

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: January 5, 2020

L. Geng
China Mobile
J. Xie
Huawei Technologies
M. McBride
Futurewei
G. Yan
Huawei Technologies
July 4, 2019

Inter-Domain Multicast Deployment using BIERv6
draft-geng-bier-ipv6-inter-domain-00

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.

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	2
2. Terminology	2
3. Inter-domain Multicast Overview	3
4. Inter-domain Multicast Deployment using BIERv6	3
4.1. Hierarchical Multicast	3
4.2. Peering Multicast	6
5. Security Considerations	8
6. IANA Considerations	8
7. Acknowledgements	8
8. References	8
8.1. Normative References	8
8.2. Informative References	9
Authors' Addresses	10

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.

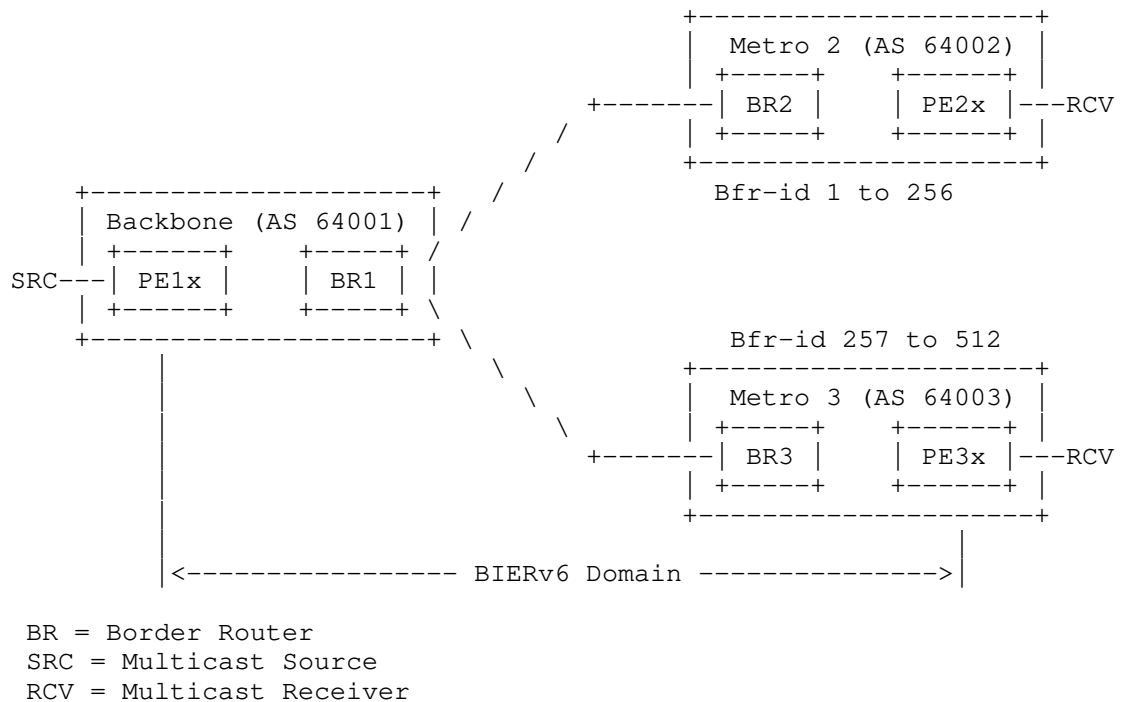


Figure 1: Inter-Domain Hierarchical Multicast

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.

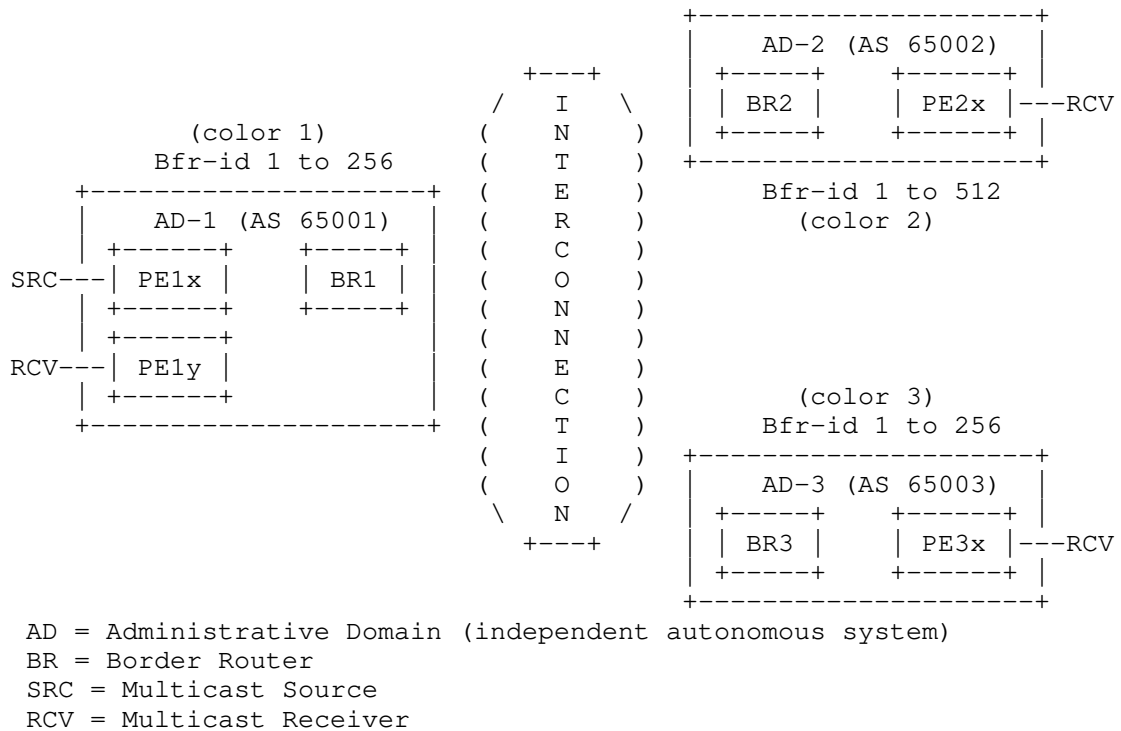


Figure 2: Inter-Domain Peering Multicast

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:

```
(VRF<X>, S<S1>, G<G1>)  
  (Color<1>, SD<6>, BSL<256>, SI<0>, BitString<0001>) ;;Ref1  
  (Color<2>, SD<6>, BSL<256>, SI<0>, BitString<0001>) ;;Ref2  
  (Color<2>, SD<6>, BSL<256>, SI<1>, BitString<0001>) ;;Ref3  
  (Color<3>, SD<6>, BSL<256>, SI<0>, BitString<0001>) ;;Ref4
```

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]  
  McBride, M., Xie, J., Dhanaraj, S., and R. Asati, "BIER  
  IPv6 Requirements", draft-ietf-bier-ipv6-requirements-00  
  (work in progress), May 2019.
```


- [I-D.ietf-bier-non-mpls-bift-encoding]
Wijnands, I., Xu, X., and H. Bidgoli, "An Optional Encoding of the BIFT-id Field in the non-MPLS BIER Encapsulation", draft-ietf-bier-non-mpls-bift-encoding-01 (work in progress), October 2018.
- [I-D.xie-bier-ipv6-encapsulation]
Xie, J., Geng, L., McBride, M., Dhanaraj, S., Yan, G., and Y. Xia, "Encapsulation for BIER in Non-MPLS IPv6 Networks", draft-xie-bier-ipv6-encapsulation-01 (work in progress), June 2019.
- [I-D.xie-bier-ipv6-mvpn]
Xie, J., McBride, M., Dhanaraj, S., and L. Geng, "Use of BIER IPv6 Encapsulation (BIERv6) for Multicast VPN in IPv6 networks", draft-xie-bier-ipv6-mvpn-01 (work in progress), July 2019.
- [RFC5512] Mohapatra, P. and E. Rosen, "The BGP Encapsulation Subsequent Address Family Identifier (SAFI) and the BGP Tunnel Encapsulation Attribute", RFC 5512, DOI 10.17487/RFC5512, April 2009, <<https://www.rfc-editor.org/info/rfc5512>>.
- [RFC6770] Bertrand, G., Ed., Stephan, E., Burbridge, T., Eardley, P., Ma, K., and G. Watson, "Use Cases for Content Delivery Network Interconnection", RFC 6770, DOI 10.17487/RFC6770, November 2012, <<https://www.rfc-editor.org/info/rfc6770>>.
- [RFC8279] Wijnands, IJ., Ed., Rosen, E., Ed., Dolganow, A., Przygienda, T., and S. Aldrin, "Multicast Using Bit Index Explicit Replication (BIER)", RFC 8279, DOI 10.17487/RFC8279, November 2017, <<https://www.rfc-editor.org/info/rfc8279>>.
- [RFC8313] Tarapore, P., Ed., Sayko, R., Shepherd, G., Eckert, T., Ed., and R. Krishnan, "Use of Multicast across Inter-domain Peering Points", BCP 213, RFC 8313, DOI 10.17487/RFC8313, January 2018, <<https://www.rfc-editor.org/info/rfc8313>>.

8.2. Informative References

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

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

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

BIER Workgroup
Internet Draft
Intended status: Standard Track

H. Bidgoli
J. Kotalwar
Nokia
Z.Zhang
Juniper Networks
Eddie Leyton
Vrizon
Mankamana Mishra
I. Wijanands
Cisco System, Inc.

Expires: May 6, 2020

November 3, 2019

M-LDP Signaling Through BIER Core
draft-hb-bier-mlbp-signaling-over-bier-01

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/lid-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 tLDP session creation	5
3.1.1.1. ECMP Method on IBBR	6
3.2. Egress BBR procedure method	6
3.2.1. IBBR procedure upon arriving upstream assigned label	6
4. Datapath Forwarding	7
4.1. Datapath traffic flow	7
5. Recursive FEC	7
6. IANA Considerations	7
7. Security Considerations	7
8. References	7
8.1. Normative References	7
8.2. Informative References	8
7. Acknowledgments	8

Authors' Addresses	8
------------------------------	---

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 the mLDP datapath labels through 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

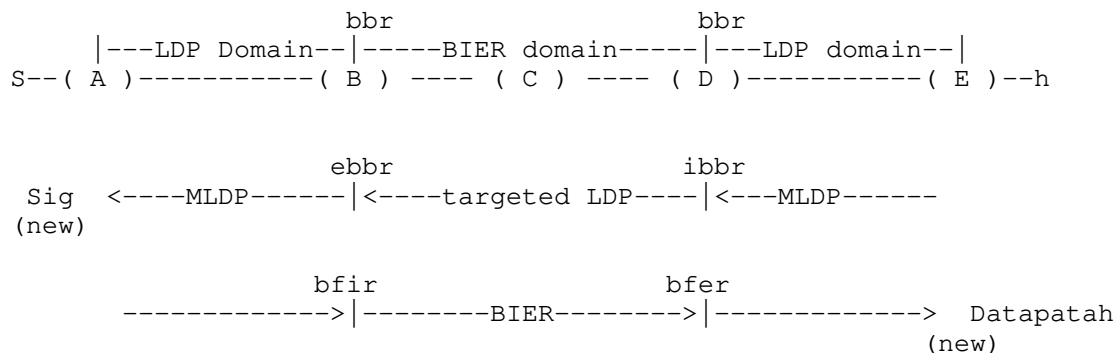


Figure 1: bier boundary router

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 documents 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 establish a 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. As an example when a mLDP FEC arrives on the IBBR, it can automatically start a tLDP Session with the EBBR. In this case both IBBR and EBBR should be in auto-discovery mode and react to the arriving FEC or tLDP Signaling packets (i.e. targeted hellos, keep-alives etc...).

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.

3.1.1. ECMP Method on IBBR

If IBBR finds multiple equal cost EBBRs on the path to the Root, it can use a vendor specific algorithm to choose between the EBBRs. These algorithms are beyond the scope of this draft. As an example the IBBR can use the smallest EBBR IP address to establish its mLDP signaling to.

3.2. Egress BBR 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 generated from IBBR. It should assign a unique "upstream assigned label" for each arriving FEC generated by IBBRs.

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)

The EBBR will also generate a new label and FEC toward the ROOT on the mLDP domain. The EBBR Should stitch this generate label with the "upstream assigned label" to complete the p2MP or MP2MP LSP. This stitch point should be stored on the datapath (ILM) table for packet forwarding.

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.

3.2.1. IBBR procedure upon arriving upstream assigned label

Upon receiving the "upstream assigned label", IBBR should create its own stitching instruction between the "upstream assigned label" and

the down stream label that was signaled to it. IBBR should download these instructions to the datapath.

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

[BIER_ARCH] Wijnands, IJ., Rosen, E., Dolganow, A., Przygienda, T., and S. Aldrin, "Multicast using Bit Index Explicit Replication",

internet-draft draft-ietf-bier-architecture-08, October 2016.

8.2. Informative References

[BIER_MVPN] Rosen, E., Ed., Sivakumar, M., Wijnands, IJ., Aldrin, S., Dolganow, A., and T. Przygienda, "Multicast VPN Using Bier", internet-draft draft-ietf-bier-mvpn-08, January 2017.

[ISIS_BIER_EXTENSIONS] Ginsberg, L., Przygienda, T., Aldrin, S., and Z. Zhang, "BIER Support via ISIS", internet-draft draft-ietf-bier-isis-extensions-06.txt, March 2017.

[OSPF_BIER_EXTENSIONS] Psenak, P., Kumar, N., Wijnands, IJ., Dolganow, A., Przygienda, T., Zhang, Z., and S. Aldrin, "OSPF Extensions for Bit Index Explicit Replication", internet-draft draft-ietf-ospf-bier-extensions-09.txt, March 2017.

7. Acknowledgments Authors would like to acknowledge Jingrong Xie for his comments and help on this draft.

Authors' Addresses

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

Email: hooman.bidgoli@nokia.com

Jayant Kotalwar
Nokia
380 N Bernardo Ave,
Mountain View, CA 94043
US

Email: jayant.kotalwar@nokia.com

Zhaohui Zhang
Juniper Networks

EMail: zzhang@juniper.net

IJsbrand Wijnands
Cisco Systems

EMail: ice@cisco.com

Eddie Leyton
Vrizon

Email: Edward.leyton@verizonwireless.com

Mankamana Mishra
Cisco System
821 alder drive
Milpitas California
USA

Email: mankamis@cisco.com

BIER WG
Internet-Draft
Intended status: Standards Track
Expires: January 4, 2020

Quan Xiong
Greg Mirsky
ZTE Corporation
Fangwei Hu
Individual
Chang Liu
China Unicom
July 3, 2019

BIER BFD
draft-hu-bier-bfd-04.txt

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

the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	2
2. Conventions used in this document	3
2.1. Terminology	3
2.2. Requirements Language	3
3. BIER BFD Encapsulation	3
4. BIER BFD Session Bootstrapping	3
4.1. BIER OAM Bootstrapping	3
4.2. IGP protocol Bootstrapping	4
4.2.1. IS-IS extension for BIER BFD	4
4.2.2. OSPF extension for BIER BFD	5
5. Discriminators and Packet Demultiplexing	6
6. Active Tail in BIER BFD	6
7. Security Considerations	7
8. Acknowledgements	7
9. IANA Considerations	7
10. References	7
10.1. Normative References	7
10.2. Informative References	8
Authors' Addresses	8

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.

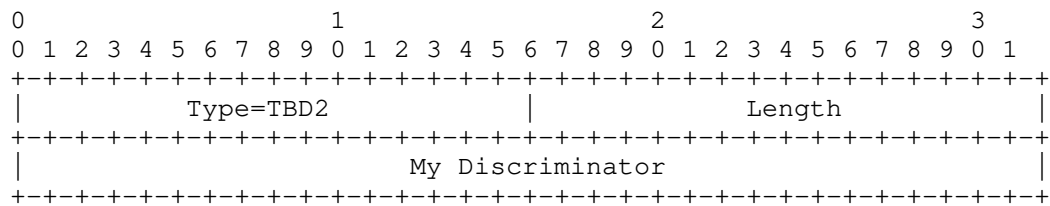


Figure 1: BFD discriminator TLV

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.

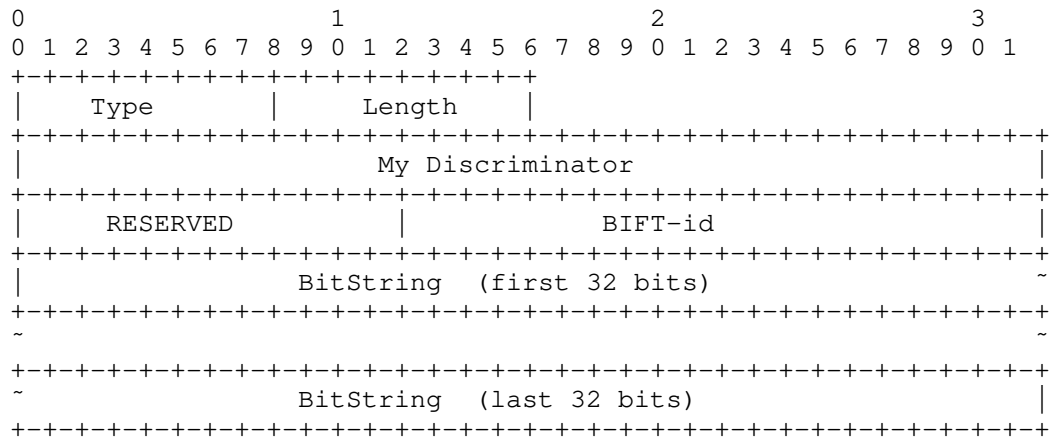


Figure 2: BIER BFD Sub-sub-TLV for IS-IS extension

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.

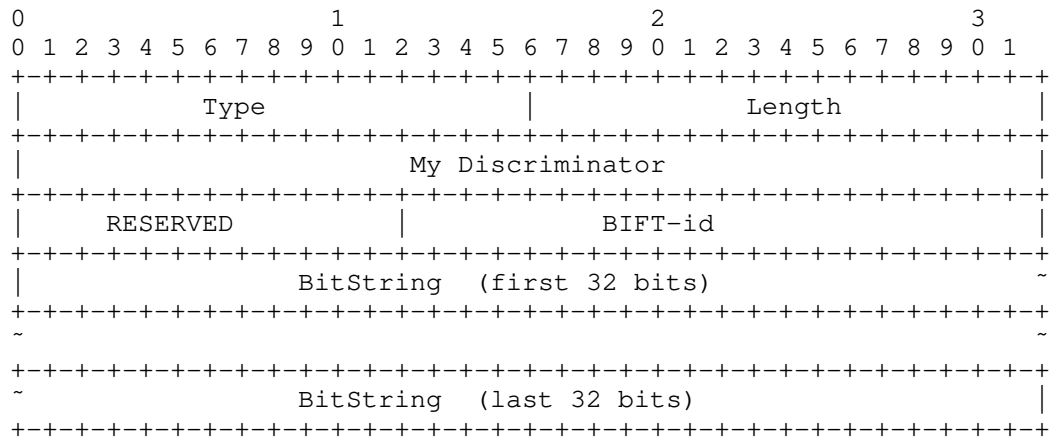


Figure 3: BIER BFD Sub-TLV for OSPF extension

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:

Value	Description	Reference
TBD1	BIER BFD	[this document]
TBD2	BFD discriminator TLV	[this document]
TBD3	BIER BFD Sub-sub-TLV for IS-IS	[this document]
TBD4	BIER BFD Sub-TLV for OSPF	[this document]

Table 1

10. References

10.1. Normative References

- [I-D.ietf-bier-ping] Kumar, N., Pignataro, C., Akiya, N., Zheng, L., Chen, M., and G. Mirsky, "BIER Ping and Trace", draft-ietf-bier-ping-05 (work in progress), April 2019.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC5880] Katz, D. and D. Ward, "Bidirectional Forwarding Detection (BFD)", RFC 5880, DOI 10.17487/RFC5880, June 2010, <<https://www.rfc-editor.org/info/rfc5880>>.

- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.
- [RFC8279] Wijnands, IJ., Ed., Rosen, E., Ed., Dolganow, A., Przygienda, T., and S. Aldrin, "Multicast Using Bit Index Explicit Replication (BIER)", RFC 8279, DOI 10.17487/RFC8279, November 2017, <<https://www.rfc-editor.org/info/rfc8279>>.
- [RFC8401] Ginsberg, L., Ed., Przygienda, T., Aldrin, S., and Z. Zhang, "Bit Index Explicit Replication (BIER) Support via IS-IS", RFC 8401, DOI 10.17487/RFC8401, June 2018, <<https://www.rfc-editor.org/info/rfc8401>>.
- [RFC8444] Psenak, P., Ed., Kumar, N., Wijnands, IJ., Dolganow, A., Przygienda, T., Zhang, J., and S. Aldrin, "OSPFv2 Extensions for Bit Index Explicit Replication (BIER)", RFC 8444, DOI 10.17487/RFC8444, November 2018, <<https://www.rfc-editor.org/info/rfc8444>>.
- [RFC8562] Katz, D., Ward, D., Pallagatti, S., Ed., and G. Mirsky, Ed., "Bidirectional Forwarding Detection (BFD) for Multipoint Networks", RFC 8562, DOI 10.17487/RFC8562, April 2019, <<https://www.rfc-editor.org/info/rfc8562>>.
- [RFC8563] Katz, D., Ward, D., Pallagatti, S., Ed., and G. Mirsky, Ed., "Bidirectional Forwarding Detection (BFD) Multipoint Active Tails", RFC 8563, DOI 10.17487/RFC8563, April 2019, <<https://www.rfc-editor.org/info/rfc8563>>.

10.2. Informative References

- [ISO9577] ISO/IEC TR 9577:1999,, "International Organization for Standardization "Information technology - Telecommunications and Information exchange between systems - Protocol identification in the network layer", 1999.

Authors' Addresses

Quan Xiong
ZTE Corporation
No.6 Huashi Park Rd
Wuhan, Hubei 430223
China

Phone: +86 27 83531060
Email: xiong.quan@zte.com.cn

Greg Mirsky
ZTE Corporation
USA

Email: gregimirsky@gmail.com

Fangwei Hu
Individual

Email: hufwei@gmail.com

Chang Liu
China Unicom
No.9 Shouti Nanlu
Beijing 100048
China

Phone: +86-010-68799999-7294
Email: liuc131@chinaunicom.cn

Network Working Group
Internet-Draft
Intended status: Informational
Expires: April 1, 2021

M. McBride
Futurewei
J. Xie
X. Geng
S. Dhanaraj
Huawei
R. Asati
Cisco
Y. Zhu
China Telecom
G. Mishra
Verizon Inc.
Z. Zhang
Juniper
September 28, 2020

BIER IPv6 Requirements
draft-ietf-bier-ipv6-requirements-09

Abstract

There have been several proposed solutions with BIER being used in IPv6. But there hasn't been a document which describes the problem and lists the requirements. The goal of this document is to describe the general BIER IPv6 encapsulation problem and detail solution requirements, thereby assisting the working group in the development of acceptable 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 April 1, 2021.

Copyright Notice

Copyright (c) 2020 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	2
1.1. Requirements Language	3
1.2. Terminology	3
2. Problem Statement	3
3. Requirements	4
3.1. Mandatory Requirements	4
3.1.1. Support various L2 link types	4
3.1.2. Support BIER architecture	4
3.1.3. Support deployment with Non-BFR routers	4
3.1.4. Support OAM	5
3.2. Optional Requirements	5
3.2.1. Support Fragmentation	5
3.2.2. Support IPSEC ESP	5
4. IANA Considerations	5
5. Security Considerations	6
6. Acknowledgement	6
7. Normative References	6
Authors' Addresses	7

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: one is BIER MPLS encapsulation for MPLS environments, the other is non-MPLS BIER encapsulation to run without MPLS. This document describes non-MPLS BIER encapsulation in IPv6 environments. We explain the requirements of transporting multicast flow overlay payload through an IPv6 network underlay using BIER. The solutions

may use IPv6 forwarding plane and may include IPv6 encapsulation and/or generic IPv6 tunnelling.

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].

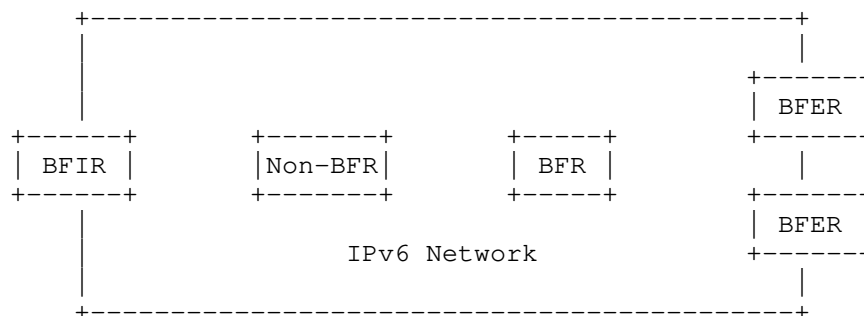
1.2. Terminology

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

2. Problem Statement

The problem is how to transport multicast packets, with non-MPLS BIER encapsulation, in an IPv6 environment. We need to determine where to put the BIER header in this IPv6 environment. With IPv6 encapsulation being increasingly used for unicast services, such as VPN or L2VPN, it may be desirable to have IPv6 encapsulation also used in BIER deployments for multicast services such as MVPN. It may also be desirable to not use IPv6 encapsulation except when IPv6 tunneling (native or GRE/UDP-like) is used to transport BIER packets over BIER-incapable routers.

Below is a simple scenario that needs BIER IPv6-based forwarding:



This scenario depicts the need to replicate BIER packets from a BFIR to BFERs across an IPv6 Service Provider core. Inside the IPv6 network, the BIER header is used to direct the packet from one BFR to the next BFRs, and either a IPv6 header or an L2/tunnel header is used to provide reachability between BFRs. The IPv6 environment may include a variety of link types, may be entirely IPv6, or may be dual stack. There may be cases where not all routers are BFR capable in

the IPv6 environment but still want to deploy BIER. Regardless of the environment, the problem is to deploy BIER, with non-MPLS BIER encapsulation, in an IPv6 network.

3. Requirements

There are several suggested requirements for BIER IPv6 solutions.

In this document, the requirements are divided into two levels: Mandatory and Optional. The requirement levels are determined based on the following factors:

If the requirement is required for a feature that is likely to be a potential deployment, the requirement level will be considered mandatory.

If the impact of not implementing the requirement may block BIER from been deployed, the requirement level will be considered mandatory.

3.1. Mandatory Requirements

Considering that these mandatory requirements are all well-known to the working group, and practical in normal deployment, they will be listed without a detailed description.

3.1.1. Support various L2 link types

The solution should support various kinds of L2 data link types.

3.1.2. Support BIER architecture

The solution must support the BIER architecture.

Supporting different multicast flow overlays, multiple sub-domains, multi-topologies, multiple sets, multiple Bit String Lengths, and deterministic ECMP are considered essential functions of BIER and need to be supported.

3.1.3. Support deployment with Non-BFR routers

The solution must support deployments with BIER-incapable routers. This is beneficial to the deployment of BIER, especially in early deployments when some routers do not support BIER forwarding but support IPv6 forwarding.

3.1.4. Support OAM

BIER OAM tools like [I-D.ietf-bier-ping] and [I-D.ietf-bier-pmmm-oam] should be supported, either directly using existing methods, or by specifying a new method for the same functionality. They are likely to be needed in normal BIER deployment for diagnostics.

3.2. Optional Requirements

The requirements in this section are listed as optional, and each requirement is explained with a detailed scenario. Note that fragmentation and IPSEC ESP are not BIER functions, they are provided by the upper IP layer.

3.2.1. Support Fragmentation

There are some cases where the Fragmentation/Assembly function is needed for BIER to work in an IPv6 network.

For example, a customer IPv6 multicast packet may be 1280 bytes and is required to be transported through an IPv6 network using BIER. Every link of the IPv6 network is no less than the requisite 1280 bytes [RFC8200], but the size of the payload that can be encapsulated in BIER (BIER-MTU) is less than 1280 bytes. In this case, it is not the appropriate action for a BFIR to drop the packet and advertise an MTU to the source [RFC8296]. Instead, some transport mechanism needs to provide the fragmentation and assembly function.

3.2.2. Support IPSEC ESP

There are some cases where the IPSEC ESP function may be needed to transport c-multicast packets through an IPv6 network with confidentiality using BIER technology.

A service provider may want to provide additional security SLA to its customer to ensure that the unencrypted c-multicast packet is not altered in the service provider's network. In this case, if the BIER technology is preferred for the multicast service, BIER with IPSEC ESP support may be a candidate solution. On the other hand, the traffic protection may be better provided via IPSEC or MACSEC at multicast flow overlay over and beyond the BIER domain.

4. IANA Considerations

Some BIER IPv6 encapsulation proposals do not require any action from IANA while other proposals require new IPv6 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.

5. Security Considerations

There are no security issues introduced by this draft.

6. Acknowledgement

Thanks to Eric Rosen for his listed set of initial requirements on the BIER WG mailing list.

7. Normative References

[I-D.ietf-bier-ping]

Nainar, N., Pignataro, C., Akiya, N., Zheng, L., Chen, M., and G. Mirsky, "BIER Ping and Trace", draft-ietf-bier-ping-07 (work in progress), May 2020.

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

Mirsky, G., Zheng, L., Chen, M., and G. Fioccola, "Performance Measurement (PM) with Marking Method in Bit Index Explicit Replication (BIER) Layer", draft-ietf-bier-pmmm-oam-08 (work in progress), May 2020.

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

[RFC8200] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", STD 86, RFC 8200, DOI 10.17487/RFC8200, July 2017, <<https://www.rfc-editor.org/info/rfc8200>>.

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

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

Authors' Addresses

Mike McBride
Futurewei

Email: michael.mcbride@futurewei.com

Jingrong Xie
Huawei

Email: xiejingrong@huawei.com

Xuesong Geng
Huawei

Email: gengxuesong@huawei.com

Senthil Dhanaraj
Huawei

Email: senthil.dhanaraj@huawei.com

Rajiv Asati
Cisco

Email: rajiva@cisco.com

Yongqing Zhu
China Telecom

Email: zhuyq8@chinatelecom.cn

Gyan Mishra
Verizon Inc.

Email: gyan.s.mishra@verizon.com

Zhaohui Zhang
Juniper

Email: zzhang@juniper.net

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: January 9, 2020

P. Pfister
IJ. Wijnands
S. Venaas
Cisco Systems
C. Wang

Z. Zhang
ZTE Corporation
M. Stenberg
July 8, 2019

BIER Ingress Multicast Flow Overlay using Multicast Listener Discovery
Protocols
draft-ietf-bier-mld-02

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.

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	2
2. Terminology	3
3. Overview	3
4. Applicability Statement	4
5. Querier and Listener Specifications	4
5.1. Configuration Parameters	5
5.2. MLDv2 instances.	5
5.2.1. Sending Queries	6
5.2.2. Sending Reports	6
5.2.3. Receiving Queries	7
5.2.4. Receiving Reports	7
5.3. Packet Forwarding	8
6. Security Considerations	8
7. IANA Considerations	8
8. Acknowledgements	8
9. References	8
9.1. Normative References	8
9.2. Informative References	9
Appendix A. BIER Use Case in Data Centers	10
A.1. Convention and Terminology	12
A.2. BIER in data centers	12
A.3. A BIER MLD solution for Virtual Network information	13
Authors' Addresses	14

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:

- o Toward all BMLD Nodes (i.e., providing to the BIER Layer the BFR-ids of all BMLD Nodes).
- o Without the IPv6 router alert option [RFC2711] in the hop-by-hop extension header [RFC8200] (or the IPv4 router alert option [RFC2113] for IPv4).
- o With the IP destination address set to the 'all BMLD Nodes' group address.
- o With the IP source address set to the BFR-Prefix of the sender.
- o 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:

- o Toward all BMLD Queriers (i.e., providing to the BIER layer the BFR-ids of all BMLD Queriers).
- o Without the IPv6 router alert option [RFC2711] in the hop-by-hop extension header [RFC8200] (or the IPv4 router alert option [RFC2113] for IPv4).
- o With the IP destination address set to the 'all BMLD Queriers' group address.
- o With the IP source address set to the BFR-Prefix of the sender.
- o 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:

- o Redirect undesired traffic toward the spoofed router by subscribing to undesired multicast traffic.
- o 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]

Mirsky, G., Przygienda, T., and A. Dolganow, "Path Maximum Transmission Unit Discovery (PMTUD) for Bit Index Explicit Replication (BIER) Layer", draft-ietf-bier-path-mtu-discovery-06 (work in progress), June 2019.

- [I-D.venaas-bier-mtud]
Venaas, S., Wijnands, I., Ginsberg, L., and M. Sivakumar,
"BIER MTU Discovery", draft-venaas-bier-mtud-02 (work in
progress), October 2018.
- [RFC2113] Katz, D., "IP Router Alert Option", RFC 2113,
DOI 10.17487/RFC2113, February 1997,
<<https://www.rfc-editor.org/info/rfc2113>>.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate
Requirement Levels", BCP 14, RFC 2119,
DOI 10.17487/RFC2119, March 1997,
<<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC3376] Cain, B., Deering, S., Kouvelas, I., Fenner, B., and A.
Thyagarajan, "Internet Group Management Protocol, Version
3", RFC 3376, DOI 10.17487/RFC3376, October 2002,
<<https://www.rfc-editor.org/info/rfc3376>>.
- [RFC3810] Vida, R., Ed. and L. Costa, Ed., "Multicast Listener
Discovery Version 2 (MLDv2) for IPv6", RFC 3810,
DOI 10.17487/RFC3810, June 2004,
<<https://www.rfc-editor.org/info/rfc3810>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC
2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174,
May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.
- [RFC8279] Wijnands, IJ., Ed., Rosen, E., Ed., Dolganow, A.,
Przygienda, T., and S. Aldrin, "Multicast Using Bit Index
Explicit Replication (BIER)", RFC 8279,
DOI 10.17487/RFC8279, November 2017,
<<https://www.rfc-editor.org/info/rfc8279>>.

9.2. Informative References

- [RFC2711] Partridge, C. and A. Jackson, "IPv6 Router Alert Option",
RFC 2711, DOI 10.17487/RFC2711, October 1999,
<<https://www.rfc-editor.org/info/rfc2711>>.
- [RFC3306] Haberman, B. and D. Thaler, "Unicast-Prefix-based IPv6
Multicast Addresses", RFC 3306, DOI 10.17487/RFC3306,
August 2002, <<https://www.rfc-editor.org/info/rfc3306>>.
- [RFC4301] Kent, S. and K. Seo, "Security Architecture for the
Internet Protocol", RFC 4301, DOI 10.17487/RFC4301,
December 2005, <<https://www.rfc-editor.org/info/rfc4301>>.

- [RFC5015] Handley, M., Kouvelas, I., Speakman, T., and L. Vicisano, "Bidirectional Protocol Independent Multicast (BIDIR-PIM)", RFC 5015, DOI 10.17487/RFC5015, October 2007, <<https://www.rfc-editor.org/info/rfc5015>>.
- [RFC7348] Mahalingam, M., Dutt, D., Duda, K., Agarwal, P., Kreeger, L., Sridhar, T., Bursell, M., and C. Wright, "Virtual eXtensible Local Area Network (VXLAN): A Framework for Overlaying Virtualized Layer 2 Networks over Layer 3 Networks", RFC 7348, DOI 10.17487/RFC7348, August 2014, <<https://www.rfc-editor.org/info/rfc7348>>.
- [RFC7365] Lasserre, M., Balus, F., Morin, T., Bitar, N., and Y. Rekhter, "Framework for Data Center (DC) Network Virtualization", RFC 7365, DOI 10.17487/RFC7365, October 2014, <<https://www.rfc-editor.org/info/rfc7365>>.
- [RFC7761] Fenner, B., Handley, M., Holbrook, H., Kouvelas, I., Parekh, R., Zhang, Z., and L. Zheng, "Protocol Independent Multicast - Sparse Mode (PIM-SM): Protocol Specification (Revised)", STD 83, RFC 7761, DOI 10.17487/RFC7761, March 2016, <<https://www.rfc-editor.org/info/rfc7761>>.
- [RFC8200] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", STD 86, RFC 8200, DOI 10.17487/RFC8200, July 2017, <<https://www.rfc-editor.org/info/rfc8200>>.

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.

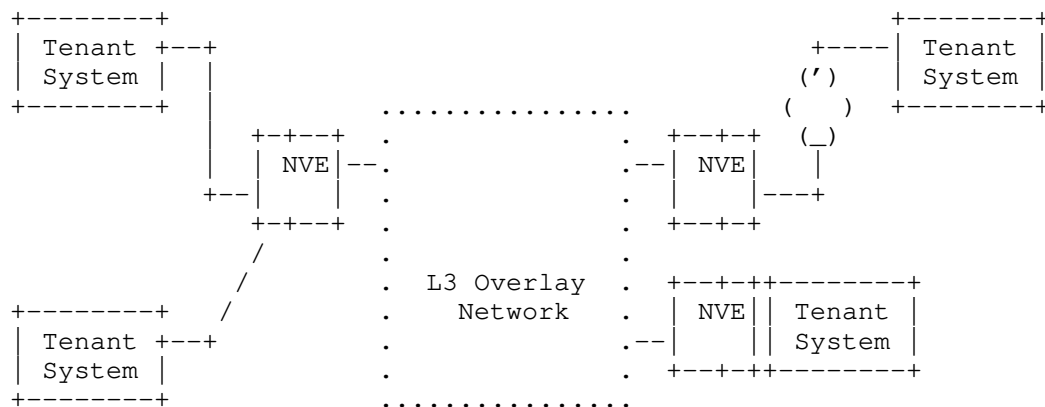


Figure 1: NVO3 Architecture

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 receives 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

Stig Venaas
Cisco Systems
Tasman Drive
San Jose, CA 95134
USA

Email: stig@cisco.com

Cui (Linda) Wang

Email: lindawangjoy@gmail.com

Zheng (Sandy) Zhang
ZTE Corporation
No.50 Software Avenue, Yuhuatai District
Nanjing, CA
China

Email: zhang.zheng@zte.com.cn

Markus Stenberg
Helsinki 00930
Finland

Email: markus.stenberg@iki.fi

Network Working Group
Internet-Draft
Intended status: Informational
Expires: December 30, 2019

D. Trossen
InterDigital Europe, Ltd
A. Rahman
C. Wang
InterDigital Communications, LLC
T. Eckert
Futurewei
June 28, 2019

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.

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	2
1.1. Reference Deployment	3
2. Conventions used in this document	5
3. Use Cases	5
3.1. HTTP-based Steaming	6
3.2. HTTP-based Software Updates	7
4. Requirements	7
5. Realization over IP Multicast	8
5.1. Mapping to Requirements	9
5.2. Problems	9
6. Realization over BIER	10
6.1. Description of a "BIER Multicast Overlay" to support HTTP Multicast	10
6.1.1. BIER Multicast Overlay Components	11
6.1.2. BIER Multicast Overlay Operations	11
6.2. Achieving Multicast Responses	13
6.3. BIER multicast Overlay Traffic Management	14
7. IANA Considerations	14
8. Security Considerations	14
9. Informative References	15
Authors' Addresses	16

1. Introduction

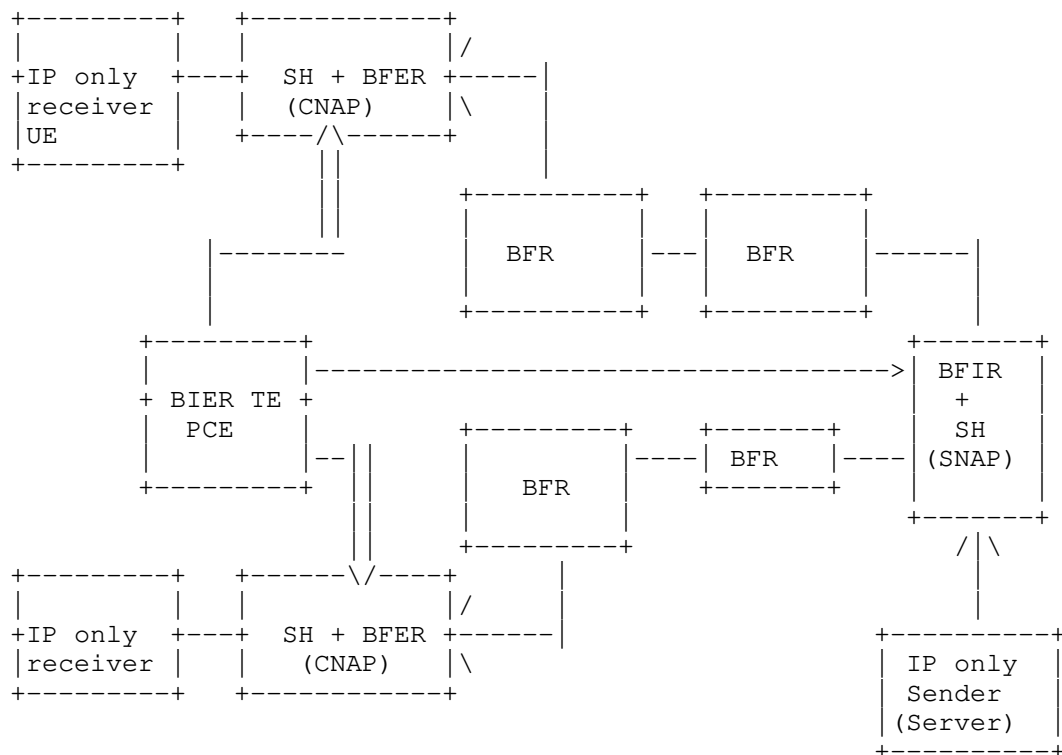
The BIER Use Cases document [I-D.ietf-bier-use-cases] describes an "HTTP Level Multicast" scenario, where HTTP Responses are carried over a BIER multicast infrastructure to multiple clients. Especially rate-adaptive HTTP solutions can benefit from the dynamic multicast group membership changes enabled 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.



[SH : Service Handler, CNAP : Client Network Attachment Point]
 [SNAP : Server Network Attachment Point]
 [PCE : Path Computation Element]

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 send and receive 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 Steaming

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 bits 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:

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

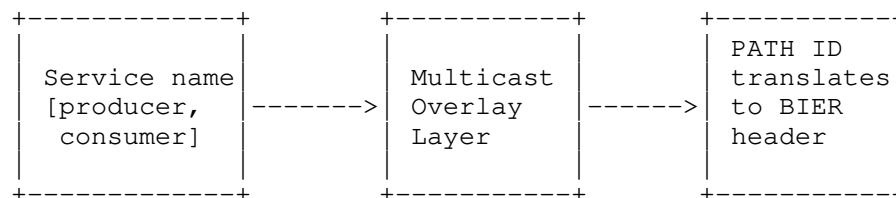


Figure 2: Service name to Path ID translation

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.

6.1.1.1. BIER Multicast Overlay Components

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

- o 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.
- o 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.
- o 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.

6.1.1.2. 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.

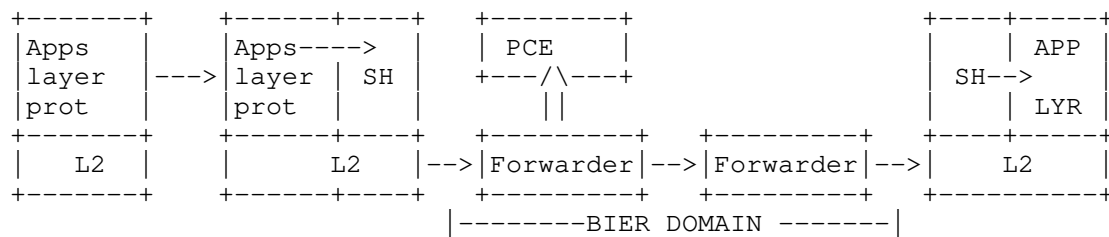


Figure 3: Protocol for Multicast Overlay Layer

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.

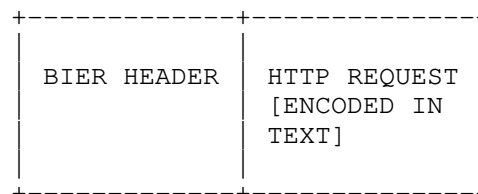


Figure 4: Encapsulation of Service Request

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.

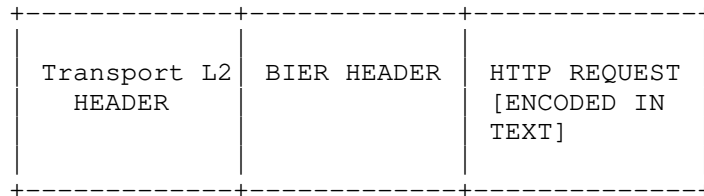


Figure 5: Transport Encapsulation of BIER payload

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_bluebook.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>>.

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

[TR_IPMC_ABR] CableLabs, "IP Multicast Adaptive Bit Rate Architecture Technical Report", OC-TR-IP-MULTI-ARCH-V01-141112 C01, October 2016, <<https://community.cablelabs.com/wiki/plugins/servlet/cablelabs/alfresco/download?id=51b3c11a-3ba4-40ab-b234-42700e0d4669;1.0>>.

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

BIER Working Group
Internet-Draft
Intended status: Standards Track
Expires: 2 October 2022

G. Mirsky
Ericsson
L. Zheng
Individual Contributor
M. Chen
G. Fioccola
Huawei Technologies
31 March 2022

Performance Measurement (PM) with Marking Method in Bit Index Explicit
Replication (BIER) Layer
draft-ietf-bier-pmmm-oam-12

Abstract

This document describes the applicability of a hybrid performance measurement method for packet loss and packet delay measurements of a 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 2 October 2022.

Copyright Notice

Copyright (c) 2022 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 Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

Table of Contents

1. Introduction	2
2. Conventions used in this document	3
2.1. Terminology	3
2.2. Requirements Language	3
3. OAM Field in BIER Header	3
4. Theory of Operation	4
4.1. Single-Marking Enabled Measurement	5
4.2. Double-Marking Enabled Measurement	6
4.3. Operational Considerations	7
5. IANA Considerations	7
6. Security Considerations	7
7. Acknowledgement	8
8. References	8
8.1. Normative References	8
8.2. Informative References	9
Authors' Addresses	9

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. [I-D.fioccola-rfc8321bis] and [I-D.fioccola-rfc8889bis] describe a hybrid performance measurement method, according to the classification of measurement methods in [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 R-5 and R-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. Terms PNPM and "marking method" in this document are used interchangeably.

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

This document uses the terms related to the Alternate Marking Method as defined in [I-D.fioccola-rfc8321bis], [I-D.fioccola-rfc8889bis]. This document uses the terms related to the Bit Indexed Explicit Replication as defined in [RFC8296].

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 of the PNPM method. The context of interpretation of the OAM field MAY be signaled via the control plane or configured using an extension to the BIER YANG data model [I-D.ietf-bier-bier-yang]. These extensions are outside the scope of this document.

```

      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.

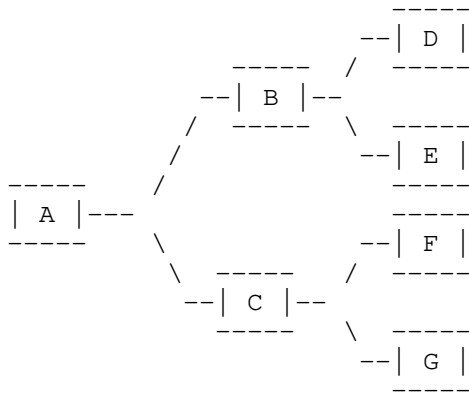


Figure 2: Multicast network

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. The marking method can be used on multiple flows concurrently. Demultiplexing of monitored flows might be achieved using n-tuple, for example, two-tuple as combination of the values in the Entropy and BFIR-id fields [RFC8296]. Also, that can be achieved by using an explicit Flow Identifier. The definition of the Flow Identifier is outside the scope of this specification. 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.

4.1. Single-Marking Enabled Measurement

As explained in [I-D.fioccola-rfc8321bis], 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 is 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 5 [I-D.fioccola-rfc8321bis] 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.

Selection of a time interval to switch the marking of a batch of packets should be based on the service requirements. In the course of the regular operation, reports, including performance metrics like packet loss ratio, packet delay, and inter-packet delay variation, are logged every 15 minutes. Thus, it is reasonable to maintain the duration of the measurement interval at 5 minutes with 100 measurements per each interval. To support these measurements, marking of the packet batch is switched every 3 seconds. In case when performance metrics are required in near-real-time, the duration interval of a single batch of identically marked packets will be in the range of tens of milliseconds.

5. IANA Considerations

This document sets no requirements to IANA. This section can be removed before the publication.

6. Security Considerations

Regarding using the marking method, [I-D.fioccola-rfc8321bis] 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

Comments from Alvaro Retana helped improve the document and are much appreciated.

Reviews and comments from Quan Xiong and Xiao Min are thankfully acknowledged.

8. References

8.1. Normative References

- [I-D.fioccola-rfc8321bis]
Fioccola, G., Cociglio, M., Mirsky, G., Mizrahi, T., Zhou, T., and X. Min, "Alternate-Marking Method", Work in Progress, Internet-Draft, draft-fioccola-rfc8321bis-03, 23 February 2022, <<https://datatracker.ietf.org/doc/html/draft-fioccola-rfc8321bis-03>>.
- [I-D.fioccola-rfc8889bis]
Fioccola, G., Cociglio, M., Sapio, A., Sisto, R., and T. Zhou, "Multipoint Alternate-Marking Method", Work in Progress, Internet-Draft, draft-fioccola-rfc8889bis-03, 23 February 2022, <<https://datatracker.ietf.org/doc/html/draft-fioccola-rfc8889bis-03>>.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.
- [RFC8296] Wijnands, IJ., Ed., Rosen, E., Ed., Dolganow, A., Tantsura, J., Aldrin, S., and I. Meilik, "Encapsulation for Bit Index Explicit Replication (BIER) in MPLS and Non-MPLS Networks", RFC 8296, DOI 10.17487/RFC8296, January 2018, <<https://www.rfc-editor.org/info/rfc8296>>.

8.2. Informative References

- [I-D.ietf-bier-bier-yang]
Chen, R., Hu, F., Zhang, Z., Dai, X., and M. Sivakumar,
"YANG Data Model for BIER Protocol", Work in Progress,
Internet-Draft, draft-ietf-bier-bier-yang-07, 8 September
2020, <<https://datatracker.ietf.org/doc/html/draft-ietf-bier-bier-yang-07>>.
- [I-D.ietf-bier-oam-requirements]
Mirsky, G., Kumar, N., Chen, M., and S. Pallagatti,
"Operations, Administration and Maintenance (OAM)
Requirements for Bit Index Explicit Replication (BIER)
Layer", Work in Progress, Internet-Draft, draft-ietf-bier-
oam-requirements-11, 15 November 2020,
<[https://datatracker.ietf.org/doc/html/draft-ietf-bier-
oam-requirements-11](https://datatracker.ietf.org/doc/html/draft-ietf-bier-oam-requirements-11)>.
- [RFC7799] Morton, A., "Active and Passive Metrics and Methods (with
Hybrid Types In-Between)", RFC 7799, DOI 10.17487/RFC7799,
May 2016, <<https://www.rfc-editor.org/info/rfc7799>>.
- [RFC8279] Wijnands, IJ., Ed., Rosen, E., Ed., Dolganow, A.,
Przygienda, T., and S. Aldrin, "Multicast Using Bit Index
Explicit Replication (BIER)", RFC 8279,
DOI 10.17487/RFC8279, November 2017,
<<https://www.rfc-editor.org/info/rfc8279>>.

Authors' Addresses

Greg Mirsky
Ericsson
Email: gregimirsky@gmail.com

Lianshu Zheng
Individual Contributor
Email: veronique_zheng@hotmail.com

Mach Chen
Huawei Technologies
Email: mach.chen@huawei.com

Giuseppe Fioccola
Huawei Technologies
Email: giuseppe.fioccola@huawei.com

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: 27 October 2022

T.T.E. Eckert, Ed.
Futurewei
M.M. Menth
University of Tuebingen
G.C. Cauchie
KOEVOO
April 2022

Tree Engineering for Bit Index Explicit Replication (BIER-TE)
draft-ietf-bier-te-arch-13

Abstract

This memo describes per-packet stateless strict and loose path steered replication and forwarding for "Bit Index Explicit Replication" (BIER, RFC8279) packets. It is called BIER Tree Engineering (BIER-TE) and is intended to be used as the path steering mechanism for Traffic Engineering with BIER.

BIER-TE introduces a new semantic for "bit positions" (BP). They indicate adjacencies of the network topology, as opposed to (non-TE) BIER in which BPs indicate "Bit-Forwarding Egress Routers" (BFER). A BIER-TE packets BitString therefore indicates the edges of the (loop-free) tree that the packet is forwarded across by BIER-TE. BIER-TE can leverage BIER forwarding engines with little changes. Co-existence of BIER and BIER-TE forwarding in the same domain is possible, for example by using separate BIER "sub-domains" (SDs). Except for the optional routed adjacencies, BIER-TE does not require a BIER routing underlay, and can therefore operate without depending on an "Interior Gateway Routing protocol" (IGP).

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 3 October 2022.

Copyright Notice

Copyright (c) 2022 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 Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

Table of Contents

1. Overview	3
1.1. Requirements Language	5
2. Introduction	5
2.1. Basic Examples	5
2.2. BIER-TE Topology and adjacencies	8
2.3. Relationship to BIER	9
2.4. Accelerated/Hardware forwarding comparison	11
3. Components	11
3.1. The Multicast Flow Overlay	12
3.2. The BIER-TE Control Plane	12
3.2.1. The BIER-TE Controller	14
3.2.1.1. BIER-TE Topology discovery and creation	14
3.2.1.2. Engineered Trees via BitStrings	15
3.2.1.3. Changes in the network topology	16
3.2.1.4. Link/Node Failures and Recovery	16
3.3. The BIER-TE Forwarding Plane	16
3.4. The Routing Underlay	17
3.5. Traffic Engineering Considerations	17
4. BIER-TE Forwarding	18
4.1. The BIER-TE Bit Index Forwarding Table (BIFT)	18
4.2. Adjacency Types	20
4.2.1. Forward Connected	21
4.2.2. Forward Routed	21
4.2.3. ECMP	21
4.2.4. Local Decapsulation	22
4.3. Encapsulation / Co-existence with BIER	22
4.4. BIER-TE Forwarding Pseudocode	23
4.5. BFR Requirements for BIER-TE forwarding	26
5. BIER-TE Controller Operational Considerations	27
5.1. Bit Position Assignments	27
5.1.1. P2P Links	27
5.1.2. BFER	27

5.1.3.	Leaf BFERs	27
5.1.4.	LANs	29
5.1.5.	Hub and Spoke	30
5.1.6.	Rings	30
5.1.7.	Equal Cost MultiPath (ECMP)	31
5.1.8.	Forward Routed adjacencies	34
5.1.8.1.	Reducing bit positions	34
5.1.8.2.	Supporting nodes without BIER-TE	35
5.1.9.	Reuse of bit positions (without DNC)	35
5.1.10.	Summary of BP optimizations	36
5.2.	Avoiding duplicates and loops	37
5.2.1.	Loops	38
5.2.2.	Duplicates	38
5.3.	Managing SI, sub-domains and BFR-ids	39
5.3.1.	Why SI and sub-domains	39
5.3.2.	Assigning bits for the BIER-TE topology	40
5.3.3.	Assigning BFR-id with BIER-TE	41
5.3.4.	Mapping from BFR to BitStrings with BIER-TE	42
5.3.5.	Assigning BFR-ids for BIER-TE	43
5.3.6.	Example bit allocations	43
5.3.6.1.	With BIER	43
5.3.6.2.	With BIER-TE	44
5.3.7.	Summary	45
6.	Security Considerations	46
7.	IANA Considerations	47
8.	Acknowledgements	47
9.	Change log [RFC Editor: Please remove]	48
10.	References	61
10.1.	Normative References	61
10.2.	Informative References	61
	Appendix A. BIER-TE and Segment Routing (SR)	64
	Authors' Addresses	65

1. Overview

BIER-TE is based on the (non-TE) BIER architecture, terminology and packet formats as described in [RFC8279] and [RFC8296]. This document describes BIER-TE in the expectation that the reader is familiar with these two documents.

BIER-TE introduces a new semantic for "bit positions" (BP). They indicate adjacencies of the network topology, as opposed to (non-TE) BIER in which BPs indicate "Bit-Forwarding Egress Routers" (BFER). A BIER-TE packets BitString therefore indicates the edges of the (loop-free) tree that the packet is forwarded across by BIER-TE. With BIER-TE, the "Bit Index Forwarding Table" (BIFT) of each "Bit Forwarding Router" (BFR) is only populated with BP that are adjacent to the BFR in the BIER-TE Topology. Other BPs are empty in the BIFT.

The BFR replicate and forwards BIER packets to adjacent BPs that are set in the packet. BPs are normally also cleared upon forwarding to avoid duplicates and loops.

BIER-TE can leverage BIER forwarding engines with little or no changes. It can also co-exist with BIER forwarding in the same domain, for example by using separate BIER sub-domains. Except for the optional routed adjacencies, BIER-TE does not require a BIER routing underlay, and can therefore operate without depending on an "Interior Gateway Routing protocol" (IGP).

This document is structured as follows:

- * Section 2 introduces BIER-TE with two forwarding examples, followed by an introduction of the new concepts of the BIER-TE (overlay) topology and finally a summary of the relationship between BIER and BIER-TE and a discussion of accelerated hardware forwarding.
- * Section 3 describes the components of the BIER-TE architecture, Flow overlay, BIER-TE layer with the BIER-TE control plane (including the BIER-TE controller) and BIER-TE forwarding plane, and the routing underlay.
- * Section 4 specifies the behavior of the BIER-TE forwarding plane with the different type of adjacencies and possible variations of BIER-TE forwarding pseudocode, and finally the mandatory and optional requirements.
- * Section 5 describes operational considerations for the BIER-TE controller, foremost how the BIER-TE controller can optimize the use of BP by using specific type of BIER-TE adjacencies for different type of topological situations, but also how to assign bits to avoid loops and duplicates (which in BIER-TE does not come for free), and finally how "Set Identifier" (SI), "sub-domain" (SD) and BFR-ids can be managed by a BIER-TE controller, examples and summary.
- * Appendix A concludes the technology specific sections of the document by further relating BIER-TE to Segment Routing (SR).

Note that related work, [I-D.ietf-roll-ccast] uses Bloom filters [Bloom70] to represent leaves or edges of the intended delivery tree. Bloom filters in general can support larger trees/topologies with fewer addressing bits than explicit BitStrings, but they introduce the heuristic risk of false positives and cannot clear bits in the BitString during forwarding to avoid loops. For these reasons, BIER-TE uses explicit BitStrings like BIER. The explicit BitStrings of BIER-TE can also be seen as a special type of Bloom filter, and this is how related work [ICC] describes it.

1.1. Requirements Language

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

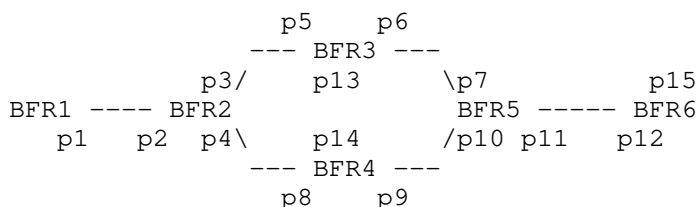
2. Introduction

2.1. Basic Examples

BIER-TE forwarding is best introduced with simple examples. These examples use formal terms defined later in the document (Figure 4), including `forward_connected()`, `forward_routed()` and `local_decap()`.

BIER-TE Topology:

Diagram:



(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
       p15 -> local_decap()

```

Figure 1: BIER-TE basic example

Consider the simple network in the above BIER-TE overview example picture with 6 BFRs. p1...p15 are the bit positions used. All BFRs can act as an ingress BFR (BFIR), BFR1, BFR3, BFR4 and BFR6 can also be BFERs. 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,p15). When this packet is examined by BIER-TE on BFR1, the only bit position 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. p15 finally makes BFR6 receive and decapsulate the packet.

To send a copy to BFR6 via BFR4 and also a copy to BFR3, the BitString needs to be (p2,p5,p8,p10,p12,p13,p15). 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,p15), the packet would be copied by BFR5 towards BFR3 because of p6 instead of being copied by BFR2 to BFR3 because of p5 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.

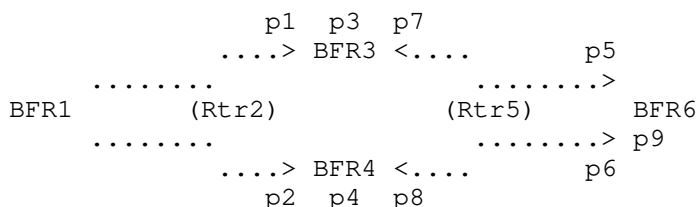
BIER-TE has various options to minimize BP assignments, many of which are based on out-of-band knowledge about the required multicast traffic paths and bandwidth consumption in the network, such as from pre-deployment planning.

Figure 2 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. To emphasize non-L2, but routed/tunneled forwarding of BIER-TE packets, these adjacencies are called "forward_routed". 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:



(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:  p7  -> forward_routed() to BFR3
       p8  -> forward_routed() to BFR4
       p9  -> local_decap()
  
```

Figure 2: BIER-TE basic overlay example

To send a BIER-TE packet from BFR1 via BFR3 to be received by BFR6, the BitString is (p1,p5,p9). From BFR1 via BFR4 to be received by BFR6, the BitString is (p2,p6,p9). A packet from BFR1 to be received by BFR3,BFR4 and from BFR3 to be received by BFR6 uses (p1,p2,p3,p4,p5,p9). A packet from BFR1 to be received by BFR3,BFR4 and from BFR4 to be received by BFR6 uses (p1,p2,p3,p4,p6,p9). A packet from BFR1 to be received by BFR4, and from BFR4 to be received by BFR6 and from there to be received by BFR3 uses (p2,p3,p4,p6,p7,p9). A packet from BFR1 to be received by BFR3, and from BFR3 to be received by BFR6 there to be received by BFR4 uses (p1,p3,p4,p5,p8,p9).

2.2. BIER-TE Topology and adjacencies

The key new component in BIER-TE compared to (non-TE) BIER is the BIER-TE topology as introduced through the two examples in Section 2.1. It is used to control where replication can or should happen and how to minimize the required number of BP for adjacencies.

The BIER-TE Topology consists of the BIFTs of all the BFR and can also be expressed as a directed graph where the edges are the adjacencies between the BFRs labelled with the BP used for the adjacency. 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 in Section 5.2.

If the BIER-TE topology represents (a subset of) the underlying (layer 2) topology of the network as shown in the first example, this may be called a "native" BIER-TE topology. A topology consisting only of "forward_routed" adjacencies as shown in the second example may be called an "overlay" BIER-TE topology. A BIER-TE topology with both forward_connected() and forward_routed() adjacencies may be called a "hybrid" BIER-TE topology.

2.3. Relationship to BIER

BIER-TE is designed so that its forwarding plane is a simple extension to the (non-TE) BIER forwarding plane, hence allowing for it to be added to BIER deployments where it can be beneficial.

BIER-TE is also intended as an option to expand the BIER architecture into deployments where (non-TE) BIER may not be the best fit, such as statically provisioned networks with needs for path steering but without desire for distributed routing protocols.

1. BIER-TE inherits the following aspects from BIER unchanged:

1. The fundamental purpose of per-packet signaled replication and delivery via a BitString.
2. The overall architecture consisting of three layers, flow overlay, BIER(-TE) layer and routing underlay.
3. The supported encapsulations [RFC8296].
4. The semantic of all [RFC8296] header elements used by the BIER-TE forwarding plane other than the semantic of the BP in the BitString.
5. The BIER forwarding plane, except for how bits have to be cleared during replication.

2. BIER-TE has the following key changes with respect to BIER:

1. In BIER, bits in the BitString of a BIER packet header indicate a BFER and bits in the BIFT indicate the BIER control plane calculated next-hop toward that BFER. In BIER-

TE, a bit in the BitString of a BIER packet header indicates an adjacency in the BIER-TE topology, and only the BFR that is the upstream of that adjacency has its BP populated with the adjacency in its BIFT.

2. In BIER, the implied reference options for the core part of the BIER layer control plane are the BIER extensions for distributed routing protocols. This includes ISIS/OSPF extensions for BIER, [RFC8401] and [RFC8444].
 3. The reference option for the core part of the BIER-TE control plane is the BIER-TE controller. Nevertheless, both the BIER and BIER-TE BIFTs forwarding plane state could equally be populated by any mechanism.
 4. Assuming the reference options for the control plane, BIER-TE replaces in-network autonomous path calculation by explicit paths calculated by the BIER-TE controller.
3. The following elements/functions described in the BIER architecture are not required by the BIER-TE architecture:
1. "Bit Index Routing Tables" (BIRTs) are not required on BFRs for BIER-TE when using a BIER-TE controller because the controller can directly populate the BIFTs. In BIER, BIRTs are populated by the distributed routing protocol support for BIER, allowing BFRs to populate their BIFTs locally from their BIRTs. Other BIER-TE control plane or management plane options may introduce requirements for BIRTs for BIER-TE BFRs.
 2. The BIER-TE layer forwarding plane does not require BFRs to have a unique BP and therefore also no unique BFR-id. See Section 5.1.3.
 3. Identification of BFRs by the BIER-TE control plane is outside the scope of this specification. Whereas the BIER control plane uses BFR-ids in its BFR to BFR signaling, a BIER-TE controller may choose any form of identification deemed appropriate.
 4. BIER-TE forwarding does not require the BFIR-id field of the BIER packet header.
4. Co-existence of BIER and BIER-TE in the same network requires the following:

1. The BIER/BIER-TE packet header needs to allow addressing both BIER and BIER-TE BIFTs. Depending on the encapsulation option, the same SD may or may not be reusable across BIER and BIER-TE. See Section 4.3. In either case, a packet is always only forwarded end-to-end via BIER or via BIER-TE (ships in the nights forwarding).
2. BIER-TE deployments will have to assign BFR-ids to BFRs and insert them into the BFIR-id field of BIER packet headers as BIER does, whenever the deployment uses (unchanged) components developed for BIER that use BFR-id, such as multicast flow overlays or BIER layer control plane elements. See also Section 5.3.3.

2.4. Accelerated/Hardware forwarding comparison

BIER-TE forwarding rules, especially the BitString parsing are designed to be as close as possible to those of BIER in the expectation that this eases the programming of BIER-TE forwarding code and/or BIER-TE forwarding hardware on platforms supporting BIER. The pseudocode in Section 4.4 shows how existing (non-TE) BIER/BIFT forwarding can be modified to support the required BIER-TE forwarding functionality (Section 4.5), by using BIER BIFT's "Forwarding Bit Mask" (F-BM): Only the clearing of bits to avoid duplicate packets to a BFR's neighbor is skipped in BIER-TE forwarding because it is not necessary and could not be done when using BIER F-BM.

Whether to use BIER or BIER-TE forwarding is simply a choice of the mode of the BIFT indicated by the packet (BIER or BIER-TE BIFT). This is determined by the BFR configuration for the encapsulation, see Section 4.3.

3. Components

BIER-TE can be thought of being constituted from the same three layers as BIER: The "multicast flow overlay", the "BIER layer" and the "routing underlay". The following picture also shows how the "BIER layer" is constituted from the "BIER-TE forwarding plane" and the "BIER-TE control plane" represent by the "BIER-TE Controller".

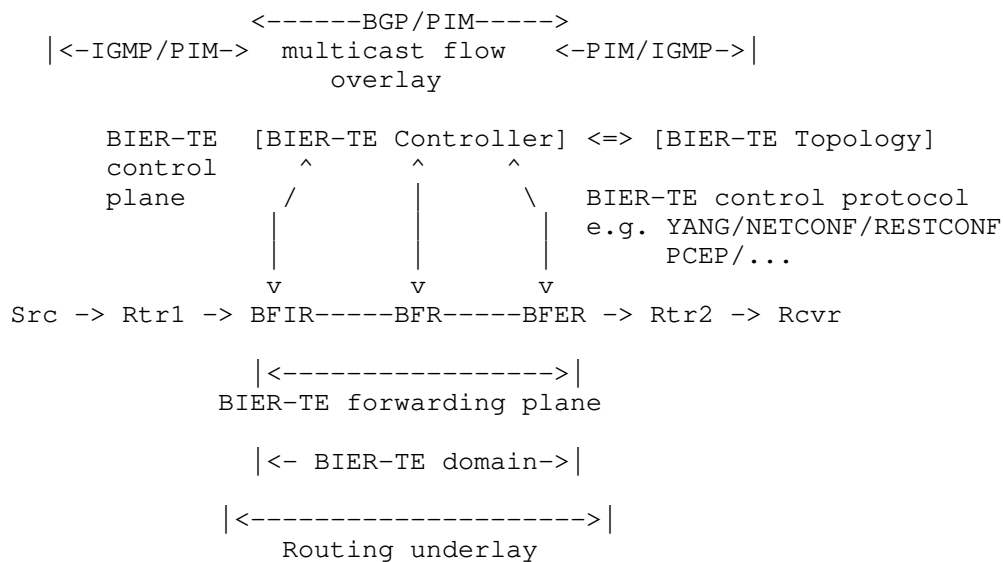


Figure 3: BIER-TE architecture

3.1. The Multicast Flow Overlay

The Multicast Flow Overlay has the same role as described for BIER in [RFC8279], Section 4.3. See also Section 3.2.1.2.

When a BIER-TE controller is used, then the signaling for the Multicast Flow Overlay may also be preferred to operate through a central point of control. For BGP based overlay flow services such as "Multicast VPN Using BIER" ([RFC8556]) this can be achieved by making the BIER-TE controller operate as a BGP Route Reflector ([RFC4456]) and combining it with signaling through BGP or a different protocol for the BIER-TE controller calculated BitStrings. See Section 3.2.1.2 and Section 5.3.4.

3.2. The BIER-TE Control Plane

In the (non-TE) BIER architecture [RFC8279], the BIER control plane is not explicitly separated from the BIER forwarding plane, but instead their functions are summarized together in Section 4.2. Example standardized options for the BIER control plane include ISIS/OSPF extensions for BIER, [RFC8401] and [RFC8444].

For BIER-TE, the control plane includes at minimum the following functionality.

1. BIER-TE topology control: During initial provisioning of the

network and/or during modifications of its topology and/or services, the protocols and/or procedures to establish BIER-TE BIFTs:

1. Determine the desired BIER-TE topology for a BIER-TE sub-domains: the native and/or overlay adjacencies that are assigned to BPs. Topology discovery is discussed in Section 3.2.1.1 and the various aspects of the BIER-TE controllers determinations about the topology are discussed throughout Section 5
 2. Determine the per-BFR BIFT from the BIER-TE topology. This is achieved by simply extracting the adjacencies of the BFR from the BIER-TE topology and populating the BFRs BIFT with them.
 3. Optionally assign BFR-ids to BFIRs for later insertion into BIER headers on BFIRs as BFIR-id. Alternatively, BFIR-id in BIER packet headers may be managed solely by the flow overlay layer and/or be unused. This is discussed in Section 5.3.3.
 4. Install/update the BIFTs into the BFRs and optionally BFR-ids into BFIRs. This is discussed in Section 3.2.1.1.
2. BIER-TE tree control: During operations of the network, protocols and/or procedures to support creation/change/removal of overlay flows on BFIRs:
1. Process the BIER-TE requirements for the multicast overlay flow: BFIR and BFERs of the flow as well as policies for the path selection of the flow. This is discussed in Section 3.5.
 2. Determine the BitStrings and optionally Entropy. This is discussed in Section 3.2.1.2, Section 3.5 and Section 5.3.4.
 3. Install state on the BFIR to impose the desired BIER packet header(s) for packets of the overlay flow. Different aspects of this and the next point are discussed throughout Section 3.2.1 and in Section 4.3, but the main responsibility of these two points is with the Multicast Flow Overlay (Section 3.1), which is architecturally inherited from BIER.
 4. Install the necessary state on the BFERs to decapsulate the BIER packet header and properly dispatch its payload.

3.2.1. The BIER-TE Controller

[RFC-Editor: the following text has three references to anchors topology-control, topology-control-1 and tree-control. Unfortunately, XMLv2 does not offer any tagging that reasonable references are generated (i had this problem already in RFCs last year. Please make sure there are useful-to-read cross-references in the RFC in these three places after you convert to XMLv3.)]

This architecture describes the BIER-TE control plane as shown in Figure 3 to consist of:

- * A BIER-TE controller.
- * BFR data-models and protocols to communicate between controller and BFRs in support of BIER-TE topology control (Section 3.2), such as YANG/NETCONF/RESTCONF ([RFC7950]/[RFC6241]/[RFC8040]).
- * BFR data-models and protocols to communicate between controller and BFIR in support of BIER-TE tree control (Section 3.2), such as BIER-TE extensions for [RFC5440].

The single, centralized BIER-TE controller is used in this document as reference option for the BIER-TE control plane but other options are equally feasible. The BIER-TE control plane could equally be implemented without automated configuration/protocols, by an operator via CLI on the BFRs. In that case, operator configured local policy on the BFIR would have to determine how to set the appropriate BIER header fields. The BIER-TE control plane could also be decentralized and/or distributed, but this document does not consider any additional protocols and/or procedures which would then be necessary to coordinate its (distributed/decentralized) entities to achieve the above described functionality.

3.2.1.1. BIER-TE Topology discovery and creation

The first item of BIER-TE topology control (Section 3.2, Paragraph 3, Item 2.2.1) includes network topology discovery and BIER-TE topology creation. The latter describes the process by which a Controller determines which routers are to be configured as BFRs and the adjacencies between them.

In statically managed networks, such as in industrial environments, both discovery and creation can be a manual/offline process.

In other networks, topology discovery may rely on protocols including extending a "Link-State-Protocol" based IGP into the BIER-TE controller itself, [RFC7752] (BGP-LS) or [RFC8345] (YANG topology) as well as BIER-TE specific methods, for example via [I-D.ietf-bier-te-yang]. These options are non-exhaustive.

Dynamic creation of the BIER-TE topology can be as easy as mapping the network topology 1:1 to the BIER-TE topology by assigning a BP for every network subnet adjacency. In larger networks, it likely involves more complex policy and optimization decisions including how to minimize the number of BPs required and how to assign BPs across different BitStrings to minimize the number of duplicate packets across links when delivering an overlay flow to BFER using different SIs/BitStrings. These topics are discussed in Section 5.

When the BIER-TE topology is determined, the BIER-TE Controller then pushes the BitPositions/adjacencies to the BIFT of the BFRs. On each BFR only those SI:BitPositions are populated that are adjacencies to other BFRs in the BIER-TE topology.

Communications between the BIER-TE Controller and BFRs for both BIER-TE topology control and BIER-TE tree control is ideally via standardized protocols and data-models such as NETCONF/RESTCONF/YANG/PCP. Vendor-specific CLI on the BFRs is also an option (as in many other SDN solutions lacking definition of standardized data models).

3.2.1.2. Engineered Trees via BitStrings

In BIER, the same set of BFER in a single sub-domain is always encoded as the same BitString. In BIER-TE, the BitString used to reach the same set of BFER in the same sub-domain can be different for different overlay flows because the BitString encodes the paths towards the BFER, so the BitStrings from different BFIR to the same set of BFER will often be different. Likewise, the BitString from the same BFIR to the same set of BFER can be different for different overlay flows for policy reasons such as shortest path trees, Steiner trees (minimum cost trees), diverse path trees for redundancy and so on.

See also [I-D.ietf-bier-multicast-http-response] for an application leveraging BIER-TE engineered trees.

3.2.1.3. Changes in the network topology

If the network topology changes (not failure based) so that adjacencies that are assigned to bit positions are no longer needed, the BIER-TE Controller can re-use those bit positions for new adjacencies. First, these bit positions 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.

3.2.1.4. Link/Node Failures and Recovery

When link or nodes fail or recover in the topology, BIER-TE could quickly respond with FRR procedures such as [I-D.eckert-bier-te-frr], the details of which are out of scope for this document. It can also more slowly react by recalculating the BitStrings of affected multicast flows. This reaction is slower than the FRR procedure because the BIER-TE 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.

3.3. The BIER-TE Forwarding Plane

[RFC-editor Q: "is constituted from" / "consists of" / "composed from..." ???]

The BIER-TE Forwarding Plane is constituted from the following components:

1. On a BFIR, imposition of the BIER header for packets from overlay flows. This is driven by a combination of state established by the BIER-TE control plane and/or the multicast flow overlay as explained in Section 3.1.
2. On BFRs (including BFIR and BFER), forwarding/replication of BIER packets according to their SD, SI, "BitStringLength" (BSL), BitString and optionally Entropy fields as explained in Section 4. Processing of other BIER header fields such as DSCP is outside the scope of this document.
3. On BFERs, removal of the BIER header and dispatching of the payload according to state created by the BIER-TE control plane and/or overlay layer.

When the BIER-TE Forwarding Plane receives a packet, it simply looks up the bit positions that are set in the BitString of the packet in the BIFT that was populated by the BIER-TE Controller. 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 clears all BPs in the BitString of the packet for which the BFR has one or more adjacencies in the BIFT. Clearing these bits inhibits packets from looping when the BitStrings erroneously includes a forwarding loop. When a `forward_connected()` adjacency has the "DoNotClear" (DNC) flag set, then this BP is re-set for the packet copied to that adjacency. See Section 4.2.1.

3.4. The Routing Underlay

For `forward_connected()` adjacencies, BIER-TE is sending BIER packets to directly connected BIER-TE neighbors as L2 (unicasted) BIER packets without requiring a routing underlay. For `forward_routed()` adjacencies, BIER-TE forwarding encapsulates a copy of the BIER packet so that it can be delivered by the forwarding plane of the routing underlay to the routable destination address indicated in the adjacency. See Section 4.2.2 for the adjacency definition.

BIER relies on the routing underlay to calculate paths towards BFRs and derive next-hop BFR adjacencies for those paths. This commonly relies on BIER specific extensions to the routing protocols of the routing underlay but may also be established by a controller. In BIER-TE, the next-hops of a packet are determined by the BitString through the BIER-TE Controller established adjacencies on the BFR for the BPs of the BitString. There is thus no need for BFR specific routing underlay extensions to forward BIER packets with BIER-TE semantics.

Encapsulation parameters can be provisioned by the BIER-TE controller into the `forward_connected()` or `forward_routed()` adjacencies directly without relying on a routing underlay.

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, e.g. from BFD. Providing these notifications is considered to be part of the routing underlay in this document.

3.5. Traffic Engineering Considerations

Traffic Engineering ([I-D.ietf-teas-rfc3272bis]) provides performance optimization of operational IP networks while utilizing network resources economically and reliably. The key elements needed to effect TE are policy, path steering and resource management. These elements require support at the control/controller level and within the forwarding plane.

Policy decisions are made within the BIER-TE control plane, i.e., within BIER-TE Controllers. Controllers use policy when composing BitStrings and BFR BIFT state. The mapping of user/IP traffic to specific BitStrings/BIER-TE flows is made based on policy. The specific details of BIER-TE policies and how a controller uses them are out of scope of this document.

Path steering is supported via the definition of a BitString. BitStrings used in BIER-TE are composed based on policy and resource management considerations. For example, when composing BIER-TE BitStrings, a Controller must take into account the resources available at each BFR and for each BP when it is providing congestion-loss-free services such as Rate Controlled Service Disciplines [RCSD94]. Resource availability could be provided for example via routing protocol information, but may also be obtained via a BIER-TE control protocol such as NETCONF or any other protocol commonly used by a Controller to understand the resources of the network it operates on. The resource usage of the BIER-TE traffic admitted by the BIER-TE controller can be solely tracked on the BIER-TE Controller based on local accounting as long as no `forward_routed()` adjacencies are used (see Section 4.2.1 for the definition of `forward_routed()` adjacencies). When `forward_routed()` adjacencies are used, the paths selected by the underlying routing protocol need to be tracked as well.

Resource management has implications on the forwarding plane beyond the BIER-TE defined steering of packets. This includes allocation of buffers to guarantee the worst case requirements of admitted RCSD traffic and potentially policing and/or rate-shaping mechanisms, typically done via various forms of queuing. This level of resource control, while optional, is important in networks that wish to support congestion management policies to control or regulate the offered traffic to deliver different levels of service and alleviate congestion problems, or those networks that wish to control latencies experienced by specific traffic flows.

4. BIER-TE Forwarding

4.1. The BIER-TE Bit Index Forwarding Table (BIFT)

The BIER-TE BIFT is the equivalent to the BIER BIFT for (non-TE) BIER. It exists on every BFR running BIER-TE. For every BIER sub-domain (SD) in use for BIER-TE, it is a table as shown in Figure 4. That example BIFT assumes a BSL of 8 bit positions (BPs) in the packets BitString. As in [RFC8279] this BSL is purely used for the example and not a BIER/BIER-TE supported BSL (minimum BSL is 64).

A BIER-TE BIFT compares to a BIER BIFT as shown in [RFC8279] as follows.

In both BIER and BIER-TE, BIFT rows/entries are indexed in their respective BIER pseudocode ([RFC8279] Section 6.5) and BIER-TE pseudocode (Section 4.4) by the BIFT-index derived from the packets SI, BSL and the one bit position of the packets BitString (BP) addressing the BIFT row: $\text{BIFT-index} = \text{SI} * \text{BSL} + \text{BP} - 1$. BP within a BitString are numbered from 1 to BSL, hence the - 1 offset when converting to a BIFT-index. This document also uses the notion SI:BP to indicate BIFT rows, [RFC8279] uses the equivalent notion SI:BitString, where the BitString is filled with only the BP for the BIFT row.

In BIER, each BIFT-index addresses one BFER by its BFR-id = BIFT-index + 1 and is populated on each BFR with the next-hop "BFR Neighbor" (BFR-NBR) towards that BFER.

In BIER-TE, each BIFT-index and therefore SI:BP indicates one or more adjacencies between BFRs in the topology and is only populated with those adjacencies forwarding entries on the BFR that is the upstream for these adjacencies. The BIFT entry are empty on all other BFRs.

In BIER, each BIFT row also requires a "Forwarding Bit Mask" (F-BM) entry for BIER forwarding rules. In BIER-TE forwarding, F-BM is not required, but can be used when implementing BIER-TE on forwarding hardware derived from BIER forwarding, that must use F-BM. This is discussed in the first BIER-TE forwarding pseudocode in Section 4.4.

BIFT-index (SI:BP)	(FBM)	Adjacencies: <empty> or one or more per entry
BIFT indices for Packets with SI=0		
0 (0:1)	...	forward_connected(interface,neighbor{,DNC})
1 (0:2)	...	forward_connected(interface,neighbor{,DNC})
	...	forward_connected(interface,neighbor{,DNC})
...
4 (0:5)	...	local_decap({VRF})
5 (0:6)	...	forward_routed({VRF},l3-neighbor)
6 (0:7)	...	<empty>
7 (0:8)	...	ECMP((adjacency1,...adjacencyN){,seed})
BIFT indices for BitString/Packet with SI=1		
9 (1:1)
...

BIER-TE Bit Index Forwarding Table (BIFT)

Figure 4: BIER-TE BIFT with different adjacencies

The BIFT is configured for the BIER-TE data plane of a BFR by the BIER-TE Controller through an appropriate protocol and data-model. The BIFT is then used to forward packets, according to the rules specified in the BIER-TE Forwarding Procedures.

Note that a BIFT index (SI:BP) may be populated in the BIFT of more than one BFR to save BPs. See Section 5.1.6 for an example of how a BIER-TE controller could assign BPs to (logical) adjacencies shared across multiple BFRs, Section 5.1.3 for an example of assigning the same BP to different adjacencies, and Section 5.1.9 for general guidelines regarding re-use of BPs across different adjacencies.

{VRF} indicates the Virtual Routing and Forwarding context into which the BIER payload is to be delivered. This is optional and depends on the multicast flow overlay.

4.2. Adjacency Types

4.2.1. 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 "DoNotClear" (DNC) set in the BIFT MUST NOT have the bit position for that adjacency cleared when the BFR creates a copy for it. The bit position will still be cleared for copies of the packet made towards other adjacencies. This can be used for example in ring topologies as explained in Section 5.1.6.

For protection against loops from misconfiguration (see Section 5.2.1), DNC is only permissible for forward_connected() adjacencies. No need or benefit of DNC for other type of adjacencies was identified and their risk was not analyzed.

4.2.2. Forward Routed

A "forward_routed()" adjacency is an adjacency towards a BFR that uses a (tunneling) encapsulation which will cause the packet to be forwarded by the routing underlay toward the adjacent BFR. This can leverage any feasible encapsulation, such as MPLS or tunneling over IP/IPv6, as long as the BIER-TE packet can be identified as a payload. This identification can either rely on the BIER/BIER-TE co-existence mechanisms described in Section 4.3, or by explicit support for a BIER-TE payload type in the tunneling encapsulation.

forward_routed() adjacencies are necessary to pass BIER-TE traffic across non BIER-TE capable routers or to minimize the number of required BP by tunneling over (BIER-TE capable) routers on which neither replication nor path-steering is desired, or simply to leverage path redundancy and FRR of the routing underlay towards the next BFR. They may also be useful to a multi-subnet adjacent BFR to leverage the routing underlay ECMP independent of BIER-TE ECMP (Section 4.2.3).

4.2.3. ECMP

(non-TE) BIER ECMP is tied to the BIER BIFT processing semantic and is therefore not directly usable with BIER-TE.

A BIER-TE "Equal Cost Multipath" (ECMP()) adjacency as shown in Figure 4 for BIFT-index 7 has a list of two or more non-ECMP adjacencies as parameters and an optional seed parameter. When a BIER-TE packet is copied onto such an ECMP() adjacency, an implementation specific so-called hash function will select one out

of the list's adjacencies to which the packet is forwarded. If the packet's encapsulation contains an entropy field, the entropy field SHOULD be respected; two packets with the same value of the entropy field SHOULD be sent on the same adjacency. The seed parameter allows to design hash functions that are easy to implement at high speed without running into polarization issues across multiple consecutive ECMP hops. See Section 5.1.7 for more explanations.

4.2.4. Local Decapsulation

A "local_decap()" adjacency passes a copy of the payload of the BIER-TE packet to the protocol ("NextProto") within the BFR (IPv4/IPv6, Ethernet,...) responsible for that payload according to the packet header fields. A local_decap() adjacency turns the BFR into a BFER for matching packets. Local_decap() adjacencies require the BFER to support routing or switching for NextProto to determine how to further process the packet.

4.3. Encapsulation / Co-existence with BIER

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

Like [RFC8279], handling of "Maximum Transmission Unit" (MTU) limitations is outside the scope of this document and instead part of the BIER-TE packet encapsulation and/or flow overlay. See for example [RFC8296], Section 3. It applies equally to BIER-TE as it does to BIER.

Because a BFR needs to interpret the BitString of a BIER-TE packet differently from a (non-TE) BIER packet, it is necessary to distinguish BIER from BIER-TE packets. In the BIER encapsulation [RFC8296], the BIFT-id field of the packet indicates the BIFT of the packet. BIER and BIER-TE can therefore be run simultaneously, when the BIFT-id address space is shared across BIER BIFT and BIER-TE BIFT. Partitioning the BIFT-id address space is subject to BIER-TE/BIER control plane procedures.

When [RFC8296] is used for BIER with MPLS, BIFT-id address ranges can be dynamically allocated from MPLS label space only for the set of actually used SD:BSL BIFT. This allows to also allocate non-overlapping label ranges for BIFT-id that are to be used with BIER-TE BIFTs.

With MPLS, it is also possible to reuse the same SD space for both BIER-TE and BIER, so that the same SD has both a BIER BIFT with a corresponding range of BIFT-ids and disjoint BIER-TE BIFTs with a non-overlapping range of BIFT-ids.

When a fixed mapping from BSL, SD and SI to BIFT-id is used which does not explicitly partition the BIFT-id space between BIER and BIER-TE, such as proposed for non-MPLS forwarding with [RFC8296] encapsulation in [I-D.ietf-bier-non-mpls-bift-encoding] revision 04, section 5, then it is necessary to allocate disjoint SDs to BIER and BIER-TE BIFTs so that both can be addressed by the BIFT-ids. The encoding proposed in section 6. of the same document does not statically encode BSL or SD into the BIFT-id, but allows for a mapping, and hence could provide for the same freedom as when MPLS is being used (same or different SD for BIER/BIER-TE).

forward_routed() requires an encapsulation that permits to direct unicast encapsulated 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 the (BSL,SD,SI) BitString. With non-MPLS encapsulation, some form of IP encapsulation would be required (for example IP/GRE).

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

4.4. BIER-TE Forwarding Pseudocode

The following pseudocode, Figure 5, for BIER-TE forwarding is based on the (non-TE) BIER forwarding pseudocode of [RFC8279], section 6.5 with one modification.

```
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;                                [3]
        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 5: BIER-TE Forwarding Pseudocode for required functions,
based on BIER Pseudocode

In step [2], the F-BM is used to clear bit(s) in PacketCopy. This step exists in both BIER and BIER-TE, but the F-BMs need to be populated differently for BIER-TE than for BIER for the desired clearing.

In BIER, multiple bits of a BitString can have the same BFR-NBR. When a received packets BitString has more than one of those bits set, the BIER replication logic has to avoid that more than one PacketCopy is sent to that BFR-NBR ([1]). Likewise, the PacketCopy sent to a BFR-NBR must clear all bits in its BitString that are not routed across BFR-NBR. This protects against BIER replication on any possible further BFR to create duplicates ([2]).

To solve both [1] and [2] for BIER, the F-BM of each bit index needs to have all bits set that this BFR wants to route across BFR-NBR. [2] clears all other bits in PacketCopy->BitString, and [1] clears those bits from Packet->BitString after the first PacketCopy.

In BIER-TE, a BFR-NBR in this pseudocode is an adjacency, forward_connected(), forward_routed() or local_decap(). There is no need for [2] to suppress duplicates in the way BIER does because in general, different BP would never have the same adjacency. If a BIER-TE controller actually finds some optimization in which this would be desirable, then the controller is also responsible to ensure that only one of those bits is set in any Packet->BitString, unless the controller explicitly wants for duplicates to be created.

The following points describe how the forwarding bit mask (F-BM) for each BP is configured in the BIFT and how this impacts the BitString of the packet being processed with that BIFT:

1. The F-BMs of all BIFT BPs without an adjacency have all their bits clear. This will cause [3] to skip further processing of such a BP.
2. All BIFT BPs with an adjacency (with DNC flag clear) have an F-BM that has only those BPs set for which this BFR does not have an adjacency. This causes [2] to clear all bits from PacketCopy->BitString for which this BFR does have an adjacency.
3. [1] is not performed for BIER-TE. All bit clearing required by BIER-TE is performed by [2].

This Forwarding Pseudocode can support the required BIER-TE forwarding functions (see Section 4.5), `forward_connected()`, `forward_routed()` and `local_decap()`, but not the recommended functions DNC flag and multiple adjacencies per bit nor the optional function, `ECMP()` adjacencies. The DNC flag cannot be supported when using only [1] to mask bits.

The modified and expanded Forwarding Pseudocode in Figure 6 specifies how to support all BIER-TE forwarding functions (required, recommended and optional):

- * This pseudocode eliminates per-bit F-BM, therefore reducing the size of BIFT state by $BSL^2 \cdot SI$ and eliminating the need for per-packet-copy BitString masking operations except for adjacencies with the DNC flag set:
 - `AdjacentBits[SI]` are bit positions with a non-empty list of adjacencies in this BFR BIFT. This can be computed whenever the BIER-TE Controller updates (add/removes) adjacencies in the BIFT.
 - The BFR needs to create packet copies for these adjacent bits when they are set in the packets BitString. This set of bits is calculated in `PktAdjacentBits`.
 - All bit positions to which the BFR creates copies have to be cleared in packet copies to avoid loops. This is done by masking the BitString of the packet with `~AdjacentBits[SI]`. When an adjacency has DNC set, this bit position is set again only for the packet copy towards that bit position.
- * BIFT entries may contain more than one adjacency in support of specific configurations such as Section 5.1.5. The code therefore includes a loop over these adjacencies.
- * The `ECMP()` adjacency is shown. Its parameters are a seed and a `ListOfAdjacencies` from which one is picked.
- * The `forward_connected()`, `forward_routed()`, `local_decap()` adjacencies are shown with their parameters.

```

void ForwardBitMaskPacket_withTE (Packet)
{
    SI = GetPacketSI(Packet);
    Offset = SI * BitStringLength;
    // Determine adjacent bits in the Packets BitString
    PktAdjacentBits = Packet->BitString & AdjacentBits[SI];

    // Clear adjacent bits in Packet header to avoid loops
    Packet->BitString &= ~AdjacentBits[SI];

    // Loop over PktAdjacentBits to create packet copies
    for (Index = GetFirstBitPosition(PktAdjacentBits); Index ;
        Index = GetNextBitPosition(PktAdjacentBits, Index)) {
        for adjacency in BIFT[Index+Offset]->Adjacencies {
            if(adjacency.type == ECMP(ListOfAdjacencies,seed) ) {
                I = ECMP_hash(sizeof(ListOfAdjacencies),
                               Packet->Entropy,seed);
                adjacency = ListOfAdjacencies[I];
            }
            PacketCopy = Copy(Packet);
            switch(adjacency.type) {
                case forward_connected(interface,neighbor,DNC):
                    if(DNC)
                        PacketCopy->BitString |= 1<<(Index-1);
                    SendToL2Unicast(PacketCopy,interface,neighbor);

                case forward_routed({VRF},l3-neighbor):
                    SendToL3(PacketCopy,{VRF},l3-neighbor);

                case local_decap({VRF},neighbor):
                    DecapBierHeader(PacketCopy);
                    PassTo(PacketCopy,{VRF},Packet->NextProto);
            }
        }
    }
}

```

Figure 6: Complete BIER-TE Forwarding Pseudocode for required, recommended and optional functions

4.5. BFR Requirements for BIER-TE forwarding

BFR that support BIER-TE and BIER MUST support configuration that enables BIER-TE instead of (non-TE) BIER forwarding rules for all BIFT of one or more BIER sub-domains. Every BP in a BIER-TE BIFT MUST support to have zero or one adjacency. BIER-TE forwarding MUST support the adjacency types `forward_connected()` with the DNC flag not set, `forward_routed()` and `local_decap()`. As explained in

Section 4.4, these required BIER-TE forwarding functions can be implemented via the same Forwarding Pseudocode as BIER forwarding except for one modification (skipping one masking with F-BM).

BIER-TE forwarding SHOULD support `forward_connected()` adjacencies with a set DNC flag, as this is highly useful to save bits in rings (see Section 5.1.6).

BIER-TE forwarding SHOULD support more than one adjacency on a bit. This allows to save bits in hub and spoke scenarios (see Section 5.1.5).

BIER-TE forwarding MAY support `ECMP()` adjacencies to save bits in ECMP scenarios, see Section 5.1.7 for an example. This is an optional requirement, because for ECMP deployments using BIER-TE one can also leverage ECMP of the routing underlay via `forwarded_routed` adjacencies and/or might prefer to have more explicit control of the path chosen via explicit BP/adjacencies for each ECMP path alternative.

5. BIER-TE Controller Operational Considerations

5.1. Bit Position Assignments

This section describes how the BIER-TE Controller can use the different BIER-TE adjacency types to define the bit positions of a BIER-TE domain.

Because the size of the BitString limits the size of the BIER-TE domain, many of the options described exist to support larger topologies with fewer bit positions.

5.1.1. P2P Links

On a P2P link that connects two BFRs, the same bit position can be used on both BFRs for the adjacency to the neighboring BFR. A P2P link requires therefore only one bit position.

5.1.2. BFER

Every non-Leaf BFER is given a unique bit position with a `local_decap()` adjacency.

5.1.3. Leaf BFERs

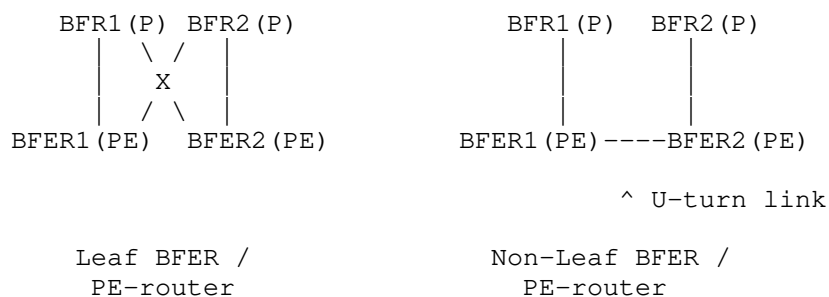


Figure 7: Leaf vs. non-Leaf BFER Example

A leaf BFER is one 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 Provider Edge (PE) router are spokes connected to Provider (P) routers, those PEs are Leaf BFERs unless there is a U-turn between two PEs.

Consider how redundant disjoint traffic can reach BFER1/BFER2 in Figure 7: When BFER1/BFER2 are Non-Leaf BFER as shown on the right-hand side, one traffic copy would be forwarded to BFER1 from BFR1, but the other one could only reach BFER1 via BFER2, which makes BFER2 a non-Leaf BFER. Likewise, BFER1 is a non-Leaf BFER when forwarding traffic to BFER2. Note that the BFERs in the left-hand picture are only guaranteed to be leaf-BFER by fitting routing configuration that prohibits transit traffic to pass through the BFERs, which is commonly applied in these topologies.

In most situations, leaf-BFER that are to be addressed via the same BitString can share a single bit position for their `local_decap()` adjacency in that BitString and therefore save bit positions. On a non-leaf BFER, a received BIER-TE packet may only need to transit the BFER or it may need to also be decapsulated. Whether or not to decapsulate the packet therefore needs to be indicated by a unique bit position populated only on the BIFT of this BFER with a `local_decap()` adjacency. On a leaf-BFER, packets never need to pass through; any packet received is therefore usually intended to be decapsulated. This can be expressed by a single, shared bit position that is populated with a `local_decap()` adjacency on all leaf-BFER addressed by the BitString.

The possible exception from this leaf-BFER bit position optimization can be cases where the bit position on the prior BIER-TE BFR (which created the packet copy for the leaf-BFER in question) is populated with multiple adjacencies as an optimization, such as in Section 5.1.4 or Section 5.1.5. With either of these two optimizations, the sender of the packet could only control explicitly

whether the packet was to be decapsulated on the leaf-BFER in question, if the leaf-BFER has a unique bit position for its `local_decap()` adjacency.

However, if the bit position is shared across leaf-BFER, and packets are therefore decapsulated potentially unnecessarily, this may still be appropriate if the decapsulated payload of the BIER-TE packet indicates whether or not the packet needs to be further processed/received. This is typically true for example if the payload is IP multicast because IP multicast on a BFER would know the membership state of the IP multicast payload and be able to discard it if the packet was delivered unnecessarily by the BIER-TE layer. If the payload has no such membership indication, and the BFIR wants to have explicit control about which BFER are to receive and decapsulate a packet, then these two optimizations can not be used together with shared bit positions optimization for leaf-BFER.

5.1.4. LANs

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

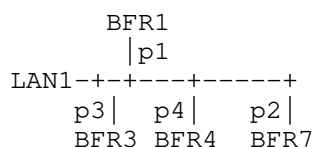


Figure 8: LAN Example

If Bandwidth on the LAN is not an issue and most BIER-TE traffic should be copied to all neighbors on a LAN, then bit positions can be saved by assigning just a single bit position to the LAN and populating the bit position 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 case of BFRs redundantly connected to more than one LAN with this optimization because these BFRs would receive duplicates and forward those duplicates into the opposite LANs. Adjacencies of such BFRs into their LAN still need a separate bit position.

5.1.5. Hub and Spoke

In a setup with a hub and multiple spokes connected via separate p2p links to the hub, all p2p adjacencies from the hub to the spokes links can share the same bit position. The bit position on the hub's BIFT is set up with a list of `forward_connected()` adjacencies, one for each Spoke.

This option is similar to the bit position optimization in LANs: Redundantly connected spokes need their own bit positions, unless they are themselves Leaf-BFER.

This type of optimized BP could be used for example when all traffic is "broadcast" traffic (very dense receiver set) such as live-TV or many-to-many telemetry including situation-awareness (SA). This BP optimization can then be used to explicitly steer different traffic flows across different ECMP paths in Data-Center or broadband-aggregation networks with minimal use of BPs.

5.1.6. Rings

In L3 rings, instead of assigning a single bit position for every p2p link in the ring, it is possible to save bit positions by setting the "DoNotClear" (DNC) flag on `forward_connected()` adjacencies.

For the rings shown in Figure 9, a single bit position will suffice to forward traffic entering the ring at BFRa or BFRb all the way up to BFR1:

On BFRa, BFRb, BFR30,... BFR3, the bit position is populated with a `forward_connected()` adjacency pointing to the clockwise neighbor on the ring and with DNC set. On BFR2, the adjacency also points to the clockwise neighbor BFR1, but without DNC set.

Handling DNC this way ensures that copies forwarded from any BFR in the ring to a BFR outside the ring will not have the ring bit position set, therefore minimizing the chance to create loops.

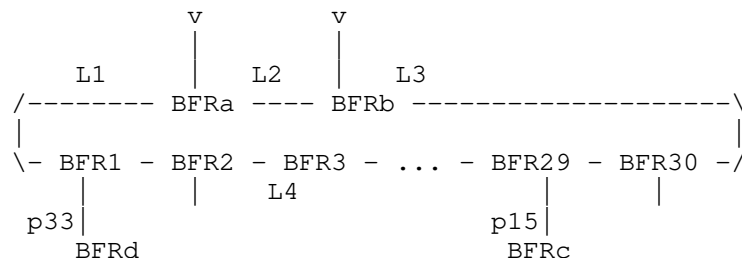


Figure 9: Ring Example

Note that this example only permits for packets intended to make it all the way around the ring to enter it 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 bit positions. One for each direction: clockwise and counterclockwise.

Both would be set up to stop rotating on the same link, e.g. L1. When the ingress ring BFR creates the clockwise copy, it will clear the counterclockwise bit position because the DNC bit only applies to the bit for which the replication is done. Likewise for the clockwise bit position for the counterclockwise copy. As a result, the ring ingress BFR will send a copy in both directions, serving BFRs on either side of the ring up to L1.

5.1.7. Equal Cost MultiPath (ECMP)

[RFC-Editor: A reviewer (Lars Eggert) noted that the infinite "to use" in the following sentence is not correct. The same was also noted for several other similar instances. The following URL seems to indicate though that this is a per-case decision, which seems undefined: <https://writingcenter.gmu.edu/guides/choosing-between-infinite-and-gerund-to-do-or-doing>. What exactly should be done about this ?].

An ECMP() adjacency allows to use just one BP to deliver packets to one of N adjacencies instead of one BP for each adjacency. In the common example case Figure 10, a link-bundle of three links L1,L2,L3 connects BFR1 and BFR2, and only one BP is used instead of three BP to deliver packets from BFR1 to BFR2.

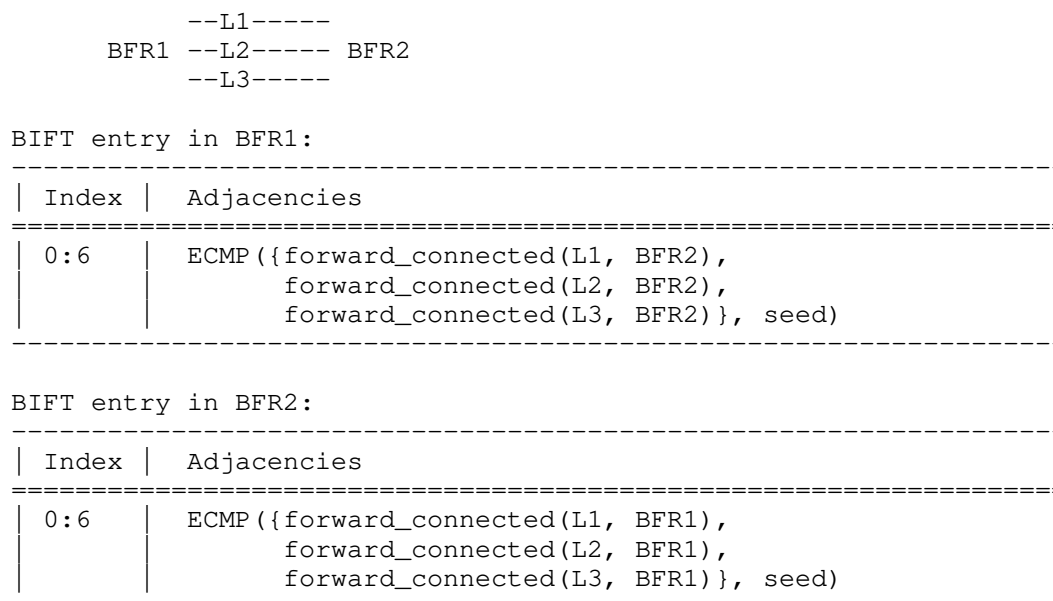


Figure 10: ECMP Example

This document does not standardize any ECMP algorithm because it is sufficient for implementations to document their freely chosen ECMP algorithm. Figure 11 shows an example ECMP algorithm, and would double as its documentation: A BIER-TE controller could determine which adjacency is chosen based on the seed and adjacencies parameters and the packet entropy.

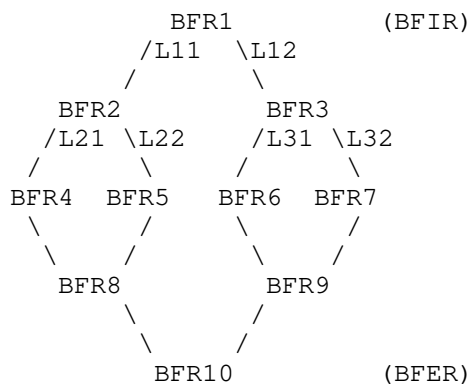
```

forward(packet, ECMP(adj(0), adj(1),... adj(N-1), seed)):
  i = (packet(bier-header-entropy) XOR seed) % N
  forward packet to adj(i)

```

Figure 11: ECMP algorithm 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, but also across alternative paths across different transit BFR, and it explains the use of the seed parameter.



BIFT entry in BFR1:

0:6	ECMP({forward_connected(L11, BFR2), forward_connected(L12, BFR3)}, seed1)
-----	--

BIFT entry in BFR2:

0:7	ECMP({forward_connected(L21, BFR4), forward_connected(L22, BFR5)}, seed1)
-----	--

BIFT entry in BFR3:

0:7	ECMP({forward_connected(L31, BFR6), forward_connected(L32, BFR7)}, seed1)
-----	--

BIFT entry in BFR4, BFR5:

0:8	forward_connected(Lxx, BFR8)	xx differs on BFR4/BFR5
-----	------------------------------	-------------------------

BIFT entry in BFR6, BFR7:

0:8	forward_connected(Lxx, BFR9)	xx differs on BFR6/BFR7
-----	------------------------------	-------------------------

BIFT entry in BFR8, BFR9:

0:9	forward_connected(Lxx, BFR10)	xx differs on BFR8/BFR9
-----	-------------------------------	-------------------------

Figure 12: Polarization Example

Note that for the following discussion of ECMP, only the BIFT ECMP adjacencies on BFR1, BFR2, BFR3 are relevant. The re-use of BP across BFR in this example is further explained in Section 5.1.9 below.

With the setup of ECMP in the topology above, 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 the list of 2 adjacencies given as parameters to the ECMP. It is link L11-to-BFR2. BFR2 performs again ECMP with two adjacencies on that subset of traffic using the same seed1, and will therefore again select the first of its two adjacencies: L21-to-BFR4. And therefore L22 and BFR5 sees no traffic. Likewise for L31 and BFR6.

This issue in BFR2/BFR3 is called polarization. It results from the re-use of the same hash function across multiple consecutive hops in topologies like these. To resolve this issue, the ECMP() adjacency on BFR1 can be set up with a different seed2 than the ECMP() adjacencies on BFR2/BFR3. BFR2/BFR3 can use the same hash because packets will not sequentially pass across both of them. Therefore, they can also use the same BP 0:7.

Note that ECMP solutions outside of BIER often hide the seed by auto-selecting it from local entropy such as unique local or next-hop identifiers. Allowing the BIER-TE Controller to explicitly set the seed gives the ability for it to control same/different path selection across multiple consecutive ECMP hops.

5.1.8. Forward Routed adjacencies

5.1.8.1. Reducing bit positions

Forward_routed() adjacencies can reduce the number of bit positions required when the path steering requirement is not hop-by-hop explicit path selection, but loose-hop selection. Forward_routed() adjacencies can also allow to operate BIER-TE across intermediate hop routers that do not support BIER-TE.

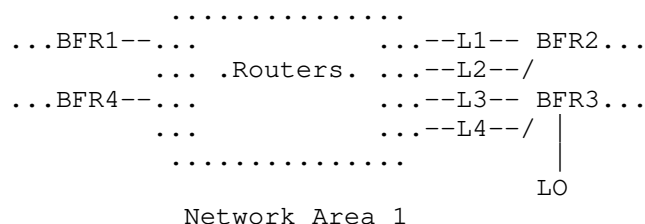


Figure 13: Forward Routed Adjacencies Example

Assume the requirement in Figure 13 is to explicitly steer traffic flows that have arrived at BFR1 or BFR4 via a path in the routing underlay "Network Area 1" to one of the following three next segments: (1) BFR2 via link L1, (2) BFR2 via link L2, or (3) via BFR3 and then not caring whether the packet is forwarded via L3 or L4.

To enable this, both BFR1 and BFR4 are set up with a `forward_routed` adjacency bit position towards an address of BFR2 on link L1, another `forward_routed()` bit position towards an address of BFR2 on link L2 and a third `forward_routed()` bit position towards a node address L0 of BFR3.

5.1.8.2. Supporting nodes without BIER-TE

`Forward_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.

5.1.9. Reuse of bit positions (without DNC)

Bit positions can be re-used across multiple BFRs to minimize the number of BP needed. This happens when adjacencies on multiple BFRs use the DNC flag as described above, but it can also be done for non-DNC adjacencies. This section only discusses this non-DNC case.

Because BP are cleared when passing a BFR with an adjacency for that BP, reuse of BP across multiple BFRs does not introduce any problems with duplicates or loops that do not also exist when every adjacency has a unique BP. Instead, the challenge when reusing BP is whether it allows to still achieve the desired Tree Engineering goals.

BP cannot be reused across two BFRs that would need to be passed sequentially for some path: The first BFR will clear the BP, so those paths cannot be built. BP can be set across BFR that would (A) only occur across different paths or (B) across different branches of the same tree.

An example of (A) was given in Figure 12, where BP 0:7, BP 0:8 and BP 0:9 are each reused across multiple BFRs because a single packet/path would never be able to reach more than one BFR sharing the same BP.

Assume the example was changed: BFR1 has no `ECMP()` adjacency for BP 0:6, but instead BP 0:5 with `forward_connected()` to BFR2 and BP 0:6 with `forward_connected()` to BFR3. Packets with both BP 0:5 and BP

0:6 would now be able to reach both BFR2 and BFR3 and the still existing re-use of BP 0:7 between BFR2 and BFR3 is a case of (B) where reuse of BP is perfect because it does not limit the set of useful path choices:

If instead of reusing BP 0:7, BFR3 used a separate BP 0:10 for its ECMP() adjacency, no useful additional path steering options would be enabled. If duplicates at BFR10 where undesirable, this would be done by not setting BP 0:5 and BP 0:6 for the same packet. If the duplicates where desirable (e.g.: resilient transmission), the additional BP 0:10 would also not render additional value.

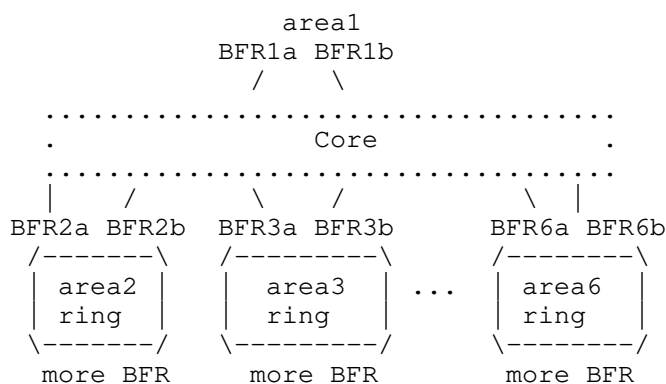


Figure 14: Reuse of BP

Reuse may also save BPs in larger topologies. Consider the topology shown in Figure 14. A BFIR/sender (e.g.: video headend) is attached to area 1, and area 2...6 contain receivers/BFER. Assume each area had a distribution ring, each with two BPs to indicate the direction (as explained before). These two BPs could be reused across the 5 areas. Packets would be replicated through other BPs for the Core to the desired subset of areas, and once a packet copy reaches the ring of the area, the two ring BPs come into play. This reuse is a case of (B), but it limits the topology choices: Packets can only flow around the same direction in the rings of all areas. This may or may not be acceptable based on the desired path steering options: If resilient transmission is the path engineering goal, then it is likely a good optimization, if the bandwidth of each ring was to be optimized separately, it would not be a good limitation.

5.1.10. Summary of BP optimizations

This section reviewed a range of techniques by which a BIER-TE Controller can create a BIER-TE topology in a way that minimizes the number of necessary BPs.

Without any optimization, a BIER-TE Controller would attempt to map the network subnet topology 1:1 into the BIER-TE topology and every subnet adjacent neighbor requires a `forward_connected()` BP and every BFER requires a `local_decap()` BP.

The optimizations described are then as follows:

- * P2P links require only one BP (Section 5.1.1).
- * All leaf-BFER can share a single `local_decap()` BP (Section 5.1.3).
- * A LAN with N BFR needs at most N BP (one for each BFR). It only needs one BP for all those BFR that are not redundantly connected to multiple LANs (Section 5.1.4).
- * A hub with p2p connections to multiple non-leaf-BFER spokes can share one BP to all spokes if traffic can be flooded to all spokes, e.g.: because of no bandwidth concerns or dense receiver sets (Section 5.1.5).
- * Rings of BFR can be built with just two BP (one for each direction) except for BFR with multiple ring connections - similar to LANs (Section 5.1.6).
- * ECMP() adjacencies to N neighbors can replace N BP with 1 BP. Multihop ECMP can avoid polarization through different seeds of the ECMP algorithm (Section 5.1.7).
- * `Forward_routed()` adjacencies allow to "tunnel" across non-BIER-TE capable routers and across BIER-TE capable routers where no traffic-steering or replications are required (Section 5.1.8).
- * BP can generally be reused across a set of nodes where it can be guaranteed that no path will ever need to traverse more than one node of the set. Depending on scenario, this may limit the feasible path steering options (Section 5.1.9).

Note that the described list of optimizations is not exhaustive. Especially when the set of required path steering choices is limited and the set of possible subsets of BFERs that should be able to receive traffic is limited, further optimizations of BP are possible. The hub and spoke optimization is a simple example of such traffic pattern dependent optimizations.

5.2. Avoiding duplicates and loops

5.2.1. Loops

Whenever BIER-TE creates a copy of a packet, the BitString of that copy will have all bit positions cleared that are associated with adjacencies on the BFR. This inhibits looping of packets. The only exception are adjacencies with DNC set.

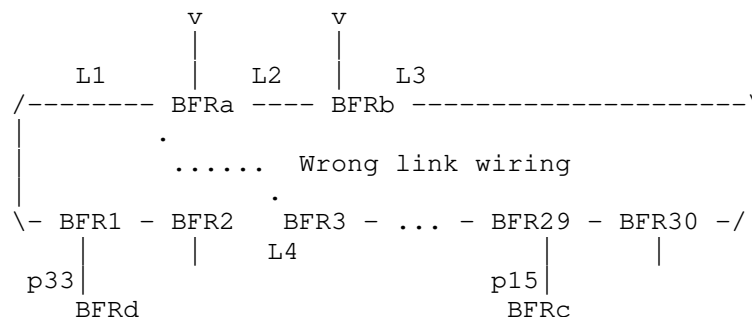


Figure 15: Miswired Ring Example

With DNC set, looping can happen. Consider in Figure 15 that link L4 from BFR3 is (inadvertently) plugged into the L1 interface of BFRa (instead of BFR2). This creates a loop where the rings clockwise bit position is never cleared 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 DNC set, and the link layer port unique unicast destination address of the adjacency (e.g. MAC address) protects against closing the loop. Link layers without port unique link layer addresses should not be used with the DNC flag set.

5.2.2. Duplicates

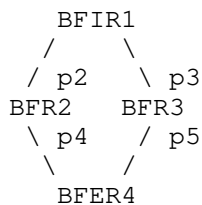


Figure 16: Duplicates Example

Duplicates happen when the graph expressed by a BitString is not a tree but redundantly connecting BFRs with each other. In Figure 16, a BitString of p2,p3,p4,p5 would result in duplicate packets to arrive on BFER4. The BIER-TE Controller must therefore ensure to only create BitStrings that are trees.

When links are incorrectly physically re-connected before the BIER-TE 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 BIER-TE Controller.

5.3. Managing SI, sub-domains and BFR-ids

When the number of bits required to represent the necessary hops in the topology and BFER exceeds the supported BitStringLength (BSL), multiple SIs and/or sub-domains 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-ids can be assigned to BFIR/BFER for BIER-TE.

5.3.1. Why SI and sub-domains

For (non-TE) 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 BFERs in different SIs or sub-domains require different BIER packets: each one with a BitString for a different (SI,sub-domain) combination. Each such BitString uses one BSL sized SI block in the BIFT of the sub-domain. We call this a BIFT:SI (block).

For BIER and BIER-TE forwarding themselves there is also no difference whether different SIs and/or sub-domains are chosen, but SI and sub-domain 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 sub-domain 0 (unless there are > 64k BFIR/BFER).

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

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

Even if creation of sub-domains and assignment of BFR-id to BFIR/BFER in those sub-domains is automated, it is not expected that individual service instances can deal with BFER in different sub-domains. A service instance may only support configuration of a single sub-domain 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 sub-domain 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 SIs as necessary (see Section 5.3.2). Different services may use different sub-domains that primarily exist to provide more efficient replication (and for BIER-TE desirable path steering) for different subsets of BFIR/BFER.

5.3.2. Assigning bits for the BIER-TE topology

In BIER, BitStrings only need to carry bits for BFERs, which leads 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 path steering across every single hop (which requires more bits), or reducing the number of required bits by exploiting optimizations such as unicast (`forward_routed()`), ECMP() or flood (DNC) over "uninteresting" sub-parts of the topology - e.g. parts where different trees do not need to take different paths due to path steering reasons.

The total number of bits to describe the topology vs. the number of BFERs in a BIFT:SI can range widely based on the size of the topology and the amount of alternative paths in it. In a BIER-TE topology crafted by a BIER-TE expert, the higher the percentage of non-BFER bits, the higher the likelihood, that those topology bits are not just BIER-TE overhead without additional benefit, but instead that they will allow to express desirable path steering alternatives.

5.3.3. Assigning BFR-id with BIER-TE

BIER-TE forwarding does not use the BFR-id, nor does it require for the BFIR-id field of the BIER header to be set to a particular value. However, other parts of a BIER-TE deployment may need a BFR-id, specifically multicast flow overlay signaling and multicast flow overlay packet disposition, and in that case BFRs need to also have BFR-ids for BIER-TE SDs.

For example, for BIER overlay signaling, BFIRs need to have a BFR-id, because this BFIR BFR-id is carried in the BFIR-id field of the BIER header to indicate to the overlay signaling on the receiving BFER which BFIR originated the packet.

In BIER, $\text{BFR-id} = \text{SI} * \text{BSL} + \text{BP}$, such that the SI and BP of a BFER can be calculated from the BFR-id and vice versa. This also means that every BFR with a BFR-id has a reserved BP in an SI, even if that is not necessary for BIER forwarding, because the BFR may never be a BFER but only a BFIR.

In BIER-TE, for a non-leaf BFER, there is usually a single BP for that BFER with a `local_decap()` adjacency on the BFER. The BFR-id for such a BFER can therefore be determined using the same procedure as in (non-TE) BIER: $\text{BFR-id} = \text{SI} * \text{BSL} + \text{BP}$.

As explained in Section 5.1.3, leaf BFERs do not need such a unique `local_decap()` adjacency. Likewise, BFIRs that are not also BFERs may not have a unique `local_decap()` adjacency either. For all those BFIRs and (leaf) BFERs, the controller needs to determine unique BFR-ids that do not collide with the BFR-ids derived from the non-leaf BFER `local_decap()` BPs.

While this document defines no requirements on how to allocate such BFR-id, a simple option is to derive it from the (SI,BP) of an adjacency that is unique to the BFR in question. For a BFIR this can be the first adjacency only populated on this BFIR, for a leaf-BFER, this could be the first BP with an adjacency towards that BFER.

5.3.4. Mapping from BFR to BitStrings with BIER-TE

In BIER, applications of the flow overlay on a BFIR can calculate the (SI,BP) of a BFER from the BFR-id of the BFER and can therefore easily determine the BitStrings for a BIER packet to a set of BFERs with known BFR-ids.

In BIER-TE this mapping needs to be equally supported for flow overlays. This section outlines two core options, based on what type of Tree Engineering the BIER-TE controller needs to perform for a particular application.

"Independent branches": For a given flow overlay instance, the branches from a BFIR to every BFER are calculated by the BIER-TE controller to be independent of the branches to any other BFER. Shortest path trees are the most common examples of trees with independent branches.

"Interdependent branches": When a BFER is added or deleted from a particular distribution tree, the BIER-TE controller has to recalculate the branches to other BFER, because they may need to change. Steiner trees are examples of interdependent branch trees.

If "independent branches" are used, the BIER-TE Controller can signal to the BFIR flow overlay for every BFER an SI:BitString that represents the branch to that BFER. The flow overlay on the BFIR can then independently of the controller calculate the SI:BitString for all desired BFERs by OR'ing their BitStrings. This allows for flow overlay applications to operate independently of the controller whenever it needs to determine which subset of BFERs need to receive a particular packet.

If "interdependent branches" are required, the application would need to inquire the SI:BitString for a given set of BFER whenever the set changes.

Note that in either case (unlike in BIER), the bits may need to change upon link/node failure/recovery, network expansion and network resource consumption by other traffic as part of traffic engineering goals (e.g.: re-optimization of lower priority traffic flows). Interactions between such BFIR applications and the BIER-TE Controller do therefore need to support dynamic updates to the SI:BitStrings.

Communications between the BFIR flow overlay and the BIER-TE controller requires some way to identify the BFER. If BFR-ids are used in the deployment, as outlined in Section 5.3.3, then those are the natural BFR identifier. If BFR-ids are not used, then any other unique identifier, such as the BFR-prefix of the BFR ([RFC8279]) could be used.

5.3.5. Assigning BFR-ids for BIER-TE

It is not currently determined if a single sub-domain could or should be allowed to forward both (non-TE) 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 sub-domain. 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 its BIER-TE BFR-id.
- B. BIER and BIER-TE share the same BFR-id. The BFR-ids 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.

5.3.6. Example bit allocations

5.3.6.1. With BIER

Consider a network setup with a BSL of 256 for a network topology as shown in Figure 17. The network has 6 areas, each with 170 BFERs, connecting via a core with 4 (core) BFRs. To address all BFERs with BIER, 4 SIs 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-ids are allocated to BFER in the network, but for efficiency further down in the network it does make a difference.

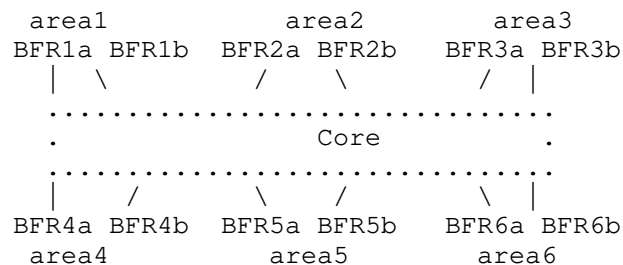


Figure 17: Scaling BIER-TE bits by reuse

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 SIs 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-ids are allocated intelligently, then all the BFER in an area would be given BFR-id with as few as possible different SIs. 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 then also go down over time when BFR-ids are only allocated sequentially, network wide. An area that initially only has BFR-id in one SI might end up with many SIs 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 BFERs after network expansion. In this example one may consider to use 6 SIs and assign one to each area.

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

5.3.6.2. With BIER-TE

In BIER-TE one needs to determine a subset of the physical topology and attached BFERs 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 BFERs, 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, one assigns random subsets of BFERs to different SIs. This will result in an outcome much worse than in (non-TE) BIER: It maximizes the amount of unnecessary topology overlap across SI and therefore reduces the number of BFER that can be reached across each individual SI. Intelligent BFER to SI assignment and selecting specific "desired" subtopologies can minimize this problem.

To set up BIER-TE efficiently for the topology of Figure 17, the following bit allocation method can be used. This method can easily be expanded to other, similarly structured larger topologies.

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

In addition, we use 4 bits in each SI: bia, bib, bea, beb: (b)it (i)ngress (a), (b)it (i)ngress (b), (b)it (e)gress (a), (b)it (e)gress (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 BFIRs in an area $j | j=1...6$, 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= $k | k=1...6$ is populated with forward_routed(BFRka) and beb in BIFT:SI= k with forward_routed(BFRkb).

For BIER-TE forwarding of a packet to a subset of BFERs 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 steering 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.

5.3.7. Summary

BIER-TE can, like BIER, support multiple SIs within a sub-domain. This allows to apply the mapping $\text{BFR-id} = \text{SI} * \text{BSL} + \text{BP}$. This allows to re-use the BIER architecture concept of BFR-id and therefore minimize BIER-TE specific functions in possible BIER layer control plane mechanisms with BIER-TE, including flow overlay methods and BIER header fields.

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

Sub-domains (SDs) in BIER-TE can be used like in BIER to create more efficient replication to known subsets of BFERs.

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

6. Security Considerations

If [RFC8296] is used, BIER-TE shares its security considerations.

BIER-TE shares the security considerations of BIER, [RFC8279], with the following overriding or additional considerations.

BIER-TE forwarding explicitly supports unicast "tunneling" of BIER packets via `forward_routed()` adjacencies. The BIER domain security model is based on a subset of interfaces on a BFR that connect to other BFRs of the same BIER domain. For BIER-TE, this security model equally applies to such unicast "tunneled" BIER packets. This does not only include the need to filter received unicast "tunneled" BIER packets to prohibit injection of such "tunneled" BIER packets from outside the BIER domain, but also prohibiting `forward_routed()` adjacencies to leak BIER packets from the BIER domain. It SHOULD be possible to configure interfaces to be part of a BIER domain solely for sending and receiving of unicast "tunneled" BIER packets even if the interface can not send/receive BIER encapsulated packets.

In BIER, the standardized methods for the routing underlays are IGPs with extensions to distribute BFR-ids and BFR-prefixes. [RFC8401] specifies the extensions for IS-IS and [RFC8444] specifies the extensions for OSPF. Attacking the protocols for the BIER routing underlay or (non-TE) BIER layer control plane, or impairment of any BFR in a domain may lead to successful attacks against the results of the routing protocol, enabling DoS attacks against paths or the addressing (BFR-id, BFR-prefixes) used by BIER.

The reference model for the BIER-TE layer control plane is a BIER-TE controller. When such a controller is used, impairment of an individual BFR in a domain causes no impairment of the BIER-TE control plane on other BFRs. If a routing protocol is used to support `forward_routed()` adjacencies, then this is still an attack vector as in BIER, but only for BIER-TE `forward_routed()` adjacencies, and not other adjacencies.

Whereas IGP routing protocols are most often not well secured through cryptographic authentication and confidentiality, communications between controllers and routers such as those to be considered for the BIER-TE controller/control-plane can be and are much more commonly secured with those security properties, for example by using Secure Shell (SSH), [RFC4253] for NETCONF ([RFC6242]), or via Transport Layer Security (TLS), such as [RFC8253] for PCEP, [RFC5440], or [RFC7589] for NETCONF. BIER-TE controllers SHOULD use security equal to or better than these mechanisms.

When any of these security mechanisms/protocols are used for communications between a BIER-TE controller and BFRs, their security considerations apply to BIER-TE. In addition, the security considerations of PCE, [RFC4655] apply.

The most important attack vector in BIER-TE is misconfiguration, either on the BFR themselves or via the BIER-TE controller. Forwarding entries with DNC could be set up to create persistent loops, in which packets only expire because of TTL. To minimize the impact of such attacks (or more likely unintentional misconfiguration by operators and/or bad BIER-TE controller software), the BIER-TE forwarding rules are defined to be as strict in clearing bits as possible. The clearing of all bits with an adjacency on a BFR prohibits that a looping packet creates additional packet amplification through the misconfigured loop on the packet's second or further times around the loop, because all relevant adjacency bits would have been cleared on the first round through the loop. In result, BIER-TE has the same degree of looping packets as possible with unintentional or malicious loops in the routing underlay with BIER or even with unicast traffic.

Deployments where BIER-TE would likely be beneficial may include operational models where actual configuration changes from the controller are only required during non-production phases of the network's life-cycle, such as in embedded networks or in manufacturing networks during e.g. plant reworking/repairs. In these type of deployments, configuration changes could be locked out when the network is in production state and could only be (re-)enabled through reverting the network/installation into non-production state. Such security designs would not only allow to provide additional layers of protection against configuration attacks, but would foremost protect the active production process from such configuration attacks.

7. IANA Considerations

This document requests no action by IANA.

8. Acknowledgements

The authors would like to thank Greg Shepherd, Ijsbrand Wijnands, Neale Ranns, Dirk Trossen, Sandy Zheng, Lou Berger, Jeffrey Zhang, Carsten Borman and Wolfgang Braun for their reviews and suggestions.

Special thanks to Xuesong Geng for shepherding the document and for IESG review/suggestions by Alvaro Retana (responsible AD/RTG), Benjamin Kaduk (SEC), Tommy Pauly (TSV), Zaheduzzaman Sarker (TSV), Eric Vyncke (INT), Martin Vigoureux (RTG), Robert Wilton (OPS), Eric

Kline (INT), Lars Eggert (GEN), Roman Danyliv (SEC), Ines Robles (RTGDIR), Robert Sparks (Gen-ART), Yingzhen Qu (RTGdir), Martin Duke (TSV).

9. Change log [RFC Editor: Please remove]

draft-ietf-bier-te-arch:

13:

Changed Gregs author association/email.

Fixed Nits in -12 from Ben Kaduk.

Fixed Alvaro's concerns: (1) Removed references to SR in Abstract/Overview (2) removed section 4.5.

12:

AD review Alvaro Retana.

Various textual/editorial nits including adding () to all instances of forwarding adjacency name instances.

3.1 Added new paragraph outlining possible use of BGP as RR in BIER-TE controller as core of multicast flow overlay component of BIER-TE.

3.2 added xref's to relevant sections to the listed control plane points.

4.1 rewrote paragraphs of 4.1 leading up to Figure 4. to eliminate any confusion in how the BIFT work and how it compares to the notions in rfc8279, as well as better linking it to the Pseudocode.

Moved SR section into appendix.

TSV review Martin Duke.

Text/editorial nits.

4.4 improved text describing handling of F-BM.

RTGdir review Yingzhen Qu.

Various text/editorial nits.

Added notion that BitStrings represent loop free tree for packet to abstract and intro.

Various text nit and editorial improvements.

Fixed some BFR-id field -> BFIR-id field mistakes.

Capitalized NETCONF/RESTCONF/YANG, added RFC references.

Improved Figure 16 with explicitly two links into BFR3 and explanatory text.

Gen-ART review Robert Sparks.

Various textual nits, editorial improvements.

3.2 Introduced terms "BIER-TE topology control" and "BIER-TE tree control" for the two functional components of the control plane.

3.2.1 - 3.2 change introduces the open RFC-editor issue of appropriate xrfs (to be resolved by RFC-editor).

3.3 Rewrote last paragraph to better describe loop prevention through clearing of bits in BitString.

4.1 Fixed up text/formula describing mapping between bfr-id, SI:BP and SI,BSL and BP. Fix offset bug.

5.3.6.2 Improved description paragraph explaining overlap of topology for different SI.

5.3.7 Improved first summary paragraph.

7. Rephrased applicability statement of control plane protocol security considerations to BIER-TE security.

RTGDIR review Ines Robles.

Fixed up adjacencies in Example 2 and explanation text to be explicit about which BFR not only passes, but also receives the packet.

7. (security considerations). Added paragraph about forward_routed() and prohibiting BIER packet leaking in/out of domain.

IESG review Roman Danyliv (SEC).

Several textual/sentence nits/editorials.

IESG review Lars Eggert (GEN).

Various good editorial word fixed.

Pointer to non-false-positive bloom filter work that looks like it happened after our IETF discussions documented in this doc, so will not add it to doc, but here is URL for folks interested: <https://ieeexplore.ieee.org/document/8486415>.

Did not change "native" to a different word for inclusivity because of my worry there is no established single replacement word, making reading/searching/understanding more difficult.

IESG review Martin Vigoureux (RTG).

Added back reference to RFC8402. Textual fixes.

IESG review Eric Kline (INT).

2.1 Fixed typo in BFR* explanations.

4.3 Added explanatio about MTU handling.

IESG review Eric Vyncke (INT).

Fixed up initial text to introduce various abbreviations.

2.4 refined wording to "with the `_intent_` to easily build common forwarding planes...".

4.2.3 refined text about entropy in ECMP - now taken text from rfc8279.

IESG review Zaheduzzaman Sarker (TSV).

5.1.7 Refined text explaining documentation of ECMP algorithm.

5.3.6.2. fixed range of areas/SI over which to build the example large network BPs - removed explanation of the large network shown to be only used for sources in area 1 (IPTV), because it was a stale explanation.

IESG review Ben Kaduk (round 2):

4.4 Advanced pseudocode still had one wrong "~". Root cause seems to have been day 0 problem in pseudocode written for -01, "~" was inserted in the wrong one of two code lines. Also enhanced textual description and comments in pseudocode, changed variable name AdjacentBits to PktAdjacentBits to avoid confusion with AdjacentBits[SI].

5.1.3 Rewrote last two paragraphs explaining the sharing of bit positions for lead-BFER hopefully better. Also detailed how it interacts with other optimizations and the type of payload BIER-TE packets may carry.

4.4 (from Carsten Borman) changed spacing in pseudocode to be consistent. Fixed {VRF}, clarified pseudocode object syntax, typos.

11: IESG review Ben Kaduk, summary:

One discuss for bug in pseudocode. turned out to be one cahrcrter typo.

Added (non-TE) prefix in places where BIER by itsels had to be better disambiguated.

enhanced text for hub-and-spoke to indicate we're only talking about hub to spoke traffic.

long list ot language fixes/improvement (nits). Thanks a lot!.

add suggestion to SHOULD use known confidentiality protocols between controller and BFR.

10: AD review Alvaro Retana, summary:

Note: rfcdiff shows more changes than actually exist because text moved around.

Summary:

1. restructuring: merged all controller sections under common controller ops main section, moved unfitting stuff out to other parts of doc. Split Intro section into Overview and Intro. Shortened Abstract, moved text into Overview, added sections overview.
2. enhanced/rewrote: 2.3 Comparison with -> Relationship to BIER-TE

3. enhanced/rewrote: 3.2 BIER-TE controller -> BIER-TE control plane, 3.2.1 BIER-TE controller, for consistency with rfc8279
4. additional subsections for Alvaros asks
5. added to: 3.3 BIER-TE forwarding plane (consistency with rfc8279)
6. Enhanced description of 4.3/encap considerations to better explain how BIER/BIER-TE can run together.

Notation: Markers (a), (b), ... at end of points are references from the review discussion with Alvaro to the changes made.

Details:.

Throughout text: changed term spelling to rfc8279 - bit positions, sub-domain, ... (i).

Reset changed to clear, also DNR changed to DNC (Do Not Clear) (q).

Abstract: Shortened. Removed name explanation note (Tree Engineering), (a).

1. Introduction -> Overview: Moved important explanation paragraph from abstract to Introduction. Fixed text, (a).

Added bullet point list explanation of structure of document (e).

Renamed to Overview because that is now more factually correct.

1.1. Fixed bug in example adding bit p15.(l).

2. (New - Introduction): Moved section 1.1 - 1.3 (examples, comparison with BIER-TE) from Introduction into new "Overview" section. Primarily so that "requirements language" section (at end of Introduction) is not in middle of document after all the Introduction.

2.1 Removed discussion of encap, moved to 4.2.2 (m).

2.2 enhanced paragraph suggesting native/overlay topology types, also suggest type hybrid (n).

2.3 Overhauled comparison text BIER/BIER-TE, structured into common, different, not-required-by-te, integration-bier-bier-te. Changed title to "Relationship" to allow including last point. (f).

2.4 moved Hardware forwarding comparison section into section 2 to allow coalescing of sections into section 5 about the controller operations (hardware forwarding was in the middle of it, wrong place). Shortened/improved third paragraph by pointing to BIFT as deciding element for selection between BIER/BIER-TE. Removed notion of experimentation (this now targets standard) (g).

3. (Components): Aligned component name and descriptions better with RFC8279. Now describe exactly same three layers. BIER layer constituted from BIER-TE control plane and BIER-TE forwarding plane. BIER-TE controller is now simply component of BIER-TE control plane. (b).

3.1. shortened/improved paragraph explaining use of SI:BP instead of also bfr-id as index into BIFT, rewrote paragraph talking about reuse of BPs(o).

3.2. rewrote explanation of BIER-TE control plane in the style of RFC8729 Section 4.2 (BIER layer) with numbered points. Note that RFC8729 mixes control and forwarding plane bullet points (this doc does not). Merged text from old sections 2.2.1 and 2.2.3 into list. (b).

3.2.1. Expanded/improved explanation of BIER-TE Controller (b).

3.2.1.1. Added subsection for topology discovery and creation (d).

3.2.1.2. Added subsection for engineered BitStrings as key novel aspect not found in BIER. (X).

3.3. Added numbered list for components of BIER-TE forwarding plane (completing the comparable text from RFC8729 Section 4.2).

3.4 Alvaro does not mind additional example, fixed bugs.

3.5 Removed notion about using IGP BIER extensions for BIER-TE, such as BIFT address ranges. After -10 making use of BIFT clearer, it now looks to authors as if use of IGP extensions would not be beneficial, as long as we do need to use the BIER-TE controller, e.g. unlike in BIER, a BFR could not learn from the IGP information what traffic to send towards a particular BIFT-ID, but instead that is the core of what the controller needs to provide.

4.2.2 Improved text to explain requirement to identify BIER-TE in the tunnel encap and compress description of use-cases (m).

4.2.3 enhanced ECMP text (p).

4.3. rewrote most of Encapsulation Considerations to better explain to Alvaros question re sharing or not sharing SD via BIER/BIER-TE. Added reference to I-D.ietf-bier-non-mpls-bift-encoding as a very helpful example. (f).

4.3 Renamed title to "...Co-Existence with BIER" as this is what it is about and to help finding it from abstract/intro ("co-exist") (j).

4.4. Moved BIER-TE Forwarding Pseudocode here to coalesce text logically. Changed text to better compare with BIER pseudo forwarding code. Numerical list of how F-BM works for BIER-TE. Removed efficiency comparison with BIER (too difficult to provide sufficient justification, derails from focus of section) (j).

4.6. (Requirements) Restructured: Removed notion of "basic" BIER-TE forwarding, simply referring to it now as "mandatory" BIER-TE forwarding. Cleaned up text to have requirements for different adjacencies in different paragraphs. (c).

5. Created new main section "BIER-TE Controller operational considerations", coalesced old sections 4., 5., 7. into this new main section. No text changes. (k).

5.1.9 Added new separate picture instead of referring to a picture later in text, adjusted text (r).

5.3.2 Changed title to not include word "comparison" to avoid this being accounted against Alvaros concern about scattering comparison (IMHO text already has little comparison, so title was misleading) (h).

co-authors internal review:

4.4 Added xref to Figure 5.

5.2.1 Duplicated ring picture, added visuals for described miswiring (s).

5.2.2 replace "topology" with graph (wrong word).

5.3.3 rewrote explanation of how to map BFR-id to SI:BP and assign them, clarified BFR-id is option. Retitled to better explain scope of section.

5.3.4 Removed considerations in 5.3.4 for sharing BFR-id across BIER/BIER-TE (t), changed title to explain how BFIR/BIER-TE controller interactions need some form of identifying BFR but this does not have to be BFR-id.

7. Added new security considerations (u).

09: Incorporated fixes for feedback from Shepherd (Xuesong Geng).

Added references for Bloom Filters and Rate Controlled Service Disciplines.

1.1 Fixed numbering of example 1 topology explanation. Improved language on second example (less abbreviating to avoid confusion about meaning).

1.2 Improved explanation of BIER-TE topology, fixed terminology of graphs (BIER-TE topology is a directed graph where the edges are the adjacencies).

2.4 Fixed and amended routing underlay explanations: detailed why no need for BFER routing underlay routing protocol extensions, but potential to re-use BIER routing underlay routing protocol extensions for non-BFER related extensions.

3.1 Added explanation for VRF and its use in adjacencies.

08: Incorporated (with hopefully acceptable fixes) for Lou suggested section 2.5, TE considerations.

Fixes are primarily to the point to a) emphasize that BIER-TE does not depend on the routing underlay unless `forward_routed()` adjacencies are used, and b) that the allocation and tracking of resources does not explicitly have to be tied to BPs, because they are just steering labels. Instead, it would ideally come from per-hop resource management that can be maintained only via local accounting in the controller.

07: Further reworking text for Lou.

Renamed BIER-PE to BIER-TE standing for "Tree Engineering" after votes from BIER WG.

Removed section 1.1 (introduced by version 06) because not considered necessary in this doc by Lou (for framework doc).

Added [RFC editor pls. remove] Section to explain name change to future reviewers.

06: Concern by Lou Berger re. BIER-TE as full traffic engineering solution.

Changed title "Traffic Engineering" to "Path Engineering"

Added intro section of relationship BIER-PE to traffic engineering.

Changed "traffic engineering" term in text to "path engineering", where appropriate

Other:

Shortened "BIER-TE Controller Host" to "BIER-TE Controller".
Fixed up all instances of controller to do this.

05: Review Jeffrey Zhang.

Part 2:

4.3 added note about leaf-BFER being also a property of routing setup.

4.7 Added missing details from example to avoid confusion with routed adjacencies, also compressed explanatory text and better justification why seed is explicitly configured by controller.

4.9 added section discussing generic reuse of BP methods.

4.10 added section summarizing BP optimizations of section 4.

6. Rewrote/compressed explanation of comparison BIER/BIER-TE forwarding difference. Explained benefit of BIER-TE per-BP forwarding being independent of forwarding for other BPs.

Part 1:

Explicitly use forwarded_connected adjacency in ECMP adjacency examples to avoid confusion.

4.3 Add picture as example for leaf vs. non-leaf BFR in topology. Improved description.

4.5 Example for traffic that can be broadcast -> for single BP in hub&spoke.

4.8.1 Simplified example picture for routed adjacency, explanatory text.

Review from Dirk Trossen:

Fixed up explanation of ICC paper vs. bloom filter.

04: spell check run.

Added remaining fixes for Sandys (Zhang Zheng) review:

4.7 Enhance ECMP explanations:

example ECMP algorithm, highlight that doc does not standardize ECMP algorithm.

Review from Dirk Trossen:

1. Added mentioning of prior work for traffic engineered paths with bloom filters.

2. Changed title from layers to components and added "BIER-TE control plane" to "BIER-TE Controller" to make it clearer, what it does.

2.2.3. Added reference to I-D.ietf-bier-multicast-http-response as an example solution.

2.3. clarified sentence about resetting BPs before sending copies (also forgot to mention DNR here).

3.4. Added text saying this section will be removed unless IESG review finds enough redeeming value in this example given how -03 introduced section 1.1 with basic examples.

7.2. Removed explicit numbers 20%/80% for number of topology bits in BIER-TE, replaced with more vague (high/low) description, because we do not have good reference material Added text saying this section will be removed unless IESG review finds enough redeeming value in this example given how -03 introduced section 1.1 with basic examples.

many typos fixed. Thanks a lot.

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 concluded 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 implementations 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. Removed MyBitsOfInterest (was pure optimization).

- Added captions to pictures.

- Part of review feedback from Sandy (Zhang Zheng) integrated.

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

- Source now on <https://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 sub-domain 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 5.3 to explain the use of SI, sub-domains 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 and CLI.

00: Initial version.

10. References

10.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.
- [RFC8279] Wijnands, IJ., Ed., Rosen, E., Ed., Dolganow, A., Przygienda, T., and S. Aldrin, "Multicast Using Bit Index Explicit Replication (BIER)", RFC 8279, DOI 10.17487/RFC8279, November 2017, <<https://www.rfc-editor.org/info/rfc8279>>.
- [RFC8296] Wijnands, IJ., Ed., Rosen, E., Ed., Dolganow, A., Tantsura, J., Aldrin, S., and I. Meilik, "Encapsulation for Bit Index Explicit Replication (BIER) in MPLS and Non-MPLS Networks", RFC 8296, DOI 10.17487/RFC8296, January 2018, <<https://www.rfc-editor.org/info/rfc8296>>.

10.2. Informative References

- [Bloom70] Bloom, B. H., "Space/time trade-offs in hash coding with allowable errors", Comm. ACM 13(7):422-6, July 1970, <<https://dl.acm.org/doi/10.1145/362686.362692>>.
- [I-D.eckert-bier-te-frr] Eckert, T., Cauchie, G., Braun, W., and M. Menth, "Protection Methods for BIER-TE", Work in Progress, Internet-Draft, draft-eckert-bier-te-frr-03, 5 March 2018, <<https://www.ietf.org/archive/id/draft-eckert-bier-te-frr-03.txt>>.
- [I-D.ietf-bier-multicast-http-response] Trossen, D., Rahman, A., Wang, C., and T. Eckert, "Applicability of BIER Multicast Overlay for Adaptive Streaming Services", Work in Progress, Internet-Draft,

draft-ietf-bier-multicast-http-response-06, 10 July 2021, <<https://www.ietf.org/archive/id/draft-ietf-bier-multicast-http-response-06.txt>>.

- [I-D.ietf-bier-non-mpls-bift-encoding]
Wijnands, I., Mishra, M., Xu, X., and H. Bidgoli, "An Optional Encoding of the BIFT-id Field in the non-MPLS BIER Encapsulation", Work in Progress, Internet-Draft, draft-ietf-bier-non-mpls-bift-encoding-04, 30 May 2021, <<https://www.ietf.org/archive/id/draft-ietf-bier-non-mpls-bift-encoding-04.txt>>.
- [I-D.ietf-bier-te-yang]
Zhang, Z., Wang, C., Chen, R., Hu, F., Sivakumar, M., and H. Chen, "A YANG data model for Tree Engineering for Bit Index Explicit Replication (BIER-TE)", Work in Progress, Internet-Draft, draft-ietf-bier-te-yang-04, 7 November 2021, <<https://www.ietf.org/archive/id/draft-ietf-bier-te-yang-04.txt>>.
- [I-D.ietf-roll-ccast]
Bergmann, O., Bormann, C., Gerdes, S., and H. Chen, "Constrained-Cast: Source-Routed Multicast for RPL", Work in Progress, Internet-Draft, draft-ietf-roll-ccast-01, 30 October 2017, <<https://www.ietf.org/archive/id/draft-ietf-roll-ccast-01.txt>>.
- [I-D.ietf-teas-rfc3272bis]
Farrel, A., "Overview and Principles of Internet Traffic Engineering", Work in Progress, Internet-Draft, draft-ietf-teas-rfc3272bis-16, 24 March 2022, <<https://www.ietf.org/archive/id/draft-ietf-teas-rfc3272bis-16.txt>>.
- [ICC]
Reed, M. J., Al-Naday, M., Thomos, N., Trossen, D., Petropoulos, G., and S. Spirou, "Stateless multicast switching in software defined networks", IEEE International Conference on Communications (ICC), Kuala Lumpur, Malaysia, 2016, May 2016, <<https://ieeexplore.ieee.org/document/7511036>>.
- [RCSD94]
Zhang, H. and D. Domenico, "Rate-Controlled Service Disciplines", Journal of High-Speed Networks, 1994, May 1994, <<https://dl.acm.org/doi/10.5555/2692227.2692232>>.
- [RFC4253]
Ylonen, T. and C. Lonvick, Ed., "The Secure Shell (SSH) Transport Layer Protocol", RFC 4253, DOI 10.17487/RFC4253, January 2006, <<https://www.rfc-editor.org/info/rfc4253>>.

- [RFC4456] Bates, T., Chen, E., and R. Chandra, "BGP Route Reflection: An Alternative to Full Mesh Internal BGP (IBGP)", RFC 4456, DOI 10.17487/RFC4456, April 2006, <<https://www.rfc-editor.org/info/rfc4456>>.
- [RFC4655] Farrel, A., Vasseur, J.-P., and J. Ash, "A Path Computation Element (PCE)-Based Architecture", RFC 4655, DOI 10.17487/RFC4655, August 2006, <<https://www.rfc-editor.org/info/rfc4655>>.
- [RFC5440] Vasseur, JP., Ed. and JL. Le Roux, Ed., "Path Computation Element (PCE) Communication Protocol (PCEP)", RFC 5440, DOI 10.17487/RFC5440, March 2009, <<https://www.rfc-editor.org/info/rfc5440>>.
- [RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011, <<https://www.rfc-editor.org/info/rfc6241>>.
- [RFC6242] Wasserman, M., "Using the NETCONF Protocol over Secure Shell (SSH)", RFC 6242, DOI 10.17487/RFC6242, June 2011, <<https://www.rfc-editor.org/info/rfc6242>>.
- [RFC7589] Badra, M., Luchuk, A., and J. Schoenwaelder, "Using the NETCONF Protocol over Transport Layer Security (TLS) with Mutual X.509 Authentication", RFC 7589, DOI 10.17487/RFC7589, June 2015, <<https://www.rfc-editor.org/info/rfc7589>>.
- [RFC7752] Gredler, H., Ed., Medved, J., Previdi, S., Farrel, A., and S. Ray, "North-Bound Distribution of Link-State and Traffic Engineering (TE) Information Using BGP", RFC 7752, DOI 10.17487/RFC7752, March 2016, <<https://www.rfc-editor.org/info/rfc7752>>.
- [RFC7950] Bjorklund, M., Ed., "The YANG 1.1 Data Modeling Language", RFC 7950, DOI 10.17487/RFC7950, August 2016, <<https://www.rfc-editor.org/info/rfc7950>>.
- [RFC7988] Rosen, E., Ed., Subramanian, K., and Z. Zhang, "Ingress Replication Tunnels in Multicast VPN", RFC 7988, DOI 10.17487/RFC7988, October 2016, <<https://www.rfc-editor.org/info/rfc7988>>.
- [RFC8040] Bierman, A., Bjorklund, M., and K. Watsen, "RESTCONF Protocol", RFC 8040, DOI 10.17487/RFC8040, January 2017, <<https://www.rfc-editor.org/info/rfc8040>>.

- [RFC8253] Lopez, D., Gonzalez de Dios, O., Wu, Q., and D. Dhody, "PCEPS: Usage of TLS to Provide a Secure Transport for the Path Computation Element Communication Protocol (PCEP)", RFC 8253, DOI 10.17487/RFC8253, October 2017, <<https://www.rfc-editor.org/info/rfc8253>>.
- [RFC8345] Clemm, A., Medved, J., Varga, R., Bahadur, N., Ananthakrishnan, H., and X. Liu, "A YANG Data Model for Network Topologies", RFC 8345, DOI 10.17487/RFC8345, March 2018, <<https://www.rfc-editor.org/info/rfc8345>>.
- [RFC8401] Ginsberg, L., Ed., Przygienda, T., Aldrin, S., and Z. Zhang, "Bit Index Explicit Replication (BIER) Support via IS-IS", RFC 8401, DOI 10.17487/RFC8401, June 2018, <<https://www.rfc-editor.org/info/rfc8401>>.
- [RFC8402] Filsfils, C., Ed., Previdi, S., Ed., Ginsberg, L., Decraene, B., Litkowski, S., and R. Shakir, "Segment Routing Architecture", RFC 8402, DOI 10.17487/RFC8402, July 2018, <<https://www.rfc-editor.org/info/rfc8402>>.
- [RFC8444] Psenak, P., Ed., Kumar, N., Wijnands, IJ., Dolganow, A., Przygienda, T., Zhang, J., and S. Aldrin, "OSPFv2 Extensions for Bit Index Explicit Replication (BIER)", RFC 8444, DOI 10.17487/RFC8444, November 2018, <<https://www.rfc-editor.org/info/rfc8444>>.
- [RFC8556] Rosen, E., Ed., Sivakumar, M., Przygienda, T., Aldrin, S., and A. Dolganow, "Multicast VPN Using Bit Index Explicit Replication (BIER)", RFC 8556, DOI 10.17487/RFC8556, April 2019, <<https://www.rfc-editor.org/info/rfc8556>>.

Appendix A. BIER-TE and Segment Routing (SR)

SR ([RFC8402]) aims to enable lightweight path steering via loose source routing. Compared to its more heavy-weight predecessor RSVP-TE, 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 bit position (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 ingress replication though (as in [RFC7988]). 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-ingress BFR.

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

Note that (non-TE) 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 in 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.

Authors' Addresses

Toerless Eckert (editor)
Futurewei Technologies Inc.
2330 Central Expwy
Santa Clara, 95050
United States of America
Email: tte+ietf@cs.fau.de

Michael Menth
University of Tuebingen
Email: menth@uni-tuebingen.de

Gregory Cauchie
KOEVOO
Email: gregory@koevoo.tech

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: January 2, 2020

J. Xie
Huawei Technologies
L. Geng
China Mobile
M. McBride
Futurewei
R. Asati
Cisco
S. Dhanaraj
G. Yan
Y. Xia
Huawei
July 1, 2019

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.

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	2
2. Terminology	3
3. BIER IPv6 Encapsulation	3
3.1. BIER Option in IPv6 Destination Options Header	3
3.2. Multicast and Unicast Destination Address	6
3.3. BIERv6 Packet Format	8
4. BIERv6 Packet Processing	9
5. Security Considerations	11
6. IANA Considerations	11
6.1. BIER Option Type	11
6.2. BIER Multicast Address	11
6.3. End.BIER Function	12
7. Acknowledgements	12
8. References	12
8.1. Normative References	12
8.2. Informative References	13
Authors' Addresses	13

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

Ethertype=0xAB37 for Non-MPLS BIER Header packets to be directly carried over the Ethernet links.

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:

- o BIERv6 - BIER IPv6.
- o BIER Option - An Option type carried in IPv6 Destination Options Header which includes the standard Non-MPLS BIER Header.
- o BIERv6 Header - An IPv6 Header with BIER Option.
- o 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.
- o 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

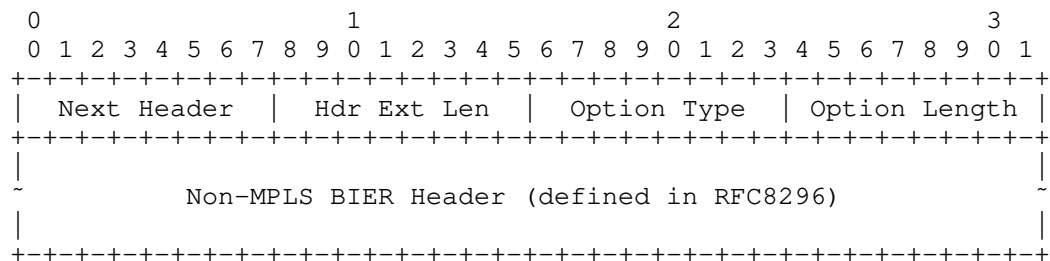
Destination Options Header and the Options that can be carried under this extension header is defined in [RFC8200]. This document defines a new Option type - BIER Option, to encode the Non-MPLS BIER header. As specified in Section 4.2 [RFC8200], the BIER Option follows type-length-value (TLV) encoding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



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.

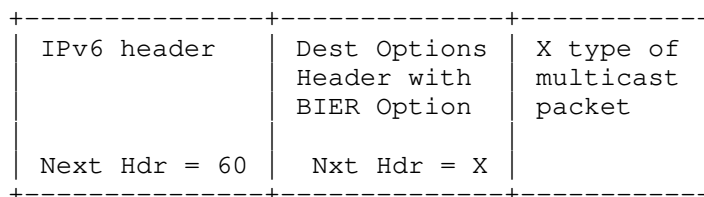
Ref3: An ICMPv6 packet using End.BIER as destination address.

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.



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:

- o After the routing header, if that header is present
- o Before the Fragment Header, if that header is present
- o 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 nexthop 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

determined by a single MFIB lookup on the BFER using both the IPv6 SA and IPv6 DA.

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.

Address(es)	Description	Reference
FF0X:0:0:0:0:0:0:AB37	ALL_BIER_FORWARDERS	This draft

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.

Value	Hex	Endpoint function	Reference
TBD	TBD	End.BIER	This draft

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

- [RFC6275] Perkins, C., Ed., Johnson, D., and J. Arkko, "Mobility Support in IPv6", RFC 6275, DOI 10.17487/RFC6275, July 2011, <<https://www.rfc-editor.org/info/rfc6275>>.
- [RFC7346] Droms, R., "IPv6 Multicast Address Scopes", RFC 7346, DOI 10.17487/RFC7346, August 2014, <<https://www.rfc-editor.org/info/rfc7346>>.
- [RFC8200] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", STD 86, RFC 8200, DOI 10.17487/RFC8200, July 2017, <<https://www.rfc-editor.org/info/rfc8200>>.
- [RFC8279] Wijnands, IJ., Ed., Rosen, E., Ed., Dolganow, A., Przygienda, T., and S. Aldrin, "Multicast Using Bit Index Explicit Replication (BIER)", RFC 8279, DOI 10.17487/RFC8279, November 2017, <<https://www.rfc-editor.org/info/rfc8279>>.

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

8.2. Informative References

- [I-D.ietf-bier-ipv6-requirements]
McBride, M., Xie, J., Dhanaraj, S., and R. Asati, "BIER IPv6 Requirements", draft-ietf-bier-ipv6-requirements-00 (work in progress), May 2019.
- [I-D.ietf-spring-srv6-network-programming]
Filsfils, C., Camarillo, P., Leddy, J., daniel.voyer@bell.ca, d., Matsushima, S., and Z. Li, "SRv6 Network Programming", draft-ietf-spring-srv6-network-programming-00 (work in progress), April 2019.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.

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

Rajiv Asati
Cisco

Email: rajiva@cisco.com

Senthil Dhanaraj
Huawei

Email: senthil.dhanaraj@huawei.com

Gang Yan
Huawei

Email: yangang@huawei.com

Yang Xia
Huawei

Email: yolanda.xia@huawei.com

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: January 2, 2020

J. Xie
Huawei Technologies
A. Wang
China Telecom
G. Yan
S. Dhanaraj
Huawei Technologies
July 1, 2019

BIER IPv6 Encapsulation (BIERv6) Support via IS-IS
draft-xie-bier-ipv6-isis-extension-00

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.

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	2
2. Terminology	3
3. Specification	3
3.1. Function sub-TLV for BIERv6	3
3.2. Encapsulation sub-sub-TLV for BIERv6	4
4. Security Considerations	4
5. IANA Considerations	4
5.1. Function sub-TLV Type Code	4
5.2. Encapsulation sub-sub-TLV Type Code	5
6. Acknowledgements	5
7. References	5
7.1. Normative References	5
7.2. Informative References	6
Authors' Addresses	6

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:

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

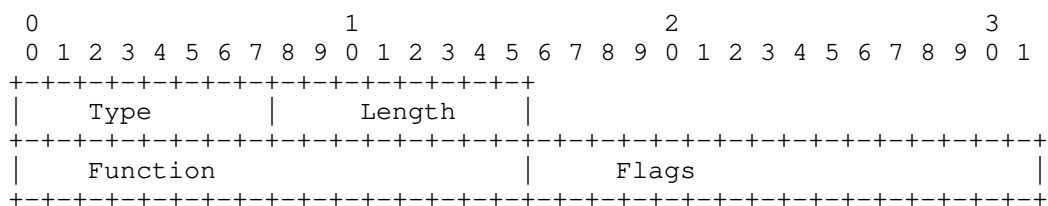
2. Terminology

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:



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

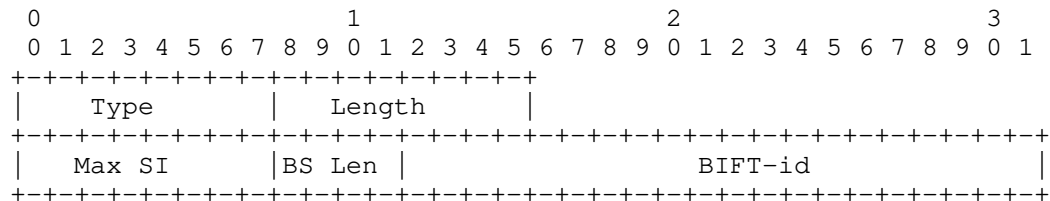
Length: 1 octet length in octets. Value 4 is set to this field.

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.



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.

Type: To be assigned by IANA.

Description: Function Info.

Reference: This document.

Type	135	235	236	237	Reference
32	y	y	y	y	RFC8401
TBD	n	n	y	y	This document

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]
Dhanaraj, S., Wijnands, I., Psenak, P., Zhang, Z., Yan, G., and J. Xie, "LSR Extensions for BIER over Ethernet", draft-ietf-bier-lsr-ethernet-extensions-00 (work in progress), May 2019.
- [I-D.ietf-spring-srv6-network-programming]
Filsfils, C., Camarillo, P., Leddy, J., daniel.voyer@bell.ca, d., Matsushima, S., and Z. Li, "SRv6 Network Programming", draft-ietf-spring-srv6-network-programming-00 (work in progress), April 2019.
- [I-D.xie-bier-ipv6-encapsulation]
Xie, J., Geng, L., McBride, M., Dhanaraj, S., Yan, G., and Y. Xia, "Encapsulation for BIER in Non-MPLS IPv6 Networks", draft-xie-bier-ipv6-encapsulation-01 (work in progress), June 2019.

- [RFC5120] Przygienda, T., Shen, N., and N. Sheth, "M-ISIS: Multi Topology (MT) Routing in Intermediate System to Intermediate Systems (IS-ISs)", RFC 5120, DOI 10.17487/RFC5120, February 2008, <<https://www.rfc-editor.org/info/rfc5120>>.
- [RFC5308] Hopps, C., "Routing IPv6 with IS-IS", RFC 5308, DOI 10.17487/RFC5308, October 2008, <<https://www.rfc-editor.org/info/rfc5308>>.
- [RFC7794] Ginsberg, L., Ed., Decraene, B., Previdi, S., Xu, X., and U. Chunduri, "IS-IS Prefix Attributes for Extended IPv4 and IPv6 Reachability", RFC 7794, DOI 10.17487/RFC7794, March 2016, <<https://www.rfc-editor.org/info/rfc7794>>.
- [RFC8279] Wijnands, IJ., Ed., Rosen, E., Ed., Dolganow, A., Przygienda, T., and S. Aldrin, "Multicast Using Bit Index Explicit Replication (BIER)", RFC 8279, DOI 10.17487/RFC8279, November 2017, <<https://www.rfc-editor.org/info/rfc8279>>.

7.2. Informative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.

Authors' Addresses

Jingrong Xie
Huawei Technologies

Email: xiejingrong@huawei.com

Aijun Wang
China Telecom

Email: wangaj.bri@chinatelecom.cn

Gang Yan
Huawei Technologies

Email: yangang@huawei.com

Senthil Dhanaraj
Huawei Technologies

Email: senthil.dhanaraj@huawei.com

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: January 2, 2020

J. Xie
Huawei Technologies
M. McBride
Futurewei
S. Dhanaraj
Huawei Technologies
L. Geng
China Mobile
July 1, 2019

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.

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	2
2. Terminology	3
3. Use of PTA and Prefix-SID Attribute in x-PMSI A-D Routes . .	4
4. MVPN over BIERv6 Core	4
5. GTM over BIERv6 Core	7
6. Data Plane	7
6.1. Encapsulation of Multicast Traffic	8
6.2. MTU	9
6.3. TTL	9
7. Security Considerations	9
8. IANA Considerations	9
9. Acknowledgements	10
10. References	10
10.1. Normative References	10
10.2. Informative References	11
Authors' Addresses	11

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.

- o BIERv6 - BIER in IPv6 networks using the BIERv6 encapsulation format defined in [I-D.xie-bier-ipv6-encapsulation].
- o SRv6 - Segment Routing instantiated on the IPv6 dataplane as defined in [I-D.ietf-spring-srv6-network-programming].
- o SRv6 SID - SRv6 Segment Identifier as defined in [I-D.ietf-spring-srv6-network-programming].

- o End.DTx - Refers to the functions End.DT6, End.DT4, End.DT46 defined in [I-D.ietf-spring-srv6-network-programming].
- o Src.DTx - Refers to the functions Src.DT4, Src.DT6, Src.DT46 defined in this document.
- o SRv6 L3 Service - L3VPN/Global-L3 service in SRv6 networks defined in [I-D.dawra-bess-srv6-services], or MVPN/GTM service in BIERv6 networks defined in this document.

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.

- o "Originating Router's IP Address" in the NLRI of Type 1 or Type 3 BGP-MVPN route is an IPv6 address.
- o "Network Address of Next Hop" field in the MP_REACH_NLRI attribute is an IPv6 address.
- o 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.
- o "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.

- o MPLS Label field in the BIER PTA MUST be set to Zero.
- o 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.
- o 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:

- o 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.
- o 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.
- o 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.
- o 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 adaptations. 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:

- o 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].
- o 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.
- o 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 the 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.

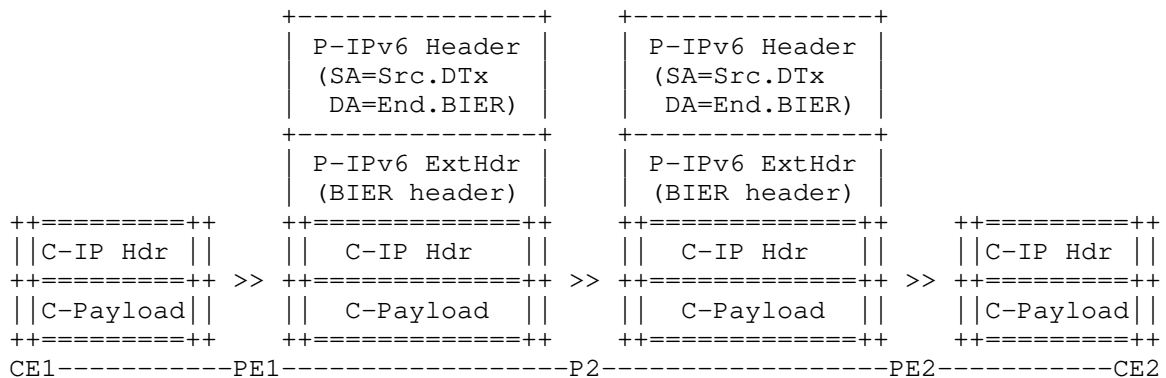


Figure 1: BIERv6 MVPN/GTM Intra-AS

In case of inter-AS scenario, BIERv6 packets may travel through unicast to a Border 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.DTx 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]
Dawra, G., Filsfils, C., Dukes, D., Brissette, P.,
Sethuram, S., Camarillo, P., Leddy, J.,
daniel.voyer@bell.ca, d., daniel.bernier@bell.ca, d.,
Steinberg, D., Raszuk, R., Decraene, B., Matsushima, S.,
and S. Zhuang, "SRv6 BGP based Overlay services", draft-
dawra-bess-srv6-services-00 (work in progress), March
2019.
- [I-D.ietf-spring-srv6-network-programming]
Filsfils, C., Camarillo, P., Leddy, J.,
daniel.voyer@bell.ca, d., Matsushima, S., and Z. Li, "SRv6
Network Programming", draft-ietf-spring-srv6-network-
programming-00 (work in progress), April 2019.
- [I-D.xie-bier-ipv6-encapsulation]
Xie, J., Geng, L., McBride, M., Dhanaraj, S., Yan, G., and
Y. Xia, "Encapsulation for BIER in Non-MPLS IPv6
Networks", draft-xie-bier-ipv6-encapsulation-01 (work in
progress), June 2019.
- [RFC6515] Aggarwal, R. and E. Rosen, "IPv4 and IPv6 Infrastructure
Addresses in BGP Updates for Multicast VPN", RFC 6515,
DOI 10.17487/RFC6515, February 2012,
<<https://www.rfc-editor.org/info/rfc6515>>.
- [RFC6625] Rosen, E., Ed., Rekhter, Y., Ed., Hendrickx, W., and R.
Qiu, "Wildcards in Multicast VPN Auto-Discovery Routes",
RFC 6625, DOI 10.17487/RFC6625, May 2012,
<<https://www.rfc-editor.org/info/rfc6625>>.
- [RFC7716] Zhang, J., Giuliano, L., Rosen, E., Ed., Subramanian, K.,
and D. Pacella, "Global Table Multicast with BGP Multicast
VPN (BGP-MVPN) Procedures", RFC 7716,
DOI 10.17487/RFC7716, December 2015,
<<https://www.rfc-editor.org/info/rfc7716>>.

- [RFC8279] Wijnands, IJ., Ed., Rosen, E., Ed., Dolganow, A., Przygienda, T., and S. Aldrin, "Multicast Using Bit Index Explicit Replication (BIER)", RFC 8279, DOI 10.17487/RFC8279, November 2017, <<https://www.rfc-editor.org/info/rfc8279>>.
- [RFC8296] Wijnands, IJ., Ed., Rosen, E., Ed., Dolganow, A., Tantsura, J., Aldrin, S., and I. Meilik, "Encapsulation for Bit Index Explicit Replication (BIER) in MPLS and Non-MPLS Networks", RFC 8296, DOI 10.17487/RFC8296, January 2018, <<https://www.rfc-editor.org/info/rfc8296>>.
- [RFC8556] Rosen, E., Ed., Sivakumar, M., Przygienda, T., Aldrin, S., and A. Dolganow, "Multicast VPN Using Bit Index Explicit Replication (BIER)", RFC 8556, DOI 10.17487/RFC8556, April 2019, <<https://www.rfc-editor.org/info/rfc8556>>.

10.2. Informative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.

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

BIER
Internet-Draft
Intended status: Standards Track
Expires: January 9, 2020

Z. Zhang
ZTE Corporation
A. Przygienda
Juniper Networks, Inc.
I. Wijnands
Cisco Systems
H. Bidgoli
Nokia
M. McBride
Futurewei
July 8, 2019

BIER in IPv6 (BIERin6)
draft-zhang-bier-bierin6-03

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.

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	2
2. IPv6 Header	3
2.1. IPv6 Options Considerations	3
3. BIER Header	4
4. IPv6 Encapsulation Advertisement	4
4.1. Format	4
4.2. Inter-area prefix redistribution	5
5. IANA Considerations	5
6. Security Considerations	5
7. Acknowledgement	5
8. References	5
8.1. Normative References	6
8.2. Informative References	6
Authors' Addresses	7

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].

[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 encapsualtion 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].

8.1. Normative References

- [RFC8200] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", STD 86, RFC 8200, DOI 10.17487/RFC8200, July 2017, <<https://www.rfc-editor.org/info/rfc8200>>.
- [RFC8279] Wijnands, IJ., Ed., Rosen, E., Ed., Dolganow, A., Przygienda, T., and S. Aldrin, "Multicast Using Bit Index Explicit Replication (BIER)", RFC 8279, DOI 10.17487/RFC8279, November 2017, <<https://www.rfc-editor.org/info/rfc8279>>.
- [RFC8296] Wijnands, IJ., Ed., Rosen, E., Ed., Dolganow, A., Tantsura, J., Aldrin, S., and I. Meilik, "Encapsulation for Bit Index Explicit Replication (BIER) in MPLS and Non-MPLS Networks", RFC 8296, DOI 10.17487/RFC8296, January 2018, <<https://www.rfc-editor.org/info/rfc8296>>.
- [RFC8401] Ginsberg, L., Ed., Przygienda, T., Aldrin, S., and Z. Zhang, "Bit Index Explicit Replication (BIER) Support via IS-IS", RFC 8401, DOI 10.17487/RFC8401, June 2018, <<https://www.rfc-editor.org/info/rfc8401>>.
- [RFC8444] Psenak, P., Ed., Kumar, N., Wijnands, IJ., Dolganow, A., Przygienda, T., Zhang, J., and S. Aldrin, "OSPFv2 Extensions for Bit Index Explicit Replication (BIER)", RFC 8444, DOI 10.17487/RFC8444, November 2018, <<https://www.rfc-editor.org/info/rfc8444>>.

8.2. Informative References

- [I-D.dhanaraj-bier-lsr-ethernet-extensions]
Dhanaraj, S., Wijnands, I., Psenak, P., Zhang, Z., Yan, G., and J. Xie, "LSR Extensions for BIER over Ethernet", draft-dhanaraj-bier-lsr-ethernet-extensions-00 (work in progress), January 2019.
- [I-D.ietf-bier-bar-ipa]
Zhang, Z., Przygienda, T., Dolganow, A., Bidgoli, H., Wijnands, I., and A. Gulko, "BIER Underlay Path Calculation Algorithm and Constraints", draft-ietf-bier-bar-ipa-04 (work in progress), May 2019.
- [I-D.ietf-bier-idr-extensions]
Xu, X., Chen, M., Patel, K., Wijnands, I., and T. Przygienda, "BGP Extensions for BIER", draft-ietf-bier-idr-extensions-06 (work in progress), January 2019.

- [I-D.ietf-bier-non-mpls-bift-encoding]
Wijnands, I., Xu, X., and H. Bidgoli, "An Optional Encoding of the BIFT-id Field in the non-MPLS BIER Encapsulation", draft-ietf-bier-non-mpls-bift-encoding-01 (work in progress), October 2018.
- [I-D.zhang-bier-babel-extensions]
Zhang, Z. and T. Przygienda, "BIER in BABEL", draft-zhang-bier-babel-extensions-01 (work in progress), June 2017.
- [I-D.zwzw-bier-prefix-redistribute]
Zhang, Z., Bo, W., Zhang, Z., and I. Wijnands, "BIER Prefix Redistribute", draft-zwzw-bier-prefix-redistribute-02 (work in progress), March 2019.
- [RFC7368] Chown, T., Ed., Arkko, J., Brandt, A., Troan, O., and J. Weil, "IPv6 Home Networking Architecture Principles", RFC 7368, DOI 10.17487/RFC7368, October 2014, <<https://www.rfc-editor.org/info/rfc7368>>.

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

Mike McBride
Futurewei

EMail: mmcbride@futurewei.com

BIER WG
Internet-Draft
Intended status: Informational
Expires: January 8, 2020

Zheng. Zhang
Greg. Mirsky
Quan. Xiong
ZTE Corporation
July 7, 2019

BIER Source Protection
draft-zhang-bier-source-protection-00

Abstract

This document describes the multicast source protection functions in 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.

Table of Contents

1. Introduction	2
2. Multicast Source Protection	2
2.1. BIER Ping	3
2.2. BIER BFD	4
3. Security Considerations	4
4. Normative References	4
Authors' Addresses	5

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, EVPN. 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.

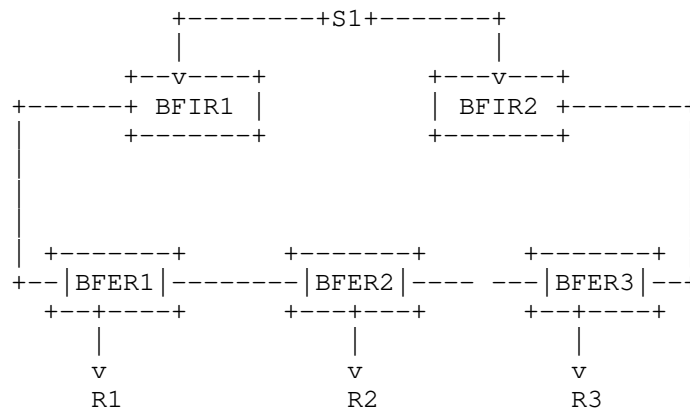


Figure 1

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], BFER1 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]

Xiong, Q., Mirsky, G., hu, f., and C. Liu, "BIER BFD", draft-hu-bier-bfd-04 (work in progress), July 2019.

[I-D.ietf-bier-ping]

Kumar, N., Pignataro, C., Akiya, N., Zheng, L., Chen, M., and G. Mirsky, "BIER Ping and Trace", draft-ietf-bier-ping-05 (work in progress), April 2019.

- [RFC8279] Wijnands, IJ., Ed., Rosen, E., Ed., Dolganow, A., Przygienda, T., and S. Aldrin, "Multicast Using Bit Index Explicit Replication (BIER)", RFC 8279, DOI 10.17487/RFC8279, November 2017, <<https://www.rfc-editor.org/info/rfc8279>>.
- [RFC8562] Katz, D., Ward, D., Pallagatti, S., Ed., and G. Mirsky, Ed., "Bidirectional Forwarding Detection (BFD) for Multipoint Networks", RFC 8562, DOI 10.17487/RFC8562, April 2019, <<https://www.rfc-editor.org/info/rfc8562>>.

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

BIER
Internet-Draft
Intended status: Standards Track
Expires: January 7, 2020

Z. Zhang
Juniper Networks
N. Warnke
Deutsche Telekom
I. Wijnands
Cisco Systems
D. Awduche
Verizon
July 6, 2019

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.

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. Terminologies	2
2. Introduction	2
3. Additional Considerations	4
4. Specification	6
4.1. Advertising from Helped Node	6
4.2. Advertising from Helper Node	7
4.3. Procedures for BGP Signaling	7
5. Security Considerations	8
6. IANA Considerations	8
7. Acknowledgements	8
8. Normative References	9
Authors' Addresses	9

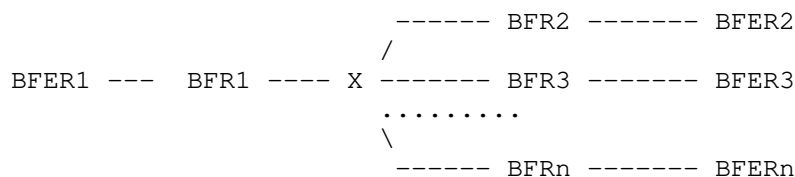
1. Terminologies

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

[To be added].

2. Introduction

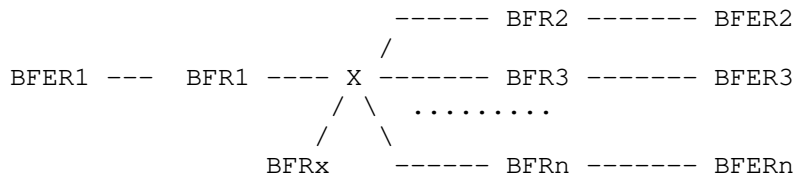
Consider the following scenario where router X does not support BIER.



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 BFRx to X:



Instead of BFR1 tunneling to BFR2, ..., BFRn directly, BFR1 will get BIER packets to BFRx, who will then tunnel to BFR2, ..., BFRn. There could be fat and local pipes between the tethered BFRx and X, so ingress replication from BFRx is acceptable.

For BFR1 to tunnel BIER packets to BFRx, the BFR1-BFRx tunnel need to be announced in IGP as a forwarding adjacency so that BFRx will appear on the SPF tree. This need to happen in a BIER specific topology so that unicast traffic would not be tunneled to BFRx. 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 BFRx will not appear on the SPF tree of BFR1.

There is a simple solution to the problem though. Even though X does not support BIER forwarding, it could advertises 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 BFRx, 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, BFRx 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 BFRx is its helper node, so that other BFRs could optionally use the Section 6.9 method to tunnel to BFRx, instead of sending native BIER packets to X and rely on X label switching to BFRx. 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.

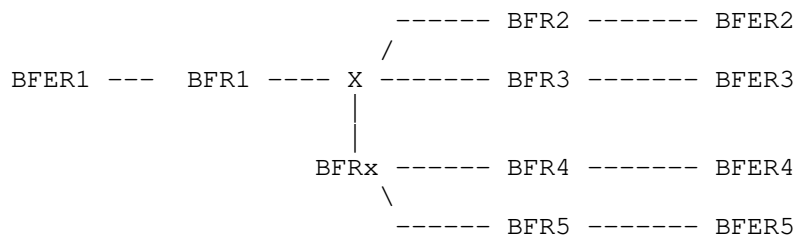
The two options both have pros and cons - the first option only needs X and BFRx to support the new procedure while the second option does not require anything to be done to the BIER incapable X.

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

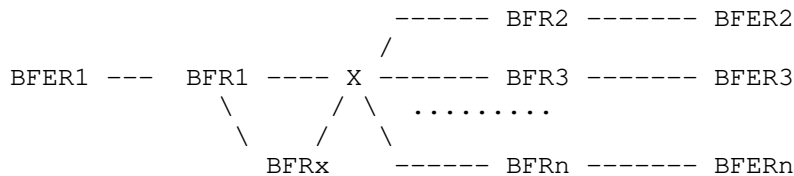
While the example shows a local connection between BFRx and X, it does not have to be like that. As long as packets can arrive at BFRx without requiring X to do BIER forwarding, it should work. For example, X could label switch incoming BIER packets through a multi-hop tunnel to BFRx, or other BFRs could 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:



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.

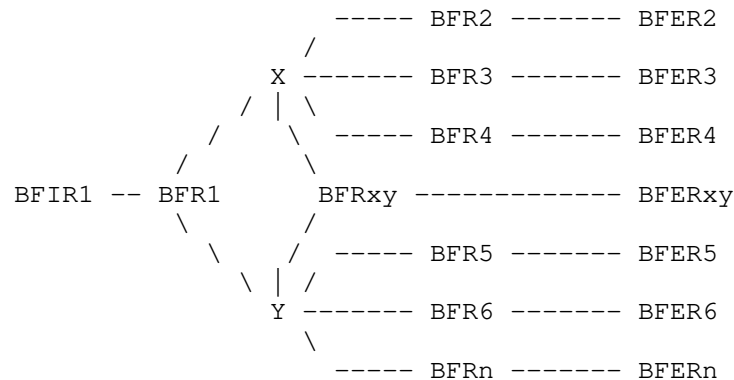


This can easily be prevented if BFR1 does an SPF calculation with the helper BFRx as the root. For any BFERN reached via X from BFR1, if BFRx's SPF path to BFERN includes BFR1 then BFR1 must not use the helper. Instead, BFR1 must directly tunnel packets for BFERN to X's BFR (grand-)child on BFR1's SPF 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.zhang-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:

Type	Name
-----	-----
TBD1	BIER Helper Node
TBD2	BIER Helped Node

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

Type	Name
-----	-----
TBD3	BIER Helper Node
TBD4	BIER Helped Node

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]
Xu, X., Chen, M., Patel, K., Wijnands, I., and T. Przygienda, "BGP Extensions for BIER", draft-ietf-bier-idr-extensions-06 (work in progress), January 2019.
- [I-D.ietf-idr-tunnel-encaps]
Patel, K., Velde, G., Ramachandra, S., and E. Rosen, "The BGP Tunnel Encapsulation Attribute", draft-ietf-idr-tunnel-encaps-12 (work in progress), May 2019.
- [I-D.zzhang-bier-multicast-as-a-service]
Zhang, Z., Rosen, E., and L. Geng, "Multicast/BIER As A Service", draft-zzhang-bier-multicast-as-a-service-00 (work in progress), October 2018.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC8279] Wijnands, IJ., Ed., Rosen, E., Ed., Dolganow, A., Przygienda, T., and S. Aldrin, "Multicast Using Bit Index Explicit Replication (BIER)", RFC 8279, DOI 10.17487/RFC8279, November 2017, <<https://www.rfc-editor.org/info/rfc8279>>.
- [RFC8401] Ginsberg, L., Ed., Przygienda, T., Aldrin, S., and Z. Zhang, "Bit Index Explicit Replication (BIER) Support via IS-IS", RFC 8401, DOI 10.17487/RFC8401, June 2018, <<https://www.rfc-editor.org/info/rfc8401>>.
- [RFC8444] Psenak, P., Ed., Kumar, N., Wijnands, IJ., Dolganow, A., Przygienda, T., Zhang, J., and S. Aldrin, "OSPFv2 Extensions for Bit Index Explicit Replication (BIER)", RFC 8444, DOI 10.17487/RFC8444, November 2018, <<https://www.rfc-editor.org/info/rfc8444>>.

Authors' Addresses

Zhaohui Zhang
Juniper Networks

EMail: zzhang@juniper.net

Nils Warnke
Deutsche Telekom

EMail: Nils.Warnke@telekom.de

IJsbrand Wijnands
Cisco Systems

EMail: ice@cisco.com

Daniel Awduche
Verizon

EMail: daniel.awduche@verizon.com