1. Introduction

Bit Index Explicit Replication (BIER) [RFC8279] is an architecture that provides optimal multicast forwarding without requiring intermediate routers to maintain any per-flow state by using a multicast-specific BIER header. BIERv6 refers to the deployment of BIER in IPv6 networks using the BIER IPv6 encapsulation format defined in [I-D.xie-bier-ipv6-encapsulation].

[I-D.ietf-spring-srv6-network-programming] introduces the Network programming concepts in SRv6 networks and explains how the 128-bit IPv6 address can be used as SRv6 SID in the format LOC:FUNCT, where LOC part of the SID is routable, while FUNCT part of the SID is an opaque identification of a local function bound to the SID. It has
also defined some well known standard functions like End.DT4 - Endpoint with decaps and IPv4 table lookup for L3VPN (equivalent to per-VRF VPN label).

[I-D.dawra-bess-srv6-services] defines the TLVs to associate a function like End.DT4 with the L3VPN Unicast routes advertised via BGP. It also details how the functions of End.DT4, End.DT6, End.DT46 (End.DTx) can be used to identify a L3VPN/EVPN instead of using a VPN Label in MPLS-VPN [RFC4364] of the received data packet and thereby realize the L3VPN Services in the SRv6 Networks. However, it covers unicast services exclusively.

This document describes a method to realize MVPN services using BIER as a P-tunnel in the IPv6 Networks (BIERV6 Networks). It defines a method to use an SRv6 Service SID, called Src.DTx in this document, as source address of an IPv6 header, to identify the MVPN instance at the Egress PE. The LOC part and FUNCT part of this SRv6 Service SID represent the context and the upstream-assigned VPN Label respectively in MVPN scenario’s as defined in [RFC8556].

In particular, MVPN deployment in IPv6 networks relies on L3VPN deployment on IPv6 networks firstly, thus the c-multicast routing procedure like UMH Selection can be done. The L3VPN deployment in IPv6 networks can be referred to [I-D.dawra-bess-srv6-services].

GTM defined in [RFC7716] is also covered in this document, as GTM shares the same BGP-MVPN signaling, while providing an approach of Non-VPN multicast over a service provider core with various P-tunnel type. For the same reason of UMH selection, and the requirement of basic operation like ping (e.g, to the multicast source address), the Global IPv4/IPv6 over SRv6 Core as described in [I-D.dawra-bess-srv6-services] is also required.

2. Terminology

Readers of this document are assumed to be familiar with the terminology and concepts of the documents listed as Normative References. Additionally the following terms are used through out the document.

- BIERv6 - BIER in IPv6 networks using the BIERv6 encapsulation format defined in [I-D.xie-bier-ipv6-encapsulation].
- SRv6 - Segment Routing instantiated on the IPv6 dataplane as defined in [I-D.ietf-spring-srv6-network-programming].
- SRv6 SID - SRv6 Segment Identifier as defined in [I-D.ietf-spring-srv6-network-programming].
3. Use of PTA and Prefix-SID Attribute in x-PMSI A-D Routes

The BGP-MVPN I-PMSI A-D (Type 1) or S-PMSI A-D (Type 3) route (called x-PMSI A-D route in this document), advertised by Ingress PE carries the BIER (Type 11) PTA as specified in [RFC8556]. The BIER PTA carried in the x-PMSI A-D route is used for explicitly tracking the receiver-site PEs which are interested in a specific multicast flow. It includes three BIER-specific fields, Sub-domain-id, BFR-id, and BFR-prefix. For BIER P-tunnel using the BIERv6 encapsulation in IPv6 networks, the BFR-prefix field in the PTA MUST be set to the BFIR IPv6 prefix and the MPLS Label field in the PTA MUST set to 0. For MVPN over BIERv6, the Src.DTx IPv6 address of the BFIR is used to identify the VRF instead of an MPLS Label. The Src.DTx IPv6 Address (Src.DT6 or Src.DT4 or Src.DT46) MUST be carried within an SRv6 L3 Service TLV [I-D.dawra-bess-srv6-services] of BGP Prefix-SID attribute in the x-PMSI A-D route.

The Ingress PE encapsulates the c-multicast IP packet with BIERv6 header and the source address in the outer IPv6 header will be set to the Src.DTx IPv6 address advertised in the BGP-MVPN x-PMSI A-D routes. See section 3 of [I-D.xie-bier-ipv6-encapsulation] for the detailed packet format.

Egress PE (BFER) receiving the x-PMSI A-D routes with BIER PTA and SRv6 L3 Service TLV learns the Src.DTx IPv6 address and uses it to identify the VRF of the c-multicast packet.

When Egress PE receives a BIERv6 packet and the self bfr-id is set in the bit-string field of the BIERv6 header, it retrieves the Src.DTx IPv6 address from the source address of the IPv6 header to determine the VRF and the Address Family (AF) of the c-multicast data packet, and performs the MFIB lookup in the corresponding table.

4. MVPN over BIERv6 Core

[RFC8556] specifies the protocol and procedures to be followed by the Ingress and Egress PEs to use BIER as a P-tunnel for MVPN in MPLS networks. This section specifies the required changes and procedures.
in addition to support BIER as a P-tunnel in IPv6 networks using BIERv6.

In a IPv6 service provider network, many of the IP address fields used in the BGP-MVPN routes are IPv6 address as specified in [RFC6515]. These are listed below.

- "Originating Router's IP Address" in the NLRI of Type 1 or Type 3 BGP-MVPN route is an IPv6 address.
- "Network Address of Next Hop" field in the MP_REACH_NLRI attribute is an IPv6 address.
- Route Targets Extended Community (EC) used in C-multicast join (Type 6 or 7) route or Leaf A-D (Type 5) route is an IPv6 Address Specific Extended Community, where the Global Administrator field will be an IPv6 address identifies the Upstream PE or the UMH.
- "VRF Route Import Extended Community (EC)" carried by unicast VPN-IPv4 or VPN-IPv6 routes as [RFC6515] specifies, or SAFI 1, 2, or 4 unicast routes, or MVPN (SAFI 5) Source-Active routes as [RFC7716] specifies.

On the Ingress PE (BFIR), the BGP-MVPN x-PMSI A-D route is constructed as per the procedures specified in [RFC8556] and with the following specifications.

- MPLS Label field in the BIER PTA MUST be set to Zero.
- BFR-prefix field in the BIER PTA MUST be set to the Ingress PEs (BFIR) IPv6 BFR-Prefix Address. It does not need to be the same as the other IPv6 address of the x-PMSI AD route.
- Route MUST also carry an BGP Prefix SID attribute with an SRv6 L3 Service TLV carrying an Src.DTx IPv6 address uniquely identifying the MVPN instance.

If the MVPN is IPv4 MVPN, the Src.DTx can be either Src.DT4 or Src.DT46. If the MVPN is IPv6 MVPN, the Src.DTx can be either Src.DT6 or Src.DT46. The distribution of the x-PMSI A-D routes uses the Src.DTx according to the local configuration, and is independent to the use of End.DTx in VPN-IP unicast routes of this VPN. For example, one can use End.DT46 for VPNv4 and VPNv6 unicast routes, but use Src.DT4 for the MVPN routes for the same VPN. Another example, one can use End.DX for VPNv4 unicast routes, but use Src.DT46 for the MVPN routes for the same VPN.
BFIR MAY carry the BGP Prefix-SID attribute only in I-PMSI A-D route when I-PMSI A-D route is used, while other S-PMSI A-D routes do not carry the BGP Prefix-SID attribute.

BFIR MAY carry the BGP Prefix-SID attribute only in wildcard S-PMSI A-D routes when the "S-PMSI Only" mode as described in [RFC6625] is used, while other S-PMSI A-D routes do not carry the BGP Prefix-SID attribute.

On the Egress PE (BFER), the BGP-MVPN x-PMSI A-D route is processed as per the procedures specified in [RFC8556] and with the following specifications:

- The MPLS Label field in the BIER PTA of the BGP-MVPN x-PMSI A-D route MUST be ignored and MUST not be used for the identification of the VRF.
- The BGP-MVPN x-PMSI A-D route MUST be dropped if the BFR-prefix field in the BIER PTA is not an IPv6 address.
- The BGP-MVPN x-PMSI A-D route MUST be dropped if it does not carry a Src.DTx IPv6 address in the SRv6 L3 Service TLV in BGP Prefix SID attribute.
- Leaf A-D route originated by the Egress PE (BFER) MUST carry the BIER PTA with the BFR-prefix field set to the BFER IPv6 BFR-prefix.

Valid BGP-MVPN x-PMSI A-D route received by an Egress PE (BFER) is stored locally, and the Src.DTx IPv6 Address carried in the SRv6 L3 service TLV is used to identify the VRF of a c-multicast data packet. This may be populated into forwarding table only when there is c-multicast flow state with UMH of the specific BFIR this Src.DTx located in.

If more than one x-PMSI A-D routes belonging to the same VRF has different Src.DTx value, the processing is determined by the local policy of the BFER.

If more than one x-PMSI A-D routes belonging to different VRF has the same Src.DTx value, the BFER must log an error, and a BIERv6 packet with this Src.DTx as the IPv6 source address MUST be dropped.

The BGP Prefix-SID attribute (which may include the Src.DTx in SRv6 L3 Service TLV) MUST NOT be carried in Leaf A-D route upon sending, and MUST be ignored upon reception.
5. GTM over BIERv6 Core

As specified in [RFC7716], Global Table Multicast (GTM) uses the same Subsequent Address Family Identifier (SAFI) value, the same Network Layer Reachability Information (NLRI) format, and the same procedures of MVPN with only a few adaptions. It support for both IPv4 and IPv6 multicast flows over either an IPv4 or IPv6 SP infrastructure. GTM over BIERv6 core is obviously a case of IPv4/IPv6 multicast over an IPv6 SP infrastructure with BIERv6 data-plane.

The BIER (Type 11) PTA attribute and the BGP Prefix-SID attribute are carried in the x-PMSI A-D route in GTM cases. When the a BGP-MVPN x-PMSI A-D route is received by Egress PE, it is stored locally, and the Src.DTx IPv6 Address of the Ingress PE in the route is used to determine the VRF of a packet, which is the 'public' VRF in the case of GTM.

There are some other attributes listed below for GTM over a BIERv6 core:

- Route Distinguishers - the RD field of a BGP-MVPN route’s NLRI MUST be set to zero (i.e., to 64 bits of zero) to represent a Non-VPN GTM. See section 2.2 of [RFC7716].

- Route Targets Extended Community (EC) - The RT EC carried by the BGP-MVPN C-multicast (Type 6 or 7) route or Leaf A-D (Type 4) route MUST be an IPv6-address-specific Extended Community (EC). The Global Administrator field identifies the Upstream PE or the UMH, and the Local Administrator field MUST always be set to zero in GTM case.

- VRF Route Import Extended Community (EC) - The VRF Route Import EC used in BIERv6 core MUST be an IPv6-address-specific EC if used, either used in UMH-eligible unicast routes having a SAFI of 1, 2, or 4, or used in the MVPN (SAFI of 5) Source Active A-D route.

GTM IPv4 multicast over an BIERv6 core may be considered an alternative to support IPv4 IPTV content delivery during transition to IPv6 period comparing to [RFC8114]. They both use IPv4-in-IPv6 encapsulation, while BIERv6 uses an additional BIER header within an IPv6 Extension header to support stateless core.

6. Data Plane
6.1. Encapsulation of Multicast Traffic

BIER IPv6 encapsulation (BIERV6) [I-D.xie-bier-ipv6-encapsulation] is used for forwarding c-multicast traffic through an IPv6 core. The following diagram shows the progression of an MVPN c-multicast packet as it enters and leaves the intra-AS service-provider network.

```
+---------------+    +---------------+
| P-IPv6 Header |    | P-IPv6 Header |
| (SA=Src.DTx   |    | (SA=Src.DTx   |
| DA=End.BIER) |    | DA=End.BIER) |
+---------------+    +---------------+
| P-IPv6 ExtHdr |    | P-IPv6 ExtHdr |
| (BIER header) |    | (BIER header) |
+--------------++=+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++++==+++++---Figure 1: BIERv6 MVPN/GTM Intra-AS
```

In case of inter-AS scenario, BIERv6 packets may travel through unicast to a Boarder Router (BR), and then replicate in a single intra-AS BIERv6 domain. How such non-segmented BIERv6 scenario can be supported is outside the scope of this document.

How segmented MVPN, for example, between BIERv6 and BIERv6, or between BIERv6 and Ingress Replication(IR) in Non-MPLS IPv6 networks, is outside the scope of this document.

The Src.DTx SHOULD support as destination address of an ICMPv6 packet. The following is an example pseudo-code of the Src.DTx function as destination address:

1. IF Last_NH = ICMPv6 ;;Ref1
2. Send to CPU.
3. ELSE
4. Drop the packet.

Ref1: ICMPv6 packet using Src.DT4, Src.DT6 or Src.DT46 as destination address.
6.2. MTU

Each BFIR is expected to know the Maximum Transmission Unit (MTU) of the BIER domain. This may be known by provisioning, or by method specified in [draft-ietf-bier-mtud]. The section 3 of [RFC8296] applies.

6.3. TTL

The ingress PE (BFIR) should not copy the Time to Live (TTL) field from the payload IP header received from a CE router to the delivery IP header. Setting the TTL of the delivery IP header is determined by the local policy of the ingress PE (BFIR) router per section 3 of [RFC8296].

7. Security Considerations

The security considerations SEC-1, SEC-2, SEC-3 defined in [I-D.ietf-spring-srv6-network-programming] apply equally to this document.

8. IANA Considerations

Allocation is expected from IANA for the following Src.DT6 functions codepoints from the "SRv6 Endpoint Behaviors" sub-registry.

Values 68, 69, 70 is suggested for Src.DT6, Src.DT4, Src.DT46 respectively.

+-------+--------+--------------------------+------------+
| Value | Hex    |    Endpoint function     | Reference  |
+-------+--------+--------------------------+------------+
| TBD   | TBD    |    Src.DT6               | This draft |
+-------+--------+--------------------------+------------+
| TBD   | TBD    |    Src.DT4               | This draft |
+-------+--------+--------------------------+------------+
| TBD   | TBD    |    Src.DT46              | This draft |
+-------+--------+--------------------------+------------+

Src.DT6  Source address indicating decapsulation and IPv6 table lookup e.g. IPv6-MVPN (equivalent to per-VRF VPN label in RFC8556)
Src.DT4  Source address indicating decapsulation and IPv4 table lookup e.g. IPv4-MVPN (equivalent to per-VRF VPN label in RFC8556)
Src.DT46 Source address indicating decapsulation and IP table lookup e.g. IP-MVPN (equivalent to per-VRF VPN label)
9. Acknowledgements

TBD.

10. References

10.1. Normative References

[I-D.dawra-bess-srv6-services]

[I-D.ietf-spring-srv6-network-programming]

[I-D.xie-bier-ipv6-encapsulation]


10.2. Informative References


Authors’ Addresses

Jingrong Xie
Huawei Technologies
Email: xiejingrong@huawei.com

Mike McBride
Futurewei
Email: mmcbride7@gmail.com

Senthil Dhanaraj
Huawei Technologies
Email: senthil.dhanaraj@huawei.com
Liang Geng
China Mobile

Email: gengliang@chinamobile.com
Abstract

BIER is a new architecture for the forwarding of multicast data packets. This document defines native IPv6 encapsulation for BIER hop-by-hop forwarding or BIERin6 for short.

Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC2119.

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 January 9, 2020.
1. Introduction

BIER [RFC8279] is a new architecture for the forwarding of multicast data packets. It provides optimal forwarding through a "multicast domain" and it does not necessarily precondition construction of a multicast distribution tree, nor does it require intermediate nodes to maintain any per-flow state.

This document specifies non-MPLS BIER forwarding in an IPv6 [RFC8200] environment, referred to as BIERin6, using non-MPLS BIER encapsulation specified in [RFC8296].

MPLS BIER forwarding in IPv6 is outside the scope of this document.

This document uses terminology defined in [RFC8279] and [RFC8296].
Internet-Draft                  BIERv6                        July 2019

[RFC8296] defines the BIER encapsulation format in MPLS and non-MPLS
environment. In case of non-MPLS environment, a BIER packet is the
payload of an "outer" encapsulation, which has a "next protocol"
codepoint that is set to a value that means "non-MPLS BIER".

That can be used as is in a pure IPv6 non-mpls environment. Between
two directly connected BFRs, a BIER header could directly follow link
layer header, e.g., an Ethernet header (with the Ethertype set to
0xAB37). If a BFR needs to tunnel BIER packets to another BFR, e.g.
per [RFC8279] Section 6.9, IPv6 encapsulation can be used, with the
destination address being the downstream BFR and the Next Header
field set to a to-be-assigned value for "non-MPLS BIER".

The IPv6 encapsulation could be used even between two directly
connected BFRs in the following two cases:

  o An operator mandates all traffic to be carried in IPv6.

  o A BFR does not have BIER support in its "fast forwarding path" and
     relies on "slow/software forwarding path", e.g. in environments
     like [RFC7368] where high throughput multicast forwarding
     performance is not critical.

2. IPv6 Header

Whenever IPv6 encapsulation is used for BIER forwarding, The Next
Header field in the IPv6 Header (if there are no extension headers),
or the Next Header field in the last extension header is set to TBD,
indicating that the payload is a BIER packet.

If the neighbor is directly connected, The destination address in
IPv6 header SHOULD be the neighbor’s link-local address on this
router’s outgoing interface, the source destination address SHOULD be
this router’s link-local address on the outgoing interface, and the
IPv6 TTL MUST be set to 1. Otherwise, the destination address SHOULD
be the BIER prefix of the BFR neighbor, the source address SHOULD be
this router’s BIER prefix, and the TTL MUST be large enough to get
the packet to the BFR neighbor.

The Flow-ID in the IPv6 packet SHOULD be copied from the entropy
field in the BIER encapsulation.

2.1. IPv6 Options Considerations

RFC 8200 section 4, defines the IPv6 extension headers. Currently
there are two defined extension headers, Hop-by-Hop and Destination
options header, which can carry a variable number of options. These
extension headers are inserted by the source node.
For directly connected BIER routers, IPv6 Hop-by-Hop or Destination options are irrelevant and SHOULD NOT be inserted by BFIR on the BIERin6 packet. In this case IPv6 header, Next Header field should be set to TBD. Any IPv6 packet arriving on BFRs and BFERs, with multiple extension header where the last extension header has a Next Header field set to TBD, SHOULD be discard and the node should transmit an ICMP Parameter Problem message to the source of the packet (BFIR) with an ICMP code value of TBD10 ('invalid options for BIERin6').

This also indicates that for disjoint BIER routers using IPv6 encapsulation, there SHOULD NOT be any IPv6 Hop-by-Hop or Destination options be present in a BIERin6 packet. In this case, if additional traffic engineering is required, IPv6 tunneling (i.e. BIERin6 over SRv6) can be implemented.

3. BIER Header

The BIER header MUST be encoded per Section 2.2 of [RFC8296].

The BIFT-id is either encoded per [I-D.ietf-bier-non-mpls-bift-encoding] or per advertised by BFRs, as specified in [I-D.dhanaraj-bier-lsr-ethernet-extensions].

4. IPv6 Encapsulation Advertisement

When IPv6 encapsulation is not required between directly connected BFRs, no signaling in addition to that specified in [I-D.dhanaraj-bier-lsr-ethernet-extensions] is needed.

Otherwise, a node that requires IPv6 encapsulation MUST advertise the BIER IPv6 transportation sub-TLV/sub-sub-TLV according to local configuration or policy in the BIER domain to request other BFRs to always use IPv6 encapsulation.

In presence of multiple encapsulation possibilities hop-by-hop it is a matter of local policy which encapsulation is imposed and the receiving router MUST accept all encapsulations that it advertised.

4.1. Format

The BIER IPv6 transportation is a new sub-TLV of BIER defined in OSPF [RFC8444], and a new sub-sub-TLV of BIER Info sub-TLV defined in ISIS [RFC8401].
4.2. Inter-area prefix redistribution

When BFR-prefixes are advertised across IGP areas per
[I-D.dhanaraj-bier-lsr-ethernet-extensions] or redistributed across
protocol boundaries per [I-D.zwzw-bier-prefix-redistribute], the BIER
IPv6 transportation sub-TLV or sub-sub-TLV MAY be re-advertised/re-
distributed as well.

5. IANA Considerations

IANA is requested to assign a new "BIER" type for "Next Header" in
the "Assigned Internet Protocol Numbers" registry.

IANA is requested to assign a new "BIERin6" type for "invalid
options" in the "ICMP code value" registry.

IANA is requested to assign a new "BIER IPv6 transportation Sub-TLV"
type in the "OSPFv2 Extended Prefix TLV Sub-TLVs" Registry.

IANA is requested to set up a new "BIER IPv6 transportation Sub-sub-
TLV" type in the "IS-IS BIER Info sub-TLV" Registry.

6. Security Considerations

General IPv6 and BIER security considerations apply.

7. Acknowledgement

The authors would like to thank Jeffrey Zhang for his review and
valuable contributions.

8. References
8.1. Normative References


8.2. Informative References


[I-D.ietf-bier-non-mpls-bift-encoding]

[I-D.zhang-bier-babel-extensions]

[I-D.zwzw-bier-prefix-redistribute]


Authors’ Addresses

Zheng(Sandy) Zhang
ZTE Corporation
EMail: zzhang_ietf@hotmail.com

Tony Przygienda
Juniper Networks, Inc.
EMail: prz@juniper.net

IJsbrand Wijnands
Cisco Systems
EMail: ice@cisco.com

Hooman Bidgoli
Nokia
EMail: hooman.bidgoli@nokia.com
Abstract

This document describes the multicast source protection functions in the Bit Index Explicit Replication BIER domain.

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 January 8, 2020.

Copyright Notice

Copyright (c) 2019 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

Bit Index Explicit Replication (BIER) [RFC8279] is an architecture that provides multicast forwarding through a "BIER domain" without requiring intermediate routers to maintain any multicast related per-flow state. BIER also does not require any explicit tree-building protocol for its operation. A multicast data packet enters a BIER domain at a "Bit-Forwarding Ingress Router" (BFIR), and leaves the BIER domain at one or more "Bit-Forwarding Egress Routers" (BFERs).

To protect the source node it may be transmitting to two or more BFIRs. Based on local policies, BFERs may elect to use the same BFIR or different BFIRs as the source of the multicast flow. The BFIR and the path in use are referred to as working while all alternative available BFIRs and paths that can be used to receive the same multicast flow are referred to as protection. For a BFER, when either the working BFIR or the working path fail, the BFER can select one of protection BFIRs to get the multicast flow. The shorter the detection time is, the faster the flow recovers.

This document discusses the functions that can be used in failure detection for multicast source protection.

2. Multicast Source Protection

Two BFIRs independently advertise the source of the multicast flow to BFERs. The precise type of advertisement depends on the overlay protocol being used, e.g., MLD, MVPN, EVVPN. BFER selects one BFIR as the UMH (Upstream Multicast Hop). Different BFERs may select the same BFIR or different BFIRs according to the local policy.
For example, a multicast source S1 is connected to BFIR1 and BFIR2. BFIR1 and BFIR2 advertise the source information to BFERs. It is assumed that BFER1, BFER2, and BFER3 all choose BFIR1 as the UMH. BFERs signal to BFIR1 to get the multicast flow from S1.

In case BFIR1 fails, or the path from BFIR1 to BFER1 is broken, BFER1 should select BFIR2 as the UMH. But if the timeout period is too long, the multicast flow will be significantly affected.

2.1. BIER Ping

[I-D.ietf-bier-ping] describes the mechanism and basic BIER OAM packet format that can be used to perform failure detection and isolation on BIER data plane without any dependency on other layers like the IP layer.

In the example of Figure 1, BFER can monitor the status of BFIR and the path status between BFER and BFIR. BFER1 sends the BIER Ping packet to BFIR1. If BFER1 does not receive responses from BFIR1 in a period of time, BFER1 will treat BFIR1 as a failed UMH, and BFER1 will select BFIR2 as the UMH and signal to BFIR2 to get multicast flow.

In this example, BFER1, BFER2, and BFER3 send BIER ping packet to BFIR1 separately. The timeout period MAY be set to a different values depending on the local performance requirement on each BFER.

In general case of more complex BIER topology, it cannot be guaranteed that the path used from BFIR1 to BFER1 is the same as in the reverse direction, i.e., from BFER1 to BFIR1. If that is not guaranteed and the paths are not co-routed, then this method may produce false results, both false negative and false positive. The
former is when ping fails while the multicast path and flow are OK. The latter is when the multicast path has defect but ping works. Thus, to improve consistency of this method of detecting a failure in multicast flow transport, the path that the echo request from BFER1 traverses to BFIR1 must be co-routed with the path that the monitored multicast flow traverses through the BIER domain from BFIR1 to BFER1.

2.2. BIER BFD

[I-D.hu-bier-bfd] describes the application of P2MP BFD in BIER network. And it describes the procedures for using such mode of BFD protocol to verify multipoint or multicast connectivity between a sender (BFIR) and one or more receivers (BFERs).

In the same example, BFIR1 sends the BIER Echo request packet to BFERs to bootstrap a p2mp BFD session. After BFER1, BFER2 and BFER3 receive the Echo request packet with BFD Discriminator and the Target SI-Bitstring TLVs, BFERs creates the BFD session of type MultipointTail [RFC8562] to monitor the status of BFIR1 and the working path. If BFERs have not received BFD packet from BFIR1 for the Detection Time [RFC8562], BFIR1 will treat BFIR1 as a failed UMH, and signal to BFIR2 to get the multicast flow.

The timeout period on each BFER MAY be set to different value depending on the local performance requirement on each BFER. BFER monitors BFIR separately and selects its UMH independently from selections reached by other BFERs.

3. Security Considerations

Security considerations discussed in [RFC8279], [RFC8562], [I-D.ietf-bier-ping] and [I-D.hu-bier-bfd] apply to this document.

4. Normative References

[I-D.hu-bier-bfd]

[I-D.ietf-bier-ping]


Authors’ Addresses

Zheng Zhang
ZTE Corporation

Email: zzhang_ietf@hotmail.com

Greg Mirsky
ZTE Corporation

Email: gregimirsky@gmail.com

Quan Xiong
ZTE Corporation

Email: xiong.quan@zte.com.cn
Tethering A BIER Router To A BIER-incapable Router
draft-zzhang-bier-tether-02

Abstract

This document specifies optional procedures to optimize the handling of Bit Index Explicit Replication (BIER) incapable routers, by tethering a BIER router to a BIER incapable router.

Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC2119.

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 January 7, 2020.

Copyright Notice

Copyright (c) 2019 IETF Trust and the persons identified as the document authors. All rights reserved.
1. Terminologies

Familiarity with BIER architecture, protocols and procedures is assumed. Some terminologies are listed below for convenience.

[To be added].

2. Introduction

Consider the following scenario where router X does not support BIER.

```
------- BFR2 ------- BFER2
/
BFER1 --- BFR1 ---- X ------- BFR3 ------- BFER3
```
```
\       
------- BFrn ------- BFERn
```

For BFR1 to forward BIER traffic towards BFR2...BFRn, it needs to tunnel individual copies through X. This degrades to "ingress" replication to those BFRs. If X’s connections to BFRs are long...
distance or bandwidth limited, and $n$ is large, it becomes very inefficient.

A solution to the inefficient tunneling from BFRs is to tether a BFR$_x$ to X:

```
    ------ BFR2 ------ BFER2
     /        \
BFER1 --- BFR1 ---- X -------- BFR3 ------- BFER3
    /   \          \       \       \       \    
   BFR$_x$ ------ BFR$_n$ ------- BFER$_n$
```

Instead of BFR$_1$ tunneling to BFR$_2$, ..., BFR$_n$ directly, BFR$_1$ will get BIER packets to BFR$_x$, who will then tunnel to BFR$_2$, ..., BFR$_n$. There could be fat and local pipes between the tethered BFR$_x$ and X, so ingress replication from BFR$_x$ is acceptable.

For BFR$_1$ to tunnel BIER packets to BFR$_x$, the BFR$_1$-BFR$_x$ tunnel need to be announced in IGP as a forwarding adjacency so that BFR$_x$ will appear on the SPF tree. This need to happen in a BIER specific topology so that unicast traffic would not be tunneled to BFR$_x$. Obviously this is operationally cumbersome.

Section 6.9 of BIER architecture specification [RFC8279] describes a method that tunnels BIER packets through incapable routers without the need to announce tunnels. However that does not work here, because BFR$_x$ will not appear on the SPF tree of BFR$_1$.

There is a simple solution to the problem though. Even though X does not support BIER forwarding, it could advertise BIER information as if it supported BIER so BFRs will send BIER packets to it. The BIER packets have a BIER label in front of the BIER header and X will use the BIER label to label switch to BFR$_x$, who will in turn do BIER forwarding to other BFRs but via tunneling as described in section 6.9 of BIER architecture spec.

Even though X advertises as if it supported BIER, BFR$_x$ needs to know that X does not really support BIER so it will tunnel to other BFRs through X. The knowledge is through static provisioning or through additional signaling. In the latter case, X could advertise that BFR$_x$ is its helper node, so that other BFRs could optionally use the Section 6.9 method to tunnel to BFR$_x$, instead of sending native BIER packets to X and rely on X label switching to BFR$_x$. This also allows it to work in the non-MPLS case.
Alternatively, instead of for X to advertise that it supports BIER but relies on helper BFRx, BFRx could advertise that it is X’s helper and other BFRs will use BFRx (instead of X’s children on the SPF tree) to replace X during its post-SPF processing as described in section 6.9 of BIER architecture spec. That way, X does not need any special knowledge, provisioning or procedure.

The two options both have pros and cons - the first option only needs X and BFRx to support the new procedure while the second option does not require anything to be done to the BIER incapable X.

BFRx could also be connected to other routers in the network so that it could send BIER packets through other routers as well, not necessarily tunneling through X. To prevent routing loops, smallest metric, which is 1, must be announced for links between X and BFRx in both directions.

3. Additional Considerations

While the example shows a local connection between BFRx and X, it does not have to be like that. As long as packets can arrive at BFRx without requiring X to do BIER forwarding, it should work. For example, X could label switch incoming BIER packets through a multi-hop tunnel to BFRx, or other BFRs could tunnel BIER packets to BFRx based on X’s advertisement that BFRx is its helper. However, BFRx must make sure that if X appears in its SPF paths to some BFERs, then it must tunnel BIER packets for those BFERs directly to X’s BFR children on BFRx’s SPF tree.

Additionally, the helper BFRx can be a transit helper, i.e., it has other connections (instead of being a stub helper that is only connected to X), as long as BFRx won’t send BIER packets tunneled to it back towards the tunnel ingress:

```
      ----- BFR2 ----- BFER2
      /                  |
BFER1 --- BFR1 ---- X ----- BFR3 ----- BFER3
      |                  |
      BFRx ---- BFR4 ---- BFER4
      \                  |
      ----- BFR5 ----- BFER5
```

In the following example, there is a connection between BFR1 and BFRx. If the link metrics are all 1 on the three sides of BFR1-X-BFRx triangle, loop won’t happen but if the BFRx-X metric is 3 while other two sides of the triangle has metric 1 then BFRx will
send BIER packets tunneled to it from BFR1 back to BFR1, causing a loop.

```
------- BFR2 ------- BFER2
/     
BFER1 --- BFR1 ---- X ------- BFR3 ------- BFER3
\    /     
  \  /      
    \   
      \  
BFRx    ------ BFRn ------- BFERn
```

This can easily be prevented if BFR1 does an SPF calculation with the helper BFRx as the root. For any BFERn reached via X from BFR1, if BFRx’s SPF path to BFERn includes BFR1 then BFR1 must not use the helper. Instead, BFR1 must directly tunnel packets for BFERn to X’s BFR (grand-)child on BFR1’s SPF path to BFERn, per section 6.9 of [RFC8279].

Notice that this SPF calculation on BFR1 with BFRx as the root is no different from the SPF done for a neighbor as part of LFA calculation. In fact, BFR1 tunneling packets to X’s helper is no different from sending packets to a LFA backup.

Also notice that, instead of a dedicated helper BFRx, any one or multiple ones of BFR2..N can also be the helper (as long as the connection between that BFR and X has enough bandwidth for replication to multiple helpers through X). To allow multiple helpers to help the same non-BFR, the "I am X’s helper" advertisement carries a priority. BFR1 will choose the helper advertising the highest priority among those satisfying the loop-free condition described above. When there are multiple helpers advertising the same priority and satisfying the loop-free condition, any one or multiple ones could be used solely at the discretion of BFR1. However, if multiple ones are used, it means that multiple copies may be tunneled through X.

The following situation where a helper BFRxy helps two different non-BFRs X and Y also works. It’s just a special situation of a transit helper.
4. Specification

The procedures in this document apply when a BFRx is tethered to a BIER incapable router X as X’s helper for BIER forwarding.

BFRx MUST not send BIER packets natively to X even if X advertises BIER information. BFRx knows that X does not really support BIER either from provisioning or from the BIER Helper Node sub-sub-TLV advertised by X.

Procedures for BGP signaling is described in Section 4.3.

Either of the following two methods may be used for ISIS [RFC8401] and OSPF [RFC8444]. The sub-sub-TLVs for both methods have the same format: the value is BIER prefix of the helper/helped node followed by a one-octet priority field, and one-octet reserved field. The length is 6 for IPv4 and 18 for IPv6 respectively.

4.1. Advertising from Helped Node

For non-MPLS encapsulation, X MUST advertise a BIER Helper Node sub-sub-TLV that specifies the BIER prefix of the helper BFRx. Other BFRs MUST use the Section 6.9 procedure modified as following: X is treated as BIER incapable (because of the BIER Helper Node sub-sub-TLV), and is replaced with the BFRx (instead of X’s children on the SPF tree) during the post-SPF processing.

This requires other BFRs to recognize the BIER Helper Node sub-sub-TLV. The same procedure MAY be used for MPLS encapsulation, though with the following alternative for MPLS encapsulation, tethering is transparent to other BFRs (except the helper node BFRx) - they do not need to be aware that X does not support BIER at all.
For MPLS encapsulation, X MAY advertises BIER information as if it supported BIER forwarding, including the MPLS Encapsulation sub-sub-TLV with a label range. X MUST set up its forwarding state such that incoming packets with a BIER label in its advertised label range are label switched to BFRx, either over a direct link or through a tunnel. The incoming label is swapped to a BIER label advertised by BFRx for the <sub-domain, bsl, set> that the incoming label corresponds to.

Notice that both methods can be used for MPLS encapsulation at the same time. In that case another BFR may send BIER packets to X natively, or tunnel to BFRx directly.

4.2. Advertising from Helper Node

With this method, the helper node (BFRx) MUST advertise a BIER Helped Node sub-sub-TLV that specifies the BIER incapable node (X) that this node helps. When other BFRs follow the post-SPF processing procedures as specified in section 6.9 of the BIER architecture spec [RFC8279], they replace the helped node on the SPF tree with the helper node (instead of the children of the helped node).

4.3. Procedures for BGP Signaling

Suppose that the BIER domain uses BGP signaling [I-D.ietf-bier-idr-extensions] instead of IGP. BFR1..N advertises BIER prefixes that are reachable through them, with BIER Path Attributes (BPA) attached. There are three situations regarding X’s involvement:

(1) X does not participate in BGP peering at all
(2) X re-advertises the BIER prefixes but does not do next-hop-self
(3) X re-advertises the BIER prefixes and does next-hop-self

With (1) and (2), the BFR1..N will tunnel BIER packets directly to each other. It works but not efficiently as explained earlier. With (3), BIER forwarding will not work, because BFR1..N would try to send BIER packets to X though X does not advertise any BIER information. If Tunnel Encapsulation Attribute (TEA) [I-D.ietf-idr-tunnel-encaps] is used as specified in [I-D.zzhang-bier-multicast-as-a-service] with (3), then it becomes similar to (2) - works but still not efficiently.

To make tethering work well with BGP signaling, the following can be done:
o Configure a BGP session between X and its helper BFRx. X re-advertises BIER prefixes (with BPA) to BFRx without changing the tunnel destination address in the TEA.

o BFRx advertises its own BIER prefix with BPA to X, and sets the tunnel destination address in the TEA to itself. X then re-advertises BFRx’s BIER prefix to BFR1..N, without changing the tunnel destination address in the TEA.

o For BIER prefixes (with BIER Path Attribute) that X re-advertises to other BFRs, the tunnel destination in the TEA is changed to the helper BFRx.

With the above, BFR1..N will tunnel BIER packets to BFRx (following the tunnel destination address in the TEA), who will then tunnel packets to other BFRs (again following the tunnel destination address in the TEA). Notice that what X does is not specific to BIER at all.

5. Security Considerations

This specification does not introduce additional security concerns beyond those already discussed in BIER architecture and OSPF/ISIS/BGP extensions for BIER signaling.

6. IANA Considerations

This document requests two new sub-sub-TLV type values from the "Sub-sub-TLVs for BIER Info Sub-TLV" registry in the "IS-IS TLV Codepoints" registry:

<table>
<thead>
<tr>
<th>Type</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>TBD1</td>
<td>BIER Helper Node</td>
</tr>
<tr>
<td>TBD2</td>
<td>BIER Helped Node</td>
</tr>
</tbody>
</table>

This document also requests two new sub-TLV type values from the OSPFv2 Extended Prefix TLV Sub-TLV registry:

<table>
<thead>
<tr>
<th>Type</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>TBD3</td>
<td>BIER Helper Node</td>
</tr>
<tr>
<td>TBD4</td>
<td>BIER Helped Node</td>
</tr>
</tbody>
</table>

7. Acknowledgements

The author wants to thank Eric Rosen and Antonie Przygienda for their review, comments and suggestions.
8. Normative References

[I-D.ietf-bier-idr-extensions]

[I-D.ietf-idr-tunnel-encaps]

[I-D.zzhang-bier-multicast-as-a-service]


Authors’ Addresses

Zhaohui Zhang
Juniper Networks

EMail: zzhang@juniper.net