BIER WG
Internet-Draft
Intended status: Standards Track
Expires: April 14, 2019

Fangwei Hu
Greg Mirsky
Quan Xiong
ZTE Corporation
Chang Liu
China Unicom

Oct 11, 2018

BIER BFD
draft-hu-bier-bfd-02.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 April 14, 2019.

Copyright Notice

Table of Contents

1. Introduction 2
2. Conventions used in this document 2
2.1. Terminology 3
2.2. Requirements Language 3
3. BIER BFD Encapsulation 3
4. Bootstrapping BIER BFD 3
5. Discriminators and Packet Demultiplexing 3
6. Security Considerations 3
7. Acknowledgements 4
8. IANA Considerations 4
9. References 4
9.1. Normative References 4
9.2. Informative References 5
Authors' Addresses 5

1. Introduction

Bit Index Explicit Replication(BIER)[RFC8279] provides optimal 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.

[I-D.ietf-bfd-multipoint] defines a method of using Bidirectional Detection(BFD) to monitor and detect unicast failures between the sender (head) and one or more receivers (tails) in multipoint or multicast networks.

This document describes the procedures for using such mode of BFD protocol to verify multipoint or multicast connectivity between a multipoint sender (the "head", Bit-Forwarding Ingress Routers(BFIRs)) and a set of one or more multipoint receivers (the"tails", Bit-Forwarding Egress Routers(BFERs)). The BIER BFD only supports the unidirectional multicast. This document defines the use of BFD, as defined in [I-D.ietf-bfd-multipoint], for BIER domain. Use of BFD for multipoint networks active tail [I-D.ietf-bfd-multipoint-active-tail] is for further study.

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 (TBD by IANA). BFD Control packet, defined in Section 4 [RFC5880] immediately follows the BIER OAM header.

4. Bootstrapping BIER BFD

The BIER OAM ping could be used for BIER BFD bootstrap. The multipoint header sends the BIER OAM packet with Target SI-Bitstring TLV (section 3.3.2 of [I-D.ietf-bier-ping]) carrying the set of BFER information (Sub-domain-id, Set ID, BS Len, Bitstring) to the multipoint tails to bootstrap the BIER BFD sessions.

5. Discriminators and Packet Demultiplexing

The tail(BFER) demultiplexes incoming BFD packets based on a combination of the source address and My discriminator as specified in [I-D.ietf-bfd-multipoint]. The source address is BFIR-id and BIER MPLS Label(MPLS network) or BFIR-id and BIFT-id(Non-MPLS network)for BIER BFD.

6. Security Considerations

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

For general multipoint BFD security considerations, see [I-D.ietf-bfd-multipoint].

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

7. Acknowledgements

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

8. IANA Considerations

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

Value	Description	Reference
TBD	BIER BFD	[this document]

Table 1

9. References

9.1. Normative References

[I-D.ietf-bfd-multipoint]
Katz, D., Ward, D., Networks, J., and G. Mirsky, "BFD for Multipoint Networks", draft-ietf-bfd-multipoint-18 (work in progress), June 2018.

[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-03 (work in progress), January 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>.

[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>.

[RFC6213] Hopps, C. and L. Ginsberg, "IS-IS BFD-Enabled TLV", RFC 6213, DOI 10.17487/RFC6213, April 2011, <https://www.rfc-editor.org/info/rfc6213>.

[RFC6328] Eastlake 3rd, D., "IANA Considerations for Network Layer Protocol Identifiers", BCP 164, RFC 6328, DOI 10.17487/RFC6328, July 2011, <https://www.rfc-editor.org/info/rfc6328>.

[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

[I-D.ietf-bfd-multipoint-active-tail] Katz, D., Ward, D., Networks, J., and G. Mirsky, "BFD Multipoint Active Tails.", draft-ietf-bfd-multipoint-active-tail-09 (work in progress), June 2018.

[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

Fangwei Hu
ZTE Corporation
No.889 Bibo Rd
Shanghai 201203
China

Phone: +86 21 68896273
Email: hu.fangwei@zte.com.cn

Greg Mirsky
ZTE Corporation
USA

Email: gregimirsky@gmail.com

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

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

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

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

BIER Working Group
Internet-Draft
Intended status: Informational
Expires: April 21, 2019

IJ. Wijnands
Cisco Systems
X. Xu
Alibaba Group
H. Bidgoli
Nokia

October 18, 2018

An Optional Encoding of the BIFT-id Field in the non-MPLS BIER Encapsulation
draft-ietf-bier-non-mpls-bift-encoding-01

Abstract

Bit Index Explicit Replication (BIER) is an architecture that provides optimal multicast forwarding through a "multicast domain", without requiring intermediate routers to maintain any per-flow state or to engage in an explicit tree-building protocol. The Multicast packet is encapsulated using a BIER Header and transported through an MPLS or non-MPLS network. When MPLS is used as the transport, the Bit Indexed Forwarding Table (BIFT) is identified by a MPLS Label. When non-MPLS transport is used, the BIFT is identified by a 20bit value. This document describes one way of encoding the 20bit value.

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 21, 2019.

Table of Contents

1. Introduction 2
2. Terminology and Definitions 3
3. Specification of Requirements 3
4. The Bit Index Forwarding Table 3
5. The Non-MPLS Static BSL-SD-SI BIFT Encoding 4
6. The Non-MPLS Static IBU-SI BIFT Encoding 4
7. Security Considerations 5
8. IANA Considerations 5
9. Acknowledgments 5
10. Normative References 5
Authors' Addresses 6

1. Introduction

Bit Index Explicit Replication (BIER) [RFC8279] is an architecture that provides optimal multicast forwarding through a "multicast domain", without requiring intermediate routers to maintain any per-flow state or to engage in an explicit tree-building protocol. The Multicast packet is encapsulated [RFC8296] using a BIER Header and transported through an MPLS or non-MPLS network. When MPLS is used as the transport, the Bit Indexed Forwarding Table (BIFT) is identified by a MPLS Label. When non-MPLS transport is used, the BIFT is identified by a 20bit value. This document describes one way of encoding the 20bit value, based on the Sub-Domain (SD), Set Identifier (SI) and BitStringLength (BSL) values.

The BIER architecture requires that a BFR has a BIFT for every combination of <SD, SI, BSL> that is being used. When processing a BIER packet, the correct BIFT is inferred from the BIFT-id field of the encapsulation. When the non-MPLS encapsulation is used in a given BIER domain, it may be desirable for the a BIFT-id to be unique in that domain. This document describes an OPTIONAL method that can be used to form domain-wide unique BIFT-ids based on the <SD, SI, BSL> triples. If in the future the BIER architecture is extended with an additional BIFT argument, this encoding does not generate domain-wide unique identifiers anymore.

This encoding, if used, is only for the convenience of the network adminstrators. When forwarding a BIER packet, the BIFT-id is used as an opaque 20-bit value that identifies a BIFT; the forwarding procedures do not parse the 20-bit value, they just use it as a lookup key.

2. Terminology and Definitions

Readers of this document are assumed to be familiar with the terminology and concepts of the documents listed as Normative References. For convenience, some of the more frequently used terms appear below.

BIER:
Bit Indexed Explicit Replication.

BIFT-id:
Bit Indexed Forwarding Table Identifier.

3. Specification of Requirements

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].

4. The Bit Index Forwarding Table

In MPLS networks a BIER label is allocated for each Bit Index Forwarding Table (BIFT) from the platform specific, downstream label database ([RFC8296]). This label is associated with a particular combination of BIER Sub-Domain (SD), Set Identifier (SI) and BitStringLength (BSL). In order for the network to know which MPLS label represents a particular combination of <SD, SI, BS>, this mapping has to be advertised through the network. This is currently done through an IGP or BGP. In MPLS networks this is not a drawback as the MPLS label has to be advertised anyway.

When the non-MPLS encoding is chosen, there is no need to advertise the BIFT-id to <SD, SI, BSL> mapping if the BIFT-id is domain-wide unique. For this reason we’re defining two encodings that MAY be used by operators to compute the domain-wide unique BIFT-id values from the SD, BSL and/or SI. Although the BIFT-id is not expected to change, it may change when the BSL mismatch procedures [RFC8279] section 6.10.2 are applied.

5. The Non-MPLS Static BSL-SD-SI BIFT Encoding

Find below the first 32 bits of the BIER header, encoding the SD, SI and BSL into the 20 bit BIFT-id field.

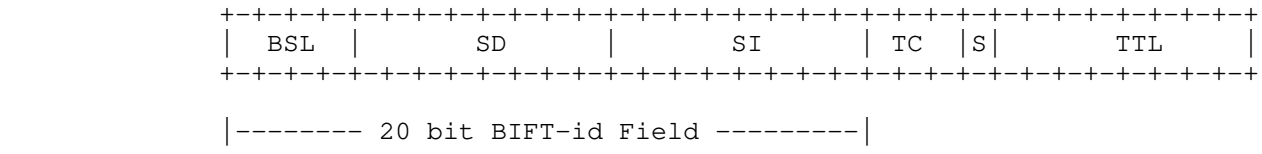


Figure 1

BSL: This 4-bit field encodes the length in bits of the BitString. These are the same values as documented in [RFC8296].

SD: This is a 8-bit field that encodes the Sub-Domain as described in [RFC8279].

SI: This is a 8-bit field that encodes the Set-ID as described in [RFC8279].

TC: This is a 3-bit field set to 000 (following [RFC8296]).

S: This is a 1-bit field set to 1 (following [RFC8296]).

TTL: See [RFC8296].

6. The Non-MPLS Static IBU-SI BIFT Encoding

Find below the first 32 bits of the BIER header, encoding the provisioned Index BIFT Unit (IBU) and SI into the 20 bit BIFT-id field. The IBU replaces the BSL and SD values as described in the encoding above. This provides additional flexibility in-case there is a need to support additional arguments other than BSL and SD to create the BIFT-id.

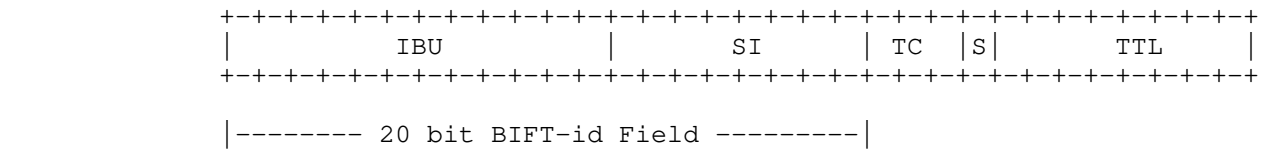


Figure 2

IBU: The IBU is a 12-bit field that encodes the provisioned Index BIFT Unit.

SI: This is a 8-bit field that encodes the Set-ID as described in [RFC8279].

TC: This is a 3-bit field set to 000 (following [RFC8296]).

S: This is a 1-bit field set to 1 (following [RFC8296]).

TTL: See [RFC8296].

7. Security Considerations

This document does not introduce any new security considerations other than already discussed in [RFC8279].

8. IANA Considerations

There is no IANA consideration.

9. Acknowledgments

The authors like to thank the following people for their comments and contributions to this document; Eric Rosen, Neale Ranns, Jeffrey Zhang.

10. 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>.

[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

IJsbrand Wijnands
Cisco Systems
De Kleetlaan 6a
Diegem 1831
Belgium

Email: ice@cisco.com

Xiaohu Xu
Alibaba Group

Email: xiaohu.xxh@alibaba-inc.com

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

Email: hooman.bidgoli@nokia.com

Network Working Group
Internet-Draft
Intended status: Experimental
Expires: April 26, 2019

T. Eckert, Ed.
Huawei
G. Cauchie
Bouygues Telecom
W. Braun
M. Menth
University of Tuebingen
October 23, 2018

Traffic Engineering for Bit Index Explicit Replication (BIER-TE)
draft-ietf-bier-te-arch-01

Abstract

This document proposes an architecture for BIER-TE: Traffic Engineering for Bit Index Explicit Replication (BIER).

BIER-TE shares part of its architecture with BIER as described in [RFC8279]. It also proposes to share the packet format with BIER.

BIER-TE forwards and replicates packets like BIER based on a BitString in the packet header but it does not require an IGP. It does support traffic engineering by explicit hop-by-hop forwarding and loose hop forwarding of packets. It does support Fast ReRoute (FRR) for link and node protection and incremental deployment. Because BIER-TE like BIER operates without explicit in-network tree-building but also supports traffic engineering, it is more similar to SR than RSVP-TE.

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 26, 2019.

Table of Contents

1. Introduction 3
1.1. Overview 3
1.2. Requirements Language 4
2. Layering 4
2.1. The Multicast Flow Overlay 5
2.2. The BIER-TE Controller Host 5
2.2.1. Assignment of BitPositions to adjacencies of the network topology 6
2.2.2. Changes in the network topology 6
2.2.3. Set up per-multicast flow BIER-TE state 6
2.2.4. Link/Node Failures and Recovery 6
2.3. The BIER-TE Forwarding Layer 7
2.4. The Routing Underlay 7
3. BIER-TE Forwarding 7
3.1. The Bit Index Forwarding Table (BIFT) 7
3.2. Adjacency Types 8
3.2.1. Forward Connected 8
3.2.2. Forward Routed 9
3.2.3. ECMP 9
3.2.4. Local Decap 9
3.3. Encapsulation considerations 10
3.4. Basic BIER-TE Forwarding Example 10
3.5. Forwarding comparison with BIER 12
3.6. Requirements 13
4. BIER-TE Controller Host BitPosition Assignments 13
4.1. P2P Links 14
4.2. BFER 14
4.3. Leaf BFERs 14
4.4. LANs 14
4.5. Hub and Spoke 15
4.6. Rings 15

4.7. Equal Cost MultiPath (ECMP) 16
4.8. Routed adjacencies 19
4.8.1. Reducing BitPositions 19
4.8.2. Supporting nodes without BIER-TE 19
5. Avoiding loops and duplicates 19
5.1. Loops 19
5.2. Duplicates 20
6. BIER-TE Forwarding Pseudocode 20
7. Managing SI, subdomains and BFR-ids 23
7.1. Why SI and sub-domains 24
7.2. Bit assignment comparison BIER and BIER-TE 25
7.3. Using BFR-id with BIER-TE 25
7.4. Assigning BFR-ids for BIER-TE 26
7.5. Example bit allocations 27
7.5.1. With BIER 27
7.5.2. With BIER-TE 28
7.6. Summary 29
8. BIER-TE and Segment Routing 29
9. Security Considerations 30
10. IANA Considerations 30
11. Acknowledgements 30
12. Change log [RFC Editor: Please remove] 30
13. References 32
Authors' Addresses 33

1. Introduction

1.1. Overview

This document specifies the architecture for BIER-TE: traffic engineering for Bit Index Explicit Replication BIER.

BIER-TE shares architecture and packet formats with BIER as described in [RFC8279].

BIER-TE forwards and replicates packets like BIER based on a BitString in the packet header but it does not require an IGP. It does support traffic engineering by explicit hop-by-hop forwarding and loose hop forwarding of packets. It does support incremental deployment and a Fast ReRoute (FRR) extension for link and node protection is given in [I-D.eckert-bier-te-frr]. Because BIER-TE like BIER operates without explicit in-network tree-building but also supports traffic engineering, it is more similar to Segment Routing (SR) than RSVP-TE.

The key differences over BIER are:

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

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

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

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

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

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

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

1.2. Requirements Language

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

2. Layering

End to end BIER-TE operations consists of four components: The "Multicast Flow Overlay", the "BIER-TE Controller Host", the "Routing Underlay" and the "BIER-TE forwarding layer".

Picture 2: Layers of BIER-TE

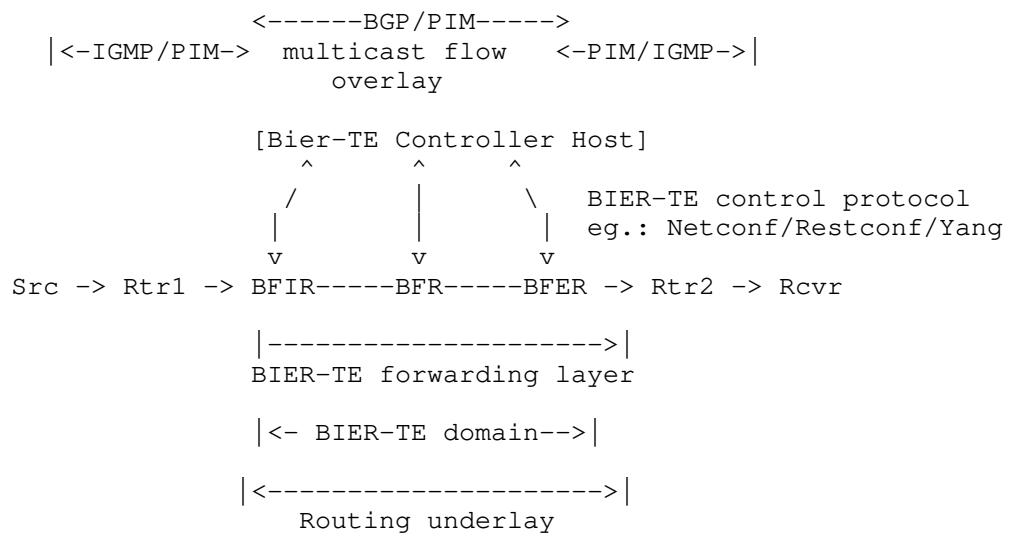


Figure 1: BIER-TE architecture

2.1. The Multicast Flow Overlay

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

2.2. The BIER-TE Controller Host

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

During bring-up or modifications of the network topology, the controller discovers the network topology, assigns BitPositions to adjacencies and signals the resulting mapping of BitPositions to adjacencies to each BFR connecting to the adjacency.

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

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

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

2.2.1. Assignment of BitPositions to adjacencies of the network topology

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

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

2.2.2. Changes in the network topology

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

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

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

2.2.4. Link/Node Failures and Recovery

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

is all performed locally on a BFR receiving the adjacency up/down notification.

2.3. The BIER-TE Forwarding Layer

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

2.4. The Routing Underlay

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

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

3. BIER-TE Forwarding

3.1. The Bit Index Forwarding Table (BIFT)

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

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

In the BIER architecture, indices into the BIFT are explained to be both BFR-id and SI:BitString (BitPosition). This is because there is a 1:1 relationship between BFR-id and SI:BitString - every bit in every SI is/can be assigned to a BFIR/BFER. In BIER-TE there are more bits used in each BitString than there are BFIR/BFER assigned to the bitstring. This is because of the bits required to express the (traffic engineered) path through the topology. The BIER-TE

forwarding definitions do therefore not use the term BFR-id at all. Instead, BFR-ids are only used as required by routing underlay, flow overlay of BIER headers. Please refer to Section 7 for explanations how to deal with SI, subdomains and BFR-id in BIER-TE.

Index: SI:BitPosition	Adjacencies: <empty> or one or more per entry
0:1	forward_connected(interface,neighbor,DNR)
0:2	forward_connected(interface,neighbor,DNR) forward_connected(interface,neighbor,DNR)
0:3	local_decap([VRF])
0:4	forward_routed([VRF,]l3-neighbor)
0:5	<empty>
0:6	ECMP({adjacency1,...adjacencyN}, seed)
...	
BitStringLength	...

Bit Index Forwarding Table

Figure 2: BIFT adjacencies

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

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

3.2. Adjacency Types

3.2.1. Forward Connected

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

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

3.2.2. Forward Routed

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

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

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

3.2.3. ECMP

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

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

3.2.4. Local Decap

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

3.3. Encapsulation considerations

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

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

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

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

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

3.4. Basic BIER-TE Forwarding Example

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

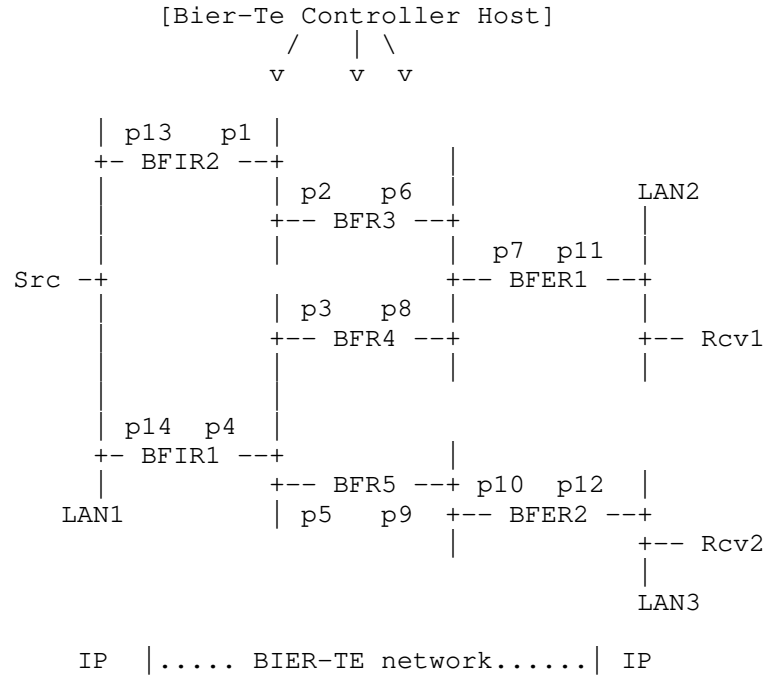


Figure 3: BIER-TE Forwarding Example

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

```
BIFT BFIR2:
  p13: local_decap()
   p2: forward_connected(BFR3)

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

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

Figure 4: BIER-TE Forwarding Example Adjacencies

...and so on.

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

```
                -> BFER1 ---------------> Rcv1
 BFIR2 -> BFR3
                -> BFR4 -> BFR5 -> BFER2 -> Rcv2
```

Figure 5: BIER-TE Forwarding Example Paths

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

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

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

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

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

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

3.5. Forwarding comparison with BIER

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

The pseudocode in Section 6 shows how existing BIER/BIFT forwarding can be amended to support basic BIER-TE forwarding, by using BIER BIFT's F-BM. Only the masking of bits due to avoid duplicates must be skipped when forwarding is for BIER-TE.

Whether to use BIER or BIER-TE forwarding can simply be a configured choice per subdomain and accordingly be set up by a BIER-TE controller host. The BIER packet encapsulation [RFC8296] too can be reused without changes except that the currently defined BIER-TE ECMP adjacency does not leverage the entropy field so that field would be unused when BIER-TE forwarding is used.

3.6. Requirements

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

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

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

4. BIER-TE Controller Host BitPosition Assignments

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

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

4.1. P2P Links

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

4.2. BFER

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

4.3. Leaf BFERs

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

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

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

4.4. LANs

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

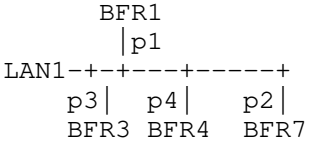


Figure 6: LAN Example

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

a list of forward_connected adjacencies to all other neighbors on the LAN.

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

4.5. Hub and Spoke

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

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

4.6. Rings

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

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

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

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

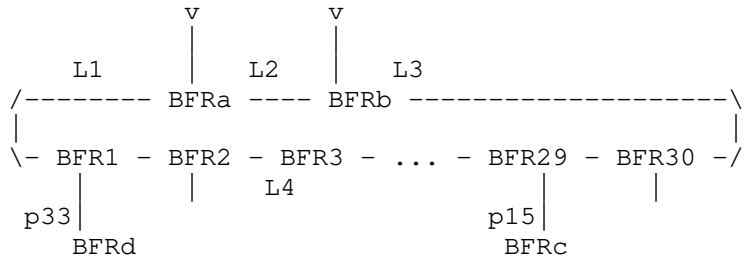


Figure 7: Ring Example

Note that this example only permits for packets to enter the ring at BFRa and BFRb, and that packets will always travel clockwise. If packets should be allowed to enter the ring at any ring BFR, then one would have to use two ring BitPositions. One for clockwise, one for counterclockwise.

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

4.7. Equal Cost MultiPath (ECMP)

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

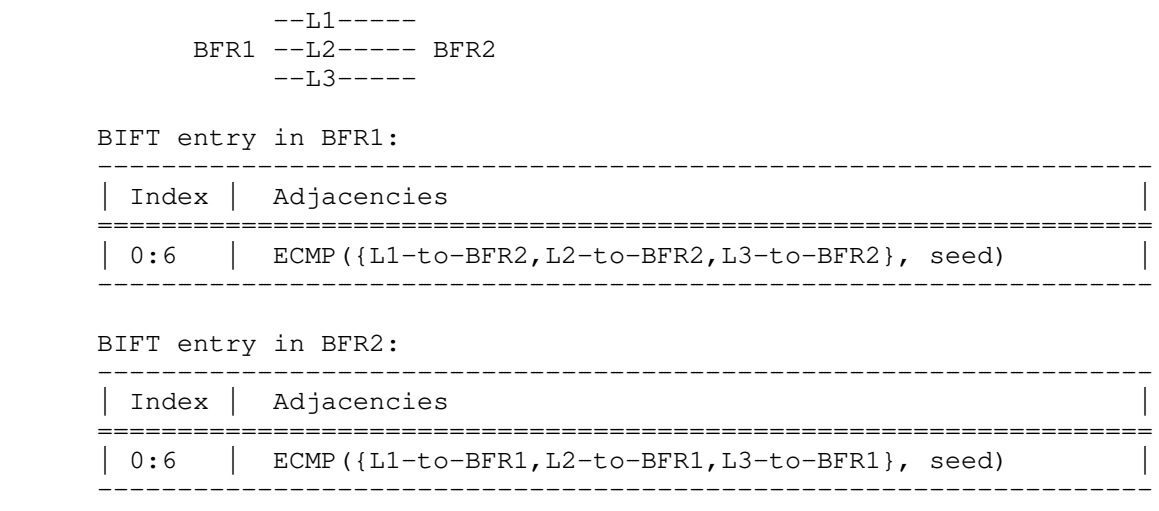
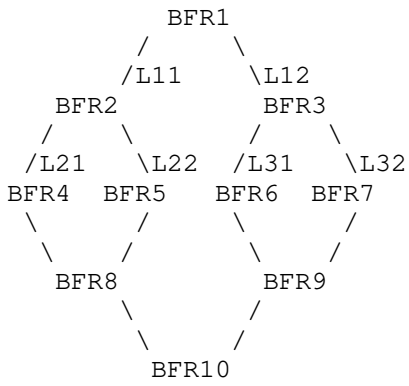


Figure 8: ECMP Example

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



BIFT entry in BFR1:

0:6	ECMP({L11-to-BFR2,L12-to-BFR3}, seed)

BIFT entry in BFR2:

0:6	ECMP({L21-to-BFR4,L22-to-BFR5}, seed)

BIFT entry in BFR3:

0:6	ECMP({L31-to-BFR6,L32-to-BFR7}, seed)

Figure 9: Polarization Example

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

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

This issue is called polarization. It depends on the ECMP hash. It is possible to build ECMP that does not have polarization, for example by taking entropy from the actual adjacency members into account, but that can make it harder to achieve evenly balanced load-

splitting on all BFR without making the ECMP hash algorithm potentially too complex for fast forwarding in the BFRs.

4.8. Routed adjacencies

4.8.1. Reducing BitPositions

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

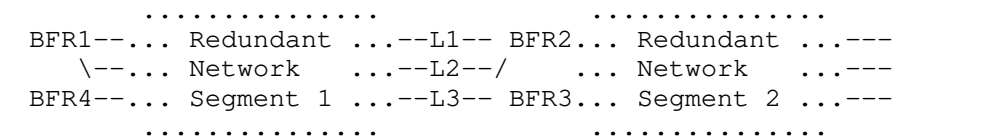


Figure 10: Routed Adjacencies Example

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

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

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

4.8.2. Supporting nodes without BIER-TE

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

5. Avoiding loops and duplicates

5.1. Loops

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

With DNR set, looping can happen. Consider in the ring picture that link L4 from BFR3 is plugged into the L1 interface of BFRa. This creates a loop where the rings clockwise BitPosition is never reset for copies of the packets traveling clockwise around the ring.

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

5.2. Duplicates

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

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

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

6. BIER-TE Forwarding Pseudocode

The following simplified pseudocode for BIER-TE forwarding is using BIER forwarding pseudocode of [RFC8279], section 6.5 with the one modification necessary to support basic BIER-TE forwarding. Like the BIER pseudo forwarding code, for simplicity it does hide the details of the adjacency processing inside PacketSend() which can be forward_connected, forward_routed or local_decap.

```
void ForwardBitMaskPacket_withTE (Packet)
{
    SI=GetPacketSI(Packet);
    Offset=SI*BitStringLength;
    for (Index = GetFirstBitPosition(Packet->BitString); Index ;
         Index = GetNextBitPosition(Packet->BitString, Index)) {
        F-BM = BIFT[Index+Offset]->F-BM;
        if (!F-BM) continue;
        BFR-NBR = BIFT[Index+Offset]->BFR-NBR;
        PacketCopy = Copy(Packet);
        PacketCopy->BitString &= F-BM;                    [2]
        PacketSend(PacketCopy, BFR-NBR);
        // The following must not be done for BIER-TE:
        // Packet->BitString &= ~F-BM;                    [1]
    }
}
```

Figure 11: Simplified BIER-TE Forwarding Pseudocode

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

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

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

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

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

The following pseudocode is comprehensive:

- This pseudocode eliminates per-bit F-BM, therefore reducing state by BitStringLength^2*SI and eliminating the need for per-packet-copy masking operation except for adjacencies with DNR flag set:
  - AdjacentBits[SI] are bits with a non-empty list of adjcencies. This can be computed whenever the BIER-TE controller host updates the adjacencies.
  - Only the AdjacentBits need to be examined in the loop for packet copies.
  - The packets BitString is masked with those AdjacentBits on ingres to avoid packet loopings.
- The code loops over the adjacencies because there may be more than one adjacency for a bit.
- When an adjacency has the DNR bit, the bit is set in the packet copy (to save bits in rings for example).
- The ECMP adjacency is shown. Its parameters are a ListOfAdjacencies from which one is picked.
- The forward_local, forward_routed, local_decap adjacencies are shown with their parameters.

```
void ForwardBitMaskPacket_withTE (Packet)
{
    SI=GetPacketSI(Packet);
    Offset=SI*BitStringLength;
    AdjacentBitstring = Packet->BitString &= ~AdjacentBits[SI];
    Packet->BitString &= AdjacentBits[SI];
    for (Index = GetFirstBitPosition(AdjacentBits); Index ;
         Index = GetNextBitPosition(AdjacentBits, Index)) {
        foreach adjacency BIFT[Index+Offset] {
            if(adjacency == ECMP(ListOfAdjacencies, seed) ) {
                I = ECMP_hash(sizeof(ListOfAdjacencies),
                              Packet->Entropy, seed);
                adjacency = ListOfAdjacencies[I];
            }
            PacketCopy = Copy(Packet);
            switch(adjacency) {
                case forward_connected(interface,neighbor,DNR):
                    if(DNR)
                        PacketCopy->BitString |= 2<<(Index-1);
                    SendToL2Unicast(PacketCopy,interface,neighbor);

                case forward_routed([VRF],neighbor):
                    SendToL3(PacketCopy,[VRF,]l3-neighbor);

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

Figure 12: BIER-TE Forwarding Pseudocode

7. Managing SI, subdomains and BFR-ids

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

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

7.1. Why SI and sub-domains

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

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

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

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

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

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

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

7.2. Bit assignment comparison BIER and BIER-TE

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

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

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

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

7.3. Using BFR-id with BIER-TE

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

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

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

"Independent branches": For a given application or (set of) trees, the branches from a BFIR to every BFER are independent of the

branches to any other BFER. For example, shortest part trees have independent branches.

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

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

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

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

7.4. Assigning BFR-ids for BIER-TE

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

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

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

It is not currently determined if a single subdomain could or should be allowed to forward both BIER and BIER-TE packets. If this should be supported, there are two options:

A. BIER and BIER-TE have different BFR-id in the same subdomain. This allows higher replication efficiency for BIER because their BFR-id can be assigned sequentially, while the bitstrings for BIER-TE will have also the additional bits for the topology. There is no relationship between a BFR BIER BFR-id and BIER-TE BFR-id.

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

7.5. Example bit allocations

7.5.1. With BIER

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

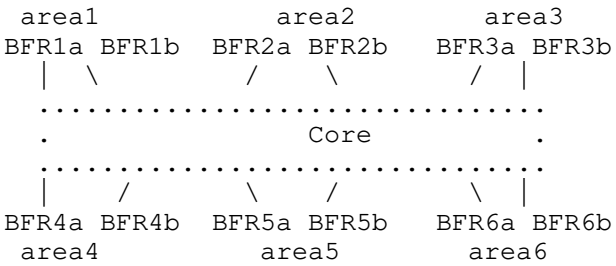


Figure 13: Scaling BIER-TE bits by reuse

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

If BFR-id are allocated intelligently, then all the BFER in an area would be given BFR-id with as few as possible different SI. Each area would only have to forward one or two packets instead of 4.

Given how networks can grow over time, replication efficiency in an area will also easily go down over time when BFR-id are network wide allocated sequentially over time. An area that initially only has BFR-id in one SI might end up with many SI over a longer period of growth. Allocating SIs to areas with initially sufficiently many spare bits for growths can help to alleviate this issue. Or renumber BFR-id after network expansion. In this example one may consider to use 6 SI and assign one to each area.

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

7.5.2. With BIER-TE

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

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

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

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

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

In addition, we use 4 bits in each SI: bia, bib, bea, beb: bit ingress a, bit ingress b, bit egress a, bit egress b. These bits will be used to pass BIER packets from any BFIR via any combination of ingress area a/b BFR and egress area a/b BFR into a specific target area. These bits are then set up with the right forward_routed adjacencies on the BFIR and area edge BFR:

On all BFIR in an area j, bia in each BIFT:SI is populated with the same forward_routed(BFRja), and bib with forward_routed(BFRjb). On all area edge BFR, bea in BIFT:SI=k is populated with forward_routed(BFRka) and beb in BIFT:SI=k with forward_routed(BFRkb).

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

7.6. Summary

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

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

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

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

8. BIER-TE and Segment Routing

Segment Routing aims to achieve lightweight path engineering via loose source routing. Compared for example to RSVP-TE, it does not require per-path signaling to each of these hops.

BIER-TE is 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.

Instead of defining BitPositions for non-replicating hops, it is equally possible to use segment routing encapsulations (eg: MPLS label stacks) for "forward_routed" adjacencies.

Note that BIER itself is also similar to SR - it achieves the same as "Shortest Path SID" where the label stack uses only one SID to indicate the egres node of the SR domain. Instead of routing such a SR packet hop-by-hop based on that SID, BIER routes the packet hop-by-hop based on the BFER-id bits of the egres nodes of the BIER domain. What BIER does not allow is to indicate intermediate hops, or terms of SR lavbel stacks with more than one SID in the stack (for the same SR domain). This is what BIER-TE provides.

9. Security Considerations

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

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

10. IANA Considerations

This document requests no action by IANA.

11. Acknowledgements

The authors would like to thank Greg Shepherd, Ijsbrand Wijnands and Neale Ranns for their extensive review and suggestions.

12. Change log [RFC Editor: Please remove]

draft-ietf-bier-te-arch:

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 pseodcode that mirrors the BIER forwarding pseudocode to make comparison easier and a second pseudocode that shows the complete set of BIER-TE

forwarding options and simplification/optimization possible vs. BIER forwarding.

- Added captions to pictures.

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

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

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

draft-eckert-bier-te-arch:

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

05: Author affiliation change only.

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

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

- Linked FRR information to new draft in Overview/Introduction

- Removed BTAFT/FRR from "Changes in the network topology"

- Linked new draft in "Link/Node Failures and Recovery"

- Removed FRR from "The BIER-TE Forwarding Layer"

- Moved FRR section to new draft

- Moved FRR parts of Pseudocode into new draft

- Left only non FRR parts

- removed FrrUpDown(..) and //FRR operations in ForwardBierTePacket(..)

- New draft contains FrrUpDown(..) and ForwardBierTePacket(Packet) from bier-arch-03

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

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

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

- Added Changes for version 04

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

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

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

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

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

01: Fixed BFIR -> BFER for section 4.3.

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

00: Initial version.

13. References

[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>.

[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

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

Email: tte+ietf@cs.fau.de

Gregory Cauchie
Bouygues Telecom

Email: GCAUCHIE@bouyguestelecom.fr

Wolfgang Braun
University of Tuebingen

Email: wolfgang.braun@uni-tuebingen.de

Michael Menth
University of Tuebingen

Email: menth@uni-tuebingen.de

BIER
Internet-Draft
Intended status: Standards Track
Expires: December 30, 2018

Shaofu. Peng
Zheng. Zhang
ZTE Corporation

June 28, 2018

Global vpnid advertisement in BIER overlay
draft-pengzhang-bier-global-vpnid-00

Abstract

This document specifies a method to achieve multipoint VPN interconnection through a BIER domain.

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 December 30, 2018.

Copyright Notice

Table of Contents

1. Terminology 2
2. Problem statement 2
3. Solution 3
3.1. Advertisement 3
3.2. Encapsulation 4
3.3. Decapsulation 4
3.4. Formats 4
4. IANA Considerations 5
5. Security Considerations 5
6. Normative References 5
Authors' Addresses 6

1. Terminology

This document uses terminologies defined in [RFC8279], [RFC6513], [RFC6514], [I-D.ietf-bier-mld].

2. Problem statement

BIER (Bit Indexed Explicit Replication) [RFC8279] introduces an architecture for the forwarding of multicast data packet. It provides optimal forwarding of multicast packet through a 'multicast domain'. It does not require explicitly building multicast distribution trees, nor does require intermediate nodes to maintain any per-flow state.

BIER MVPN [I-D.ietf-bier-mvpn] introduces a method which using BIER as multicast tunnels (P-tunnels) to carry multicast traffic across the BIER domain. The advertising method from [RFC6513] and [RFC6514] is general and flexible, but it is complicated in some situations at the same time because of the program of many parameters, like RD, RT, etc. In many situations which only interconnect different sites across a domain, the comprehensive MVPN configuration increases the network administrative complication.

In the other hand, BIER MVPN using upstream assigned label to indicate the corresponding multicast flow in a MVPN. The pair of ingress PE and upsteam assigned labels increases label administration and flow forwarding complication.
[I-D.zzhang-bess-mvpn-evpn-aggregation-label] arises a discussion about using common label assigned by controller in MVPN. But in a

network without a controller, it is still a problem to achieve the multipoint interconnection without MVPN configuration.

So for the networks that need flow isolation across domain but do not need complicated configuration, this document specifies a method to achieve multipoint VPN interconnection across a BIER domain by advertising global vpn-id in BIER forwarding overlay, and defines encapsulation and forwarding functions to carry and execute the global vpn-id. It is similar as the usage of VNI-VSID in case of EVPN VXLAN/ NVGRE described in [I-D.ietf-bier-evpn].

3. Solution

The multipoint VPN here means some flows should be forwarded to multiple edge routers across a domain. In the simple multipoint interconnection situations that does not deploy MVPN configuration like RD, RT, etc., a global vpn-id is used to indicate the corresponding VPN. This global vpn-id is encapsulated between BIER header and actual data packet. The BIER forwarding function is also modified to execute this kind of packet.

3.1. Advertisement

BIER overlay protocols include BMLD [I-D.ietf-bier-mld], MVPN [I-D.ietf-bier-mvpn], and PIM [I-D.ietf-bier-pim-signaling], EVPN [I-D.ietf-bier-evpn]. Global vpn-id extension should be added in these BIER overlay protocols by a TLV format. When using BGP as BIER overlay protocol to advertise global vpn-id, specific VPN parameters like RD, RT defined in [RFC6513] and [RFC6514] need not be used.

A BIER domain edge router can belong to several VPNs. A unique global vpn-id is assigned to a particular VPN. An edge router belongs to several VPNs is assigned several global vpn-ids.

Edge routers belong to a same VPN should be assigned a same global vpn-id. The two edge routers which have same global vpn-id indicates that the two routers belong to a same particular VPN.

When BIER domain edge routers exchange BIER overlay information, the edge routers belong to one or more VPNs should advertise the corresponding global vpn-ids extension.

After a router receives global vpn-id extensions from the other edge routers, the router MUST store the edge routers which have same global vpn-ids with local VPNs.

The router SHOULD store the edge routers which have different global vpn-ids with local VPNs in order to increase converged efficiency that caused by configuration modification.

3.2. Encapsulation

After ingress router gathers the information of edge routers which have same global vpn-ids, ingress router generates forwarding items which include global vpn-id and BFR-ids of egress routers.

When ingress router encapsulates the data packet which should be sent to the egress routers according to a global vpn-id, the value of global vpn-id MUST be added between BIER header and actual data packet. The encapsulation function is the same as [RFC8296], the 'Proto' field in BIER header should be set to the value for a new type of global vpn-id.

The forwarding of intermediate routers is unchanged according to the forwarding function defined in [RFC8279].

3.3. Decapsulation

Finally the packet reaches egress routers. Egress router looks for the forwarding items indexed by the global vpn-id according to the 'Proto' field in BIER header. After decapsulation, egress router forwards data packet to corresponding local receivers.

3.4. Formats

[RFC2685] defines a globally unique VPN identifier to connect same VPN in different sites. The format of global vpn-id defined in [RFC2685] is 7 octets. But in actually deployment, a global vpn-id with 20 bits is enough to indicate the corresponding VPN. So the global vpn-id can be used as BIFT-ID defined in [RFC8296]directly.

When MLD protocol is used as BIER overlay, a new type of TLV is added in BMLD report messages.

When BGP protocol is used as BIER overlay, a new type of TLV is added in BGP update message.

When PIM protocol is used as BIER overlay, a new type of TLV is added in PIM join/ prune messages.

For the edge routers which act as ingress routers or egress routers, the corresponding global vpn-ids are carried in the new TLV. And the BFR-id of the router itself is also included in the TLV.

4. IANA Considerations

A new type which indicates the global vpn-id should be added in BIER 'Proto' assignment. A new type of global vpn-id extension should be added in each BIER overlay protocols, includes MLD, PIM, BGP.

5. Security Considerations

There is no further security requirements in this document.

6. Normative References

[I-D.ietf-bier-evpn]
Zhang, Z., Przygienda, T., Sajassi, A., and J. Rabadan, "EVPN BUM Using BIER", draft-ietf-bier-evpn-01 (work in progress), April 2018.

[I-D.ietf-bier-mld]
Pfister, P., Wijnands, I., Venaas, S., Wang, C., Zhang, Z., and M. Stenberg, "BIER Ingress Multicast Flow Overlay using Multicast Listener Discovery Protocols", draft-ietf-bier-mld-00 (work in progress), June 2017.

[I-D.ietf-bier-mvpn]
Rosen, E., Sivakumar, M., Aldrin, S., Dolganow, A., and T. Przygienda, "Multicast VPN Using BIER", draft-ietf-bier-mvpn-11 (work in progress), March 2018.

[I-D.ietf-bier-pim-signaling]
Bidgoli, H., Dolganow, A., Kotalwar, J., Xu, F., mishra, m., and Z. Zhang, "PIM Signaling Through BIER Core", draft-ietf-bier-pim-signaling-03 (work in progress), June 2018.

[I-D.zzhang-bess-mvpn-evpn-aggregation-label]
Zhang, Z., Rosen, E., Lin, W., Li, Z., and I. Wijnands, "MVPN/EVPN Tunnel Aggregation with Common Labels", draft-zzhang-bess-mvpn-evpn-aggregation-label-01 (work in progress), April 2018.

[RFC2685] Fox, B. and B. Gleeson, "Virtual Private Networks Identifier", RFC 2685, DOI 10.17487/RFC2685, September 1999, <https://www.rfc-editor.org/info/rfc2685>.

[RFC6513] Rosen, E., Ed. and R. Aggarwal, Ed., "Multicast in MPLS/BGP IP VPNs", RFC 6513, DOI 10.17487/RFC6513, February 2012, <https://www.rfc-editor.org/info/rfc6513>.

[RFC6514] Aggarwal, R., Rosen, E., Morin, T., and Y. Rekhter, "BGP Encodings and Procedures for Multicast in MPLS/BGP IP VPNs", RFC 6514, DOI 10.17487/RFC6514, February 2012, <https://www.rfc-editor.org/info/rfc6514>.

[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

Shaofu Peng
ZTE Corporation

EMail: peng.shaofu@zte.com.cn

Zheng(Sandy) Zhang
ZTE Corporation

EMail: zzhang_ietf@hotmail.com

Network Working Group
Internet-Draft
Intended status: Informational
Expires: April 21, 2019

D. Purkayastha
A. Rahman
D. Trossen
InterDigital Communications, LLC
T. Eckert
Huawei

October 18, 2018

Applicability of BIER Multicast Overlay for Adaptive Streaming Services draft-purkayastha-bier-multicast-http-response-01

Abstract

HTTP Level multicast, using BIER, is described as a use case in 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. A realization of "HTTP Multicast" over "IP Multicast" is described. IP multicast is commonly used for IPTV services. DVB and BBF is also developing a reference architecture for IP Multicast service. 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. 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.

Table of Contents

1. Introduction 2
1.1. Reference Deployment 3
2. Conventions used in this document 5
3. Use cases 5
4. Requirements 6
5. Realization over IP Multicast 6
5.1. Mapping to Requirements 7
5.2. Problems 8
6. Realization over BIER 8
6.1. Description of a "BIER Multicast Overlay" to support HTTP Multicast 9
6.1.1. BIER Multicast Overlay Components 9
6.1.2. BIER Multicast Overlay Operations 10
6.2. Achieving Multicast Responses 12
6.3. BIER multicast Overlay Traffic Management 13
7. Next Steps 13
8. IANA Considerations 14
9. Security Considerations 14
10. Informative References 14
Authors' Addresses 15

1. Introduction

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

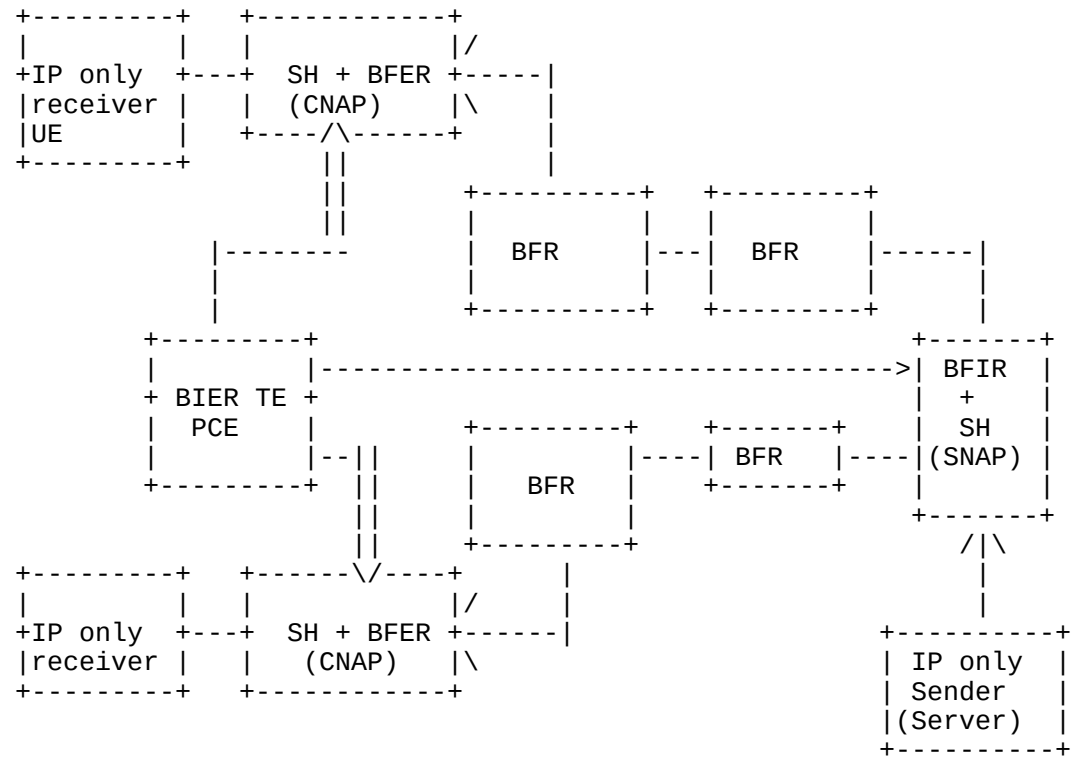
client requests are converted into the BIER or BIER-TE bitstring and used to send the HTTP response.

In this draft, we describe how this class of use cases can be realized over IP Multicast and how the operation of the 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.

We conclude with future steps and seek input from the WG.

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 sends and receives HTTP transactions through CNAP.

On the server side, the Service handling function can be part of the Server Network Attachment Point (SNAP). It includes the BFIR function and SH. SNAP is responsible for aggregating the relevant

HTTP Requests and sending one or more BIER Multicast HTTP response to multiple clients who requested the same content.

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

As part of POINT/RIFE EU Horizon 2020 project, 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 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.

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.

HTTP request and response used in media streaming services like HLS, use HTTP response for delivery of content. 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.

This solution 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.

4. Requirements

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

- MUST support multiple FQDN-based service endpoints to exist in the overlay
- MUST send FQDN-based service requests at the network level to a suitable FQDN-based service endpoint via policy-based selection of appropriate path information
- MUST allow for multicast delivery of HTTP response to same HTTP request URI
- MUST provide direct path mobility, where the path between the egress and ingress Service Routers(SR) can be determined as being optimal (e.g., shortest path or direct path to a selected instance), is needed to avoid the use of anchor points and further reduce service-level latency

5. Realization over IP Multicast

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

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

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

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

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

5.1. Mapping to Requirements

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

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

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

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

5.2. Problems

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

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

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

6. Realization over BIER

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

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

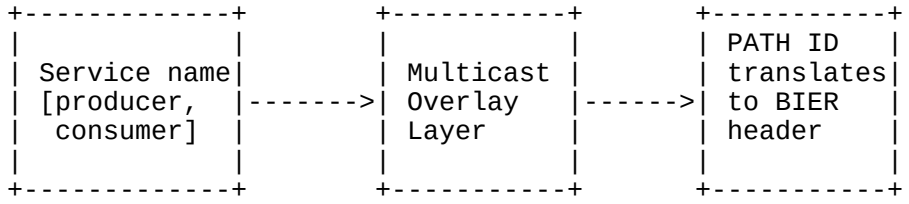


Figure 2: Service name to Path ID translation

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

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

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

6.1.1. BIER Multicast Overlay Components

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

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

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

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

6.1.2. BIER Multicast Overlay Operations

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

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

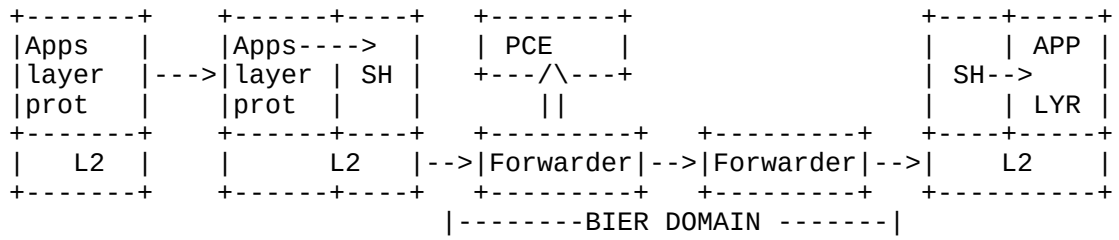


Figure 3: Protocol for Multicast Overlay Layer

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.

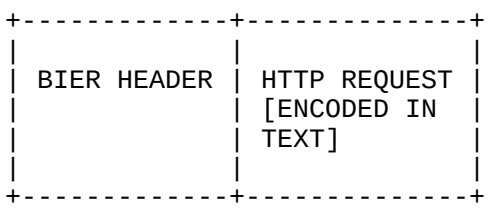


Figure 4: Encapsulation of Service Request

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.

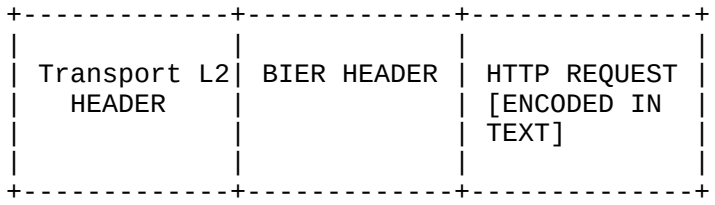


Figure 5: Transport Encapsulation of BIER payload

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. Next Steps

This Applicability Statement document describes how HTTP multicast responses can be realized over BIER. This document describes the functionalities in the multicast overlay layer to enable this functionality. We would like to get feedback and support from the WG to continue this work. We will elaborate further on specific protocols for the overlay layer and request adoption as a WG draft.

8. IANA Considerations

This document requests no IANA actions.

9. 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. Future updates to this draft will outline the support for such HTTPS-based exchanges.

10. 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_blue book.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-07 (work in progress), July 2018.

[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-04 (work in progress), September 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>.

[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

Debashish Purkayastha
InterDigital Communications, LLC
Conshohocken
USA

Email: Debashish.Purkayastha@InterDigital.com

Akbar Rahman
InterDigital Communications, LLC
Montreal
Canada

Email: Akbar.Rahman@InterDigital.com

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

Email: Dirk.Trossen@InterDigital.com
URI: http://www.InterDigital.com/

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

Email: tte+ietf@cs.fau.de

Network Working Group
Internet-Draft
Intended status: Experimental
Expires: April 19, 2019

S. Venaas
IJ. Wijnands
L. Ginsberg
Cisco Systems, Inc.
M. Sivakumar
Juniper Networks

October 16, 2018

BIER MTU Discovery
draft-venaas-bier-mtud-02

Abstract

This document defines an IGP based mechanism for discovering the MTU of a BIER sub-domain. This document defines extensions to OSPF and IS-IS, but other protocols could potentially be extended. MTU discovery is usually done for a given path, while this document defines it for a sub-domain. This allows the computed MTU to be independent of the set of receivers. Also, the MTU is independent of rerouting events within the sub-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 April 19, 2019.

Copyright Notice

Table of Contents

1. Introduction 2
2. Conventions used in this document 3
3. MTU discovery procedure 3
4. IS-IS BIER Sub-Domain MTU Sub-sub-TLV 4
5. OSPF BIER Sub-Domain MTU Sub-TLV 5
6. IANA considerations 5
7. Acknowledgments 5
8. References 5
8.1. Normative References 6
8.2. Informative References 6
Authors' Addresses 6

1. Introduction

This document defines an IGP based mechanism for discovering the MTU of a BIER sub-domain. The discovered MTU indicates the largest possible BIER packet that can be sent across any link in a BIER sub-domain. This is different from [I-D.ietf-bier-path-mtu-discovery] which performs Path MTU Discovery (PMTUD) for a set of receivers. PMTUD is based on probing, and when there are routing changes, e.g., a link going down, the actual MTU for a path may become less than was previously discovered, and there will be some delay until the next probe is performed. Also, the set of receivers for a flow may change at any time, which may cause the MTU to change. This document instead discovers a BIER sub-domain MTU, which is independent of paths and receivers within the sub-domain.

Discovering the sub-domain MTU is much simpler than discovering the multicast path MTU, and is more robust with regards to path changes as discussed above. However, the sub-domain MTU may be a lot smaller than the path MTU would have been for a given flow. The discovery mechanisms may be combined, allowing the discovery of the path MTU for certain flows as needed.

The BIER sub-domain MTU defined here provides the maximum size of a BIER packet that can be forwarded through the sub-domain regardless of path. A BIER router that performs BIER encapsulation will need to subtract the encapsulation overhead to find the largest size packet that can be encapsulated. This would give the IP MTU, and may be

used for IP PMTUD by for instance sending an ICMP Packet too big message if an IP packet will be too large after BIER encapsulation.

2. Conventions used in this document

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

3. MTU discovery procedure

An interface on a router is said to be a BIER interface if the router has a BIER neighbor on the interface. That is, there is a directly connected router on that interface that is announcing a BIER prefix. Further, the BIER interface is said to belong to a given sub-domain if the router itself announces a prefix tagged with the sub-domain, and there is BIER neighbor on the interface also announcing a prefix tagged with the sub-domain.

The BIER MTU of an interface is the largest BIER packet that can be sent out of the interface. Further, the local sub-domain MTU of a router is the minimum of all the BIER MTUs of the BIER interfaces in the sub-domain. Note that the local sub-domain MTU of a router is only defined if it has at least one BIER interface in the sub-domain.

A BIER router announces a BIER prefix in either IS-IS or OSPF as specified in [RFC8401] and [I-D.ietf-bier-ospf-bier-extensions]. They both define a BIER Sub-TLV to be included with the prefix. There is one BIER Sub-TLV included for each sub-domain. This document defines how a router includes its local sub-domain MTU in each of the BIER Sub-TLVs it advertizes.

A router can discover the MTU of a BIER sub-domain by identifying all the prefixes that have a BIER Sub-TLV for the sub-domain. It then computes the minimum of the advertised MTU values for that sub-domain. This includes its own local sub-domain MTU. This allows all the routers in the sub-domain to discover the same sub-domain wide MTU.

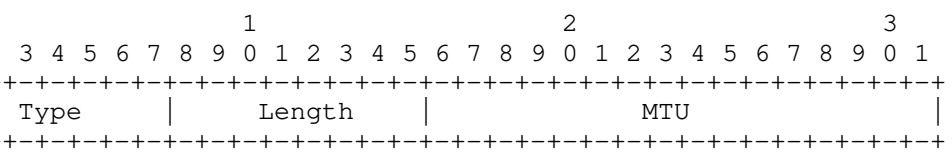
Note that a router should announce a new local MTU for a sub-domain immediately if the value becomes smaller than what it currently announces. This would happen if the MTU of an interface is configured to a smaller value, or the first BIER neighbor for a sub-domain is detected on an interface, and the MTU of the interface is less than all the other local BIER interfaces in the sub-domain. However, if BIER neighbors go away, or if an interface goes down, so

that the local MTU becomes larger, a router SHOULD NOT immediately announce the larger value. A router MAY after some delay announce the new larger MTU. The intention is that dynamic events such as a quick link flap should not cause the announced MTU to be increased.

It is a concern that the sub-domain MTU will be based on the link with the smallest MTU. This means that if for instance a single link is accidentally configured with an extra small MTU, it will impact the sub-domain MTU and potentially all the flows through the sub-domain. As an example, an administrator might decide to use jumbo frames and has configured that on all the links. But accidentally forget to configure it on a new link before it is brought up. To provide some protection against this, an implementation SHOULD provide a configurable minimum BIER sub-domain MTU. When this is configured, the MTU discovery is still done according to the above procedure, but if the resulting MTU value is less than the configured minimum, the configured minimum MUST be used instead. If the discovery procedure later should provide an MTU larger than the minimum, then the discovered MTU MUST be used. An implementation SHOULD provide notification to the administrator when the discovered MTU is less than the minimum, as this is likely a configuration mistake that should be corrected.

4. IS-IS BIER Sub-Domain MTU Sub-sub-TLV

A router uses the BIER Sub-Domain MTU Sub-sub-TLV to announce the minimum BIER MTU of all its BIER enabled interfaces in a sub-domain. The BIER Sub-Domain MTU is the largest BIER packet that can be sent out of all the interfaces in a sub-domain. The Sub-sub-TLV MUST be ignored if it is included multiple times.



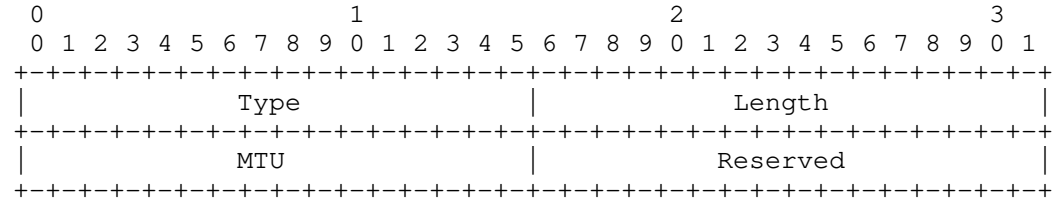
Type: TBD

Length: 2

MTU: MTU in octets

5. OSPF BIER Sub-Domain MTU Sub-TLV

A router uses the BIER Sub-Domain MTU Sub-TLV to announce the minimum BIER MTU of all its BIER enabled interfaces in a sub-domain. The BIER Sub-Domain MTU is the largest BIER packet that can be sent out of all the interfaces in a sub-domain. The Sub-TLV MUST be ignored if it is included multiple times.



Type: TBD2

Length: 4

MTU: MTU in octets

6. IANA considerations

An allocation from the "sub-sub-TLVs for BIER Info sub-TLV" registry as defined in [RFC8401] is requested for the IS-IS BIER Sub-Domain MTU Sub-sub-TLV. Please replace the string TBD in this document with the appropriate value.

An allocation from the "OSPF Extended Prefix sub-TLV" registry as defined in [RFC7684] is requested for the OSPF BIER Sub-Domain MTU Sub-TLV. Please replace the string TBD2 in this document with the appropriate value.

7. Acknowledgments

The authors would like to thank Greg Mirsky in particular for fruitful discussions and input. Valuable comments were also provided by Alia Atlas, Eric C Rosen, Toerless Eckert, Tony Przygienda and Xie Jingrong.

8. References

8.1. Normative References

[I-D.ietf-bier-ospf-bier-extensions]
Psenak, P., Kumar, N., Wijnands, I., Dolganow, A., Przygienda, T., Zhang, Z., and S. Aldrin, "OSPFv2 Extensions for BIER", draft-ietf-bier-ospf-bier-extensions-18 (work in progress), June 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>.

[RFC7684] Psenak, P., Gredler, H., Shakir, R., Henderickx, W., Tantsura, J., and A. Lindem, "OSPFv2 Prefix/Link Attribute Advertisement", RFC 7684, DOI 10.17487/RFC7684, November 2015, <https://www.rfc-editor.org/info/rfc7684>.

[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>.

[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>.

8.2. Informative 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-04 (work in progress), June 2018.

Authors' Addresses

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

Email: stig@cisco.com

IJsbrand Wijnands
Cisco Systems, Inc.
De kleetlaan 6a
Diegem 1831
Belgium

Email: ice@cisco.com

Les Ginsberg
Cisco Systems, Inc.
Tasman Drive
San Jose CA 95134
USA

Email: ginsberg@cisco.com

Mahesh Sivakumar
Juniper Networks
1133 Innovation Way
Sunnyvale CA 94089
USA

Email: sivakumar.mahesh@gmail.com

Network Working Group
Internet-Draft
Intended status: Experimental
Expires: April 25, 2019

S. Venaas
IJ. Wijnands
M. Mishra
Cisco Systems, Inc.
M. Sivakumar
Juniper Networks

October 22, 2018

PIM Flooding Mechanism and Source Discovery for BIER
draft-venaas-bier-pfm-sd-00

Abstract

PIM Flooding Mechanism and Source Discovery (PFM-SD) is a mechanism for source discovery within a PIM domain. PIM signaling over BIER has been defined, allowing for BIER to interoperate with PIM. This document defines PFM-SD over BIER, such that PFM-SD can be used by PIM in a PIM domain to discover sources that are reachable via BIER. Also, this document provides PFM-SD extensions to discover the BIER ingress router closest to the source. This can be used by BIER overlays, such as PIM signaling over BIER, to determine which router to signal.

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 25, 2019.

Table of Contents

1. Introduction 2
1.1. Conventions Used in This Document 3
2. PFM over BIER 3
3. PFM Ingress BIER Router TLV 3
4. Group Source Holdtime Metric TLV 4
5. BIER signaling enhancements 6
6. Security Considerations 6
7. IANA Considerations 6
8. References 6
8.1. Normative References 6
8.2. Informative References 7
Authors' Addresses 7

1. Introduction

PIM Flooding Mechanism (PFM) and Source Discovery (SD) [RFC8364] provides a generic flooding mechanism for distributing information throughout a PIM domain. In particular it allows for source discovery. There are various deployment scenarios where PIM and BIER need to co-exist. For instance, consider migration scenarios where a few routers in a PIM domain are upgraded to support BIER. In that case, one may use PIM Signaling Through BIER Core [I-D.ietf-bier-pim-signaling], allowing PIM to build trees passing through the BIER routers. This document defines PFM over BIER. This allows PFM to pass through the BIER routers, allowing PFM to be used in the PIM domain.

One challenge with PIM signaling over BIER [I-D.ietf-bier-pim-signaling] is to determine which BIER router is closest to the source. A number of options are discussed in that document. This document provides an alternative solution for discovering which BIER router to signal. It may also be used with other signaling mechanisms such as IGMP/MLD [I-D.ietf-bier-mld]. This is achieved by introducing two new PFM TLVs. When a BIER router forwards a PFM message into BIER, it adds a new TLV specifying the BIER sub-domain, its BFR-ID and its BIER prefix. Also, any Group Source Holdtime TLVs, defined in [RFC8364], are replaced with new TLVs that include the router's cost of reaching the sources.

1.1. 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].

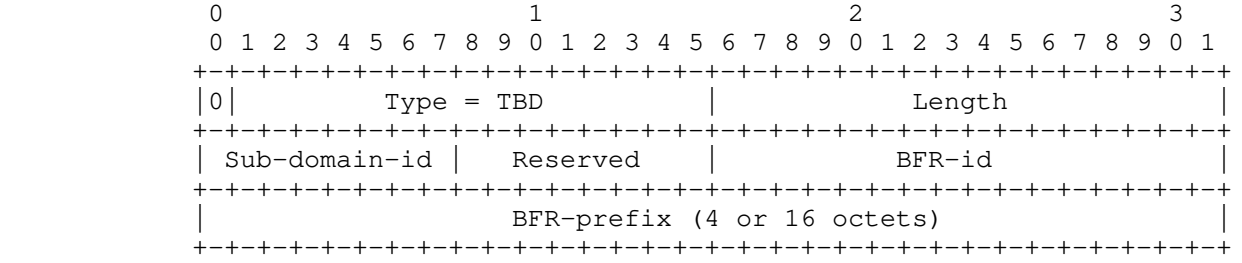
2. PFM over BIER

When a BIER enabled router accepts a PFM message from a PIM neighbor according to [RFC8364], it SHOULD in addition to the forwarding defined in [RFC8364], also send a copy to all BIER routers (an implementation SHOULD allow the set of BIER routers to send PFM messages to, to be configured).

When a router receives a BIER encapsulated PFM message, it MUST process the message according to [RFC8364], except there is no requirement for the message to come from a PIM neighbor, and there is no RPF check. The message MUST be forwarded out on the PIM interfaces according to [RFC8364]. It MAY also be BIER forwarded, if the router acts as a border router between BIER domains.

3. PFM Ingress BIER Router TLV

When a router is forwarding a PFM message into a BIER domain, it MUST add this TLV. If the TLV is already present, all occurrences should be removed. This TLV encodes the BIER prefix, sub-domain ID and BFR-ID of the router. This TLV SHOULD only be present within the BIER domain. When a router receives a PFM message with this TLV, all occurrences of the TLV SHOULD be removed. If the router is forwarding the message into a new BIER domain, it should add a new TLV with its own prefix, sub-domain ID and BFR-ID. A PFM message is expected to have at most one such TLV. A router MUST NOT add more than one such TLV. When forwarding a PFM message, the TLV in the received message MUST be removed from the forwarded message.



0: The Transitive bit is set to 0.

Type: Type is TBD.

Length: The length of the value in octets.

Sub-domain-id: The ID of the sub-domain that this PFM is forwarded into. The length is 1 octet.

Reserved: MUST be set to 0, and ignored when received. The length is 1 octet.

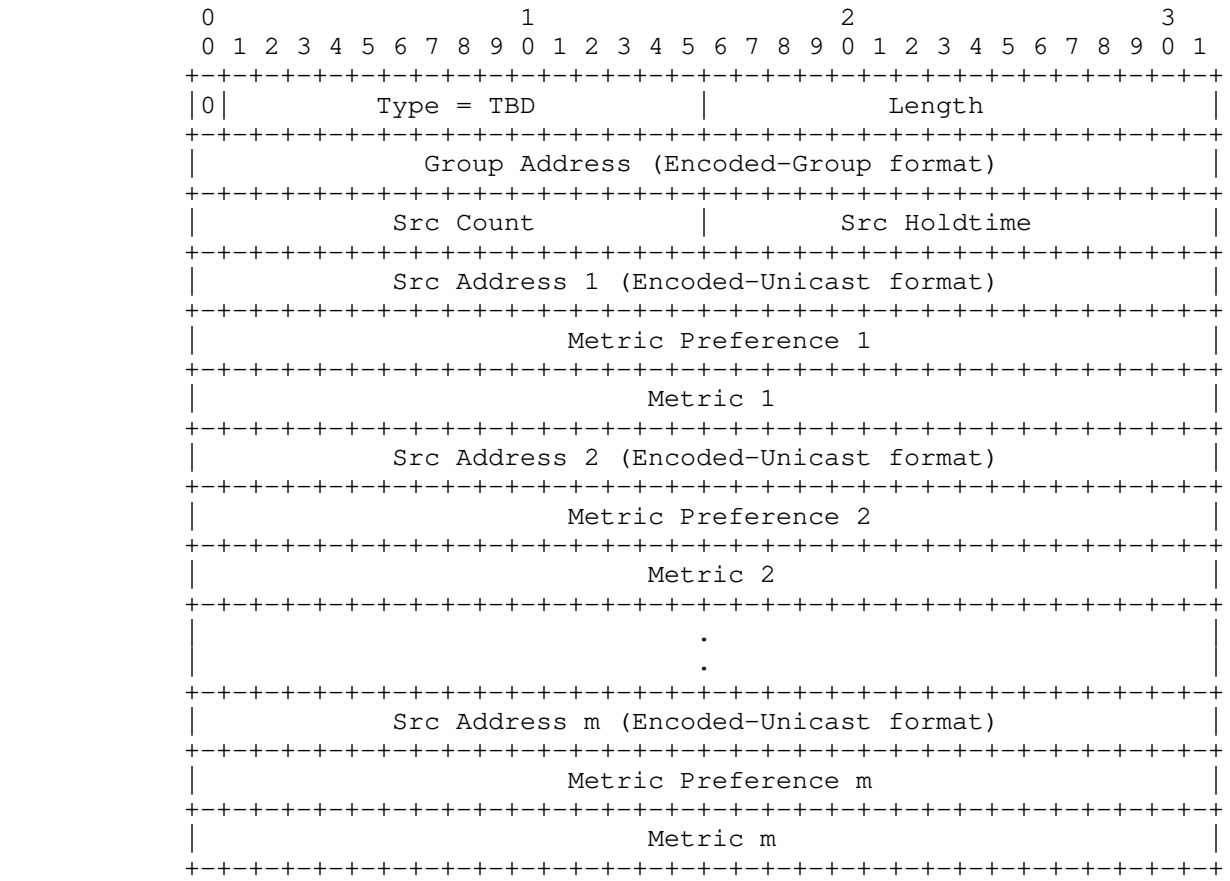
BFR-id: The BFR-id of the router that added this TLV in the sub-domain specified. The length is 2 octets.

BFR-prefix: The BFR-prefix of the router that added this TLV in the sub-domain specified. This length is 4 octets for IPv4 and 16 octets for IPv6.

4. Group Source Holdtime Metric TLV

When a router forwards a PFM message into a BIER domain, it should replace all Group Source Holdtime TLVs defined in [RFC8364] with the Group Source Holdtime Metric TLVs defined here. They are the same, except here we also add metric preference and metric. The metric preference and metric MUST be set to this router's metric and preference to reach the specified source. If the source is not reachable, the TLV MUST be omitted. This TLV is used together with the PFM Ingress BIER Router TLV is used to indicate the ingress router's cost of reaching the source.

When a router receives a message containing this TLV, it SHOULD store this information, but it MUST NOT forward these TLVs. If forwarding into another BIER domain, the metric preference and metric MUST be updated with this router's cost of reaching the source. If forwarding into a PIM domain, all the TLVs SHOULD be replaced with Group Source Holdtime TLVs as defined in [RFC8364]. The same information is used, except that the metric preference and metric are left out. One could potentially make use of the metric in a PIM domain as well, but it is not clear whether this is useful, and the PIM routers may not support this TLV.



0: The Transitive bit is set to 0.

Type: Type is TBD.

Length: The length of the value in octets.

Group Address: The group that sources are to be announced for. The format for this address is given in the Encoded-Group format in [RFC7761].

Src Count: The number of source addresses that are included.

Src Holdtime: The Holdtime (in seconds) for the included source(s).

Src Address: The source address for the corresponding group. The format for these addresses is given in the Encoded-Unicast address in [RFC7761].

Metric Preference: Preference value assigned to the unicast routing protocol that provided the route to the source.

Metric: The unicast routing table metric associated with the route used to reach the source. The metric is in units applicable to the unicast routing protocol used.

5. BIER signaling enhancements

A BIER border router SHOULD cache all the Group Source Holdtime Metric TLVs it receives, along with the respective PFM Ingress BIER Router TLV. This allows the router to determine which sources are active, and which BIER border router is closest to the source. The sub-domain ID, BFR-id and BFR-prefix in the TLV provide the necessary information for use by signaling mechanisms such as [I-D.ietf-bier-pim-signaling] to signal the preferred ingress router. It may also be used by [I-D.ietf-bier-mld]. IGMP/MLD reports would generally be sent to all BIER routers as it is not known which sources are active and which routers can reach them. But by using the enhancements in this document, a source-specific report can be sent to the router closest to the source. Also a group report might be set to the set of routers that are closest to the sources for that group. This reduces the amount of receiver state on the BIER routers, and also the amount of messages each routers needs to process.

6. Security Considerations

TBD

7. IANA Considerations

This document defines two new PFM TLVs that needs to be assigned from the "PIM Flooding Mechanism Message Types" registry.

8. References

8.1. 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>.

[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>.

[RFC8364] Wijnands, IJ., Venaas, S., Brig, M., and A. Jonasson, "PIM Flooding Mechanism (PFM) and Source Discovery (SD)", RFC 8364, DOI 10.17487/RFC8364, March 2018, <https://www.rfc-editor.org/info/rfc8364>.

8.2. Informative References

[I-D.ietf-bier-mld]
Pfister, P., Wijnands, I., Venaas, S., Wang, C., Zhang, Z., and M. Stenberg, "BIER Ingress Multicast Flow Overlay using Multicast Listener Discovery Protocols", draft-ietf-bier-mld-01 (work in progress), June 2018.

[I-D.ietf-bier-pim-signaling]
Bidgoli, H., Dolganow, A., Kotalwar, J., Xu, F., mishra, m., and Z. Zhang, "PIM Signaling Through BIER Core", draft-ietf-bier-pim-signaling-04 (work in progress), October 2018.

Authors' Addresses

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

Email: stig@cisco.com

IJsbrand Wijnands
Cisco Systems, Inc.
De kleetlaan 6a
Diegem 1831
Belgium

Email: ice@cisco.com

Mankamana Mishra
Cisco Systems, Inc.
Tasman Drive
San Jose CA 95134
USA

Email: mankamis@cisco.com

Mahesh Sivakumar
Juniper Networks
1133 Innovation Way
Sunnyvale CA 94089
USA

Email: sivakumar.mahesh@gmail.com

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: March 5, 2019

J. Xie
Huawei Technologies
L. Geng
L. Wang
China Mobile
G. Yan
M. McBride
Y. Xia
Huawei

September 1, 2018

Encapsulation for BIER in Non-MPLS IPv6 Networks
draft-xie-bier-6man-encapsulation-02

Abstract

Bit Index Explicit Replication (BIER) introduces a new multicast-specific BIER Header. Currently BIER has two types of encapsulation formats: one is MPLS encapsulation, the other is Ethernet encapsulation. This document proposes a BIER IPv6 encapsulation for Non-MPLS IPv6 Networks using an 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].

Status of This Memo

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

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

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

This Internet-Draft will expire on March 5, 2019.

Table of Contents

1. Introduction 2
2. Terminology 3
3. Problem Statement and Requirements 3
3.1. Problem Statement 3
3.2. Requirements 3
4. IPv6 BIER Encapsulation 4
4.1. Considerations 4
4.2. IPv6 BIER Destination Option 4
4.3. The whole IPv6 header for BIER packets 5
5. IPv6 BIER Forwarding 6
6. Security Considerations 7
7. IANA Considerations 7
8. Acknowledgements 8
9. Appendix A - BIER over IPv6 SRH Tunnel 8
10. References 9
10.1. Normative References 9
10.2. Informative References 10
Authors' Addresses 10

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 two types of BIER encapsulation to run on physical links: one is BIER MPLS encapsulation to run on various physical links that support MPLS, the other is BIER Ethernet encapsulation to run on ethernet links, with an ethertype 0xAB37. This document proposes a BIER IPv6 encapsulation for Non-MPLS IPv6 Networks using an IPv6 Destination Option extension header.

2. Terminology

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

3. Problem Statement and Requirements

3.1. Problem Statement

MPLS is a very popular and successful encapsulation. With MPLS encapsulation, packets forwarding can not only run on various physical links hop-by-hop, but also leverage the MPLS bypass tunnel to gain the "fast reroute" capability.

This same label benefit is also available for BIER by using an MPLS encapsulation. For example, an MPLS-encapsulated BIER packet can be forwarding on various physical links hop-by-hop, as well as on any MPLS bypass tunnels to support "fast reroute".

With a BIER Ethernet encapsulation, however, a packet can not be forwarded on any other type of links except for ethernet links in hop-by-hop case. It can not run on an MPLS bypass tunnel to support "fast reroute" either.

In an IPv6 network, there are considerations of using a non-MPLS encapsulation for unicast as the data-plane, such as SRH defined in [I-D.ietf-6man-segment-routing-header], where the function of a bypass tunnel uses an SRH header, with one or many Segments (or SIDs), instead of MPLS Labels. In such case, it is expected to have a BIER IPv6 encapsulation, which can run on IPv6 to be compliant with various kind of physical link in hop-by-hop case, as well as on SRH tunnel to have the significant benefit of "fast reroute" and so on.

3.2. Requirements

This chapter lists the BIER IPv6 encapsulation requirements needed to make the deployment of BIER on IPv6 network with SRH data-plane the same as on IPv4/IPv6 network with MPLS data-plane. These BIER IPv6 encapsulation requirements should provide similar benefits to MPLS encapsulation such as "fast reroute" or "run on any link or interface".

1. The listed requirements MUST be supported with any L1/L2 over which BIER layer can be realized.

2. It SHOULD support a hop-by-hop replication to multiple destinations in a BIER Domain.

3. It SHOULD support BIER on an "SRH tunnel".

4. It SHOULD align with the recommendations of the 6MAN working group.

4. IPv6 BIER Encapsulation

4.1. Considerations

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 diagrams in BIER IPv6 encapsulation. It is also required for a BIER IPv6 encapsulation to include the BIER Header ([RFC8296]) as an IPv6 Extension Header, to pilot the hop-by-hop BIER replication.

According to [RFC8200], [RFC6564], and [RFC7045], a new defined IPv6 extention header is not recommended, and an IPv6 Destination Option extension header is suitable and recommended for such a well-known BIER header as its Option.

4.2. IPv6 BIER Destination Option

The IPv6 BIER Destination Option is carried by the IPv6 Destination Option Header (indicated by a Next Header value 60). It is initialized in a packet sent by an IPv6 BFIR router to inform the following BFR routers in an IPv6 BIER domain to replicate to destination BFER routers hop-by-hop.

The IPv6 BIER Destination Option is encoded in type-length-value (TLV) format as follows:

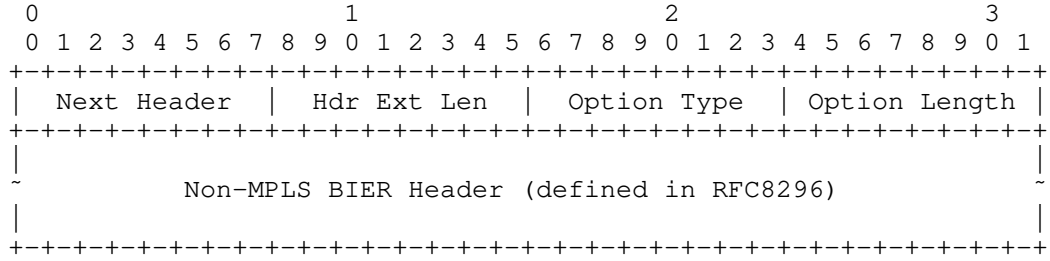


Figure 1: IPv6 BIER Destination Option

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 TBD. Need to be allocated by IANA.

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, including the BIFT-id. The function of TTL field is replaced by the Hop Limit field in IPv6 header and MUST be set to a non-zero value. The function of Entropy field is replaced by the Flow Label field in IPv6 header and MUST be set to zero value.

4.3. The whole IPv6 header for BIER packets

[RFC8200] specifies that the Destination Option Header can be located either before the Routing Header or after the Routing Header. However, this document requires that the Destination Option Header with a BIER Destination Option TLV is always located after the Routing Header if the Routing Header is present.

This is because the BIER header is always handled after the tunnels (or bypass tunnels) have been handled. BIER MPLS encapsulation has the same behavior. To quote [RFC8296]:

- It is crucial to understand that in an MPLS network the first four octets of the BIER encapsulation header are also the last four octets of the MPLS header. Therefore, any prior MPLS label stack entries MUST have the S bit (see [RFC3032]) clear (i.e., the S bit must be 0).

Other IPv6 extension headers are not commonly used in the current Internet. For Example, [RFC6744] says that "IPv6 Destination Options headers, and the options carried by such headers, are extremely uncommon in the deployed Internet". [RFC6564] says that "Extension headers, with the exception of the Hop-by-Hop Options header, are not usually processed on intermediate nodes", and that "Reports from the field indicate that some IP routers deployed within the global Internet are configured either to ignore the presence of headers with hop-by-hop behavior or to drop packets containing headers with hop-by-hop behavior."

Such IPv6 extension headers will even be more uncommon when a BIER encapsulation is used in data-plane forwarding. The entire IPv6 header, with BIER encapsulation and Routing Header, is expected to look like this:

IPv6 header [Multicast Address in DA]

Hop-by-Hop Options header [No use]

Destination Options header [No use]

Routing header [SRH Header may be used, Appendix A]

Fragment header [No use]

Authentication header [No use]

Encapsulating Security Payload header [No use]

Destination Options header [BIER header in BIER Option TLV]

Upper-layer header [BIER payload]

In a hop-by-hop BIER IPv6 replication scenario, there is only an IPv6 header with DA being a "BIER specific" Multicast address, and an IPv6 Destination Option header with a BIER destination option TLV.

BIER header has a 'proto' field to identify the type of BIER packet payload, and the IANA has created a registry called "BIER Next Protocol Identifiers" to assign the value. That means the 'Upper-layer header' of a BIER packet have already been identified by the 'proto' field of the BIER header in the Destination Option Header. Thus the 'Next Header' in the Destination Option Header is not need to identify the 'Upper-layer header' any more, and is recommended to be set to 'No Next Header (value 59)'.

Procedures for encapsulating a BIER IPv6 packet in SRH tunnel are outside the scope of this document.

Procedures for encapsulating a BIER IPv6 packet in other types of tunnels are outside the scope of this document.

5. IPv6 BIER Forwarding

In an IPv6 BIER domain, the Multicast Address of the IPv6 DA in an incoming BIER IPv6 packet indicates the BIER information of this 'host', and the packet will be forwarded according to the BIER Header in the BIER Destination Option TLV in the IPv6 Destination Option extension header. A router is required to ignore the IPv6 BIER Destination Option if the IPv6 Destination Address of a packet is not a multicast address, or is a multicast adddress without indicating the BIER information of this 'host'.

Below is the procedure that a BFR uses for forwarding a BIER IPv6 encapsulated packet.

1. Read the IPv6 header, get the the IPv6 DA, and get the indication of the multicast address if the IPv6 DA is a multicast address. The case when IPv6 DA not being a multicast address is outside the scope of this document.

2. If the multicast address is interested by this router, and the 'Next Header' of the IPv6 header indicates a IPv6 Destination Option Header, then read the IPv6 Destination Option Header, and get the BIER Option (BIER Header). The case when the multicast address not being interested by this router is outside the scope of this document.

3. The following steps are the same as step 1 to 9 described in chapter 6.5 in [RFC8279]. One difference need to point out is that, the copied packet includes a IPv6 header, a IPv6 Destination Header and its BIER Destination Option Type and Option Length before the BIER Header. If the copied packet is forwarded to a BFR-NBR, the 'Hop Limit' field of the IPv6 header MUST be decremented, whereas the TTL in the BIER header MUST be unchanged.

Procedures for forwarding a BIER IPv6 packet in SRH tunnel, and hand-off to a hop-by-hop replication, can refer to Appendix A.

Procedures for forwarding a BIER IPv6 packet in other types of tunnels, and hand-off to a hop-by-hop replication, are outside the scope of this document.

6. Security Considerations

An IPv6 BIER Destination Option with Multicast Address Destination would be used only when an IPv6 BIER state with the specific Multicast Address Destination has been built by the control-plane. Otherwise the packet with an IPv6 BIER Destination Option will be discarded.

7. IANA Considerations

Allocation is expected from IANA for a BIER Destination Option Type codepoint from the "Destination Options and Hop-by-Hop Options" sub-registry of the "Internet Protocol Version 6 (IPv6) Parameters" registry [RFC2780] at <https://www.iana.org/assignments/ipv6-parameters/>.

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 at <https://www.iana.org/assignments/ipv6-multicast-addresses/>.

8. Acknowledgements

TBD.

9. Appendix A - BIER over IPv6 SRH Tunnel

In a Non-MPLS IPv6 Network, BIER may be deployed in a hop-by-hop manner, or possibly be deployed through an SRH tunnel either for "bypassing Non-capable BIER routers" or "fast rerouting". Here is an example where a packet is firstly forwarded through an SRH tunnel and then through a hop-by-hop BIER domain.

When a router along the Segment Routing path receives an IPv6 BIER packet with an SRH header, and if the IPv6 destination address is not one of the router's address, then the packet is forwarded by an IPv6 FIB lookup of the destination address and none of the IPv6 extension headers will be checked. If the IPv6 Destination Address is one of the router's address, and also one of the router's Segment (or SID) of some type, then the router will do a specific function indicated by the Segment, as defined in [I-D.filsfils-spring-srv6-network-programming]. If the IPv6 Destination Address is a specific type of Segment, called BIER Segment or BIER SID, then the according function is called Endpoint BIER function or 'End.BF' function for short.

When router receives a packet destined to X and X is a local End.BF SID, the router does:

```
1.    IF SL > 0
2.      decrement SL
3.      update IPv6 DA with SRH[SL]
4.      IF SL = 0 & STATE(SRH[0]) = BIER
5.        update IPv6 header NH with SRH NH
6.        pop the SRH
7.        forward the updated packet
8.      ELSE
9.        drop the packet
10.   ELSE
11.     drop the packet
```

Figure 2: End.BF Function

The End.BF function is used for the SRH tunnel destination router to terminate the source-routing SRH forwarding and begin the hop-by-hop BIER IPv6 forwarding. After the SRH header is popped, the multicast address in the updated IPv6 Destination Address indicates the BIER information of this 'host', and the packet will be forwarded according to the BIER Header in the BIER Destination Option TLV in the IPv6 Destination Option extension header of this 'host'.

In the following hop-by-hop forwarding procedure, the IPv6 Destination Address in an incoming packet indicates the BIER information of this 'host', and the packet will be forwarded according to the BIER Header in the BIER Destination Option TLV in the IPv6 Destination Option extension header. A router is required to ignore the IPv6 BIER Destination Option if the IPv6 Destination Address of a packet is not a multicast address, or is a multicast adddress without indicating the BIER information of this 'host'.

10. References

10.1. Normative References

[I-D.filsfils-spring-srv6-network-programming]
Filsfils, C., Camarillo, P., Leddy, J., daniel.voyer@bell.ca, d., Matsushima, S., and Z. Li, "SRv6 Network Programming", draft-filsfils-spring-srv6-network-programming-05 (work in progress), July 2018.

[I-D.ietf-6man-segment-routing-header]
Filsfils, C., Previdi, S., Leddy, J., Matsushima, S., and d. daniel.voyer@bell.ca, "IPv6 Segment Routing Header (SRH)", draft-ietf-6man-segment-routing-header-14 (work in progress), June 2018.

[RFC6564] Krishnan, S., Woodyatt, J., Kline, E., Hoagland, J., and M. Bhatia, "A Uniform Format for IPv6 Extension Headers", RFC 6564, DOI 10.17487/RFC6564, April 2012, <https://www.rfc-editor.org/info/rfc6564>.

[RFC7045] Carpenter, B. and S. Jiang, "Transmission and Processing of IPv6 Extension Headers", RFC 7045, DOI 10.17487/RFC7045, December 2013, <https://www.rfc-editor.org/info/rfc7045>.

[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>.

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>.

Authors' Addresses

Jingrong Xie
Huawei Technologies

Email: xiejingrong@huawei.com

Liang Geng
China Mobile
Beijing 10053

Email: gengliang@chinamobile.com

Lei Wang
China Mobile
Beijing 10053

Email: wangleiyjy@chinamobile.com

Gang Yan
Huawei

Email: yangang@huawei.com

Mike McBride
Huawei

Email: mmcbride7@gmail.com

Yang Xia
Huawei

Email: yolanda.xia@huawei.com

BIER WG
Internet-Draft
Intended status: Standards Track
Expires: April 15, 2019

Quan Xiong
Fangwei Hu
Greg Mirsky
ZTE Corporation

October 12, 2018

The Resilience for BIER
draft-xiong-bier-resilience-01.txt

Abstract

Bit Index Explicit Replication (BIER) is an architecture that specifies a solution for the forwarding of multicast data packets. In some scenarios, the resilience should be provided to guarantee the multicast data is protected by a given backup resource and forwarded successfully to the receivers in BIER-specific network.

This document discusses the resilience use cases, requirements and proposes solutions for BIER, including the protection mechanisms and detection methods.

Table of Contents

1. Introduction 2
1.1. Requirements Language 3
1.2. Terminology 3
2. Requirements 3
3. BIER Resilience Use Cases 3
3.1. End-to-End 1+1 Protection 3
3.2. End-to-End 1:1 Protection 4
3.3. BIER Link Protection 5
4. Security Considerations 6
5. IANA Considerations 6
6. Acknowledgements 6
7. References 6
7.1. Normative References 6
7.2. Informational References 7
Authors' Addresses 7

1. Introduction

[RFC8279] introduces Bit Index Explicit Replication (BIER) architecture and specifies a solution for the forwarding of multicast data packets. The routers which support BIER are known as Bit-Forwarding Router (BFR) and the multicast data packet enters a BIER domain at a Bit-Forwarding Ingress Router (BFIR) and leave at one or more Bit-Forwarding Egress Routers (BFERs).

[I-D.eckert-bier-te-frr] provides some protection mechanisms for traffic engineering of BIER. However, there is no mechanism to protect multicast traffic against BIER-specific network failures. In some scenarios, the resilience should be provided to guarantee the multicast data is protected by a given backup resource and forwarded successfully to the receivers in BIER-specific network.

This document describes the resilience use cases and requirements for BIER-specific network and discusses the protection mechanisms and detection methods.

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 [RFC2119].

1.2. Terminology

The terminology is defined as [RFC8279].

2. Requirements

The following lists the resilience requirements for BIER-specific multicast domain including the protection mechanisms and detection methods.

(1) The listed requirements MUST be supported by any transport layer over which the BIER layer can be realized.

(2) BIER protection type MAY be defined and configured from a centralized controller or management network including BIER end-to-end protection and link/node protection and related information.

(3) It is required to support the failure detection and notification mechanisms.

(4) It is required to support the fast protection switching for the BIER packets within the limited time.

3. BIER Resilience Use Cases

The resilience use cases for a BIER-specific network should be considered including end-to-end and link protection scenarios. The protection and related detection mechanisms MAY be provided for BIER resilience against failures such as link or nodes.

3.1. End-to-End 1+1 Protection

The end-to-end protection mechanisms for a BIER-specific network should be considered in some scenarios like shown in Figure 1. It includes end-to-end 1+1 and 1:1 protection use cases. Two disjoint end-to-end paths that are available for 1+1 or 1:1 protection from BFIR to BFERs should be provided, and one of them may be configured to be the protection path for the working path. In this example the working path could be BFIR->BFR1->BFR2->BFR3->BFER1 and BFIR->BFR1->BFR2->BFR3->BFER2; and then the protection path is BFIR->BFR6->BFR5->BFR4->BFER1 and BFIR->BFR6->BFR5->BFR4->BFER2.

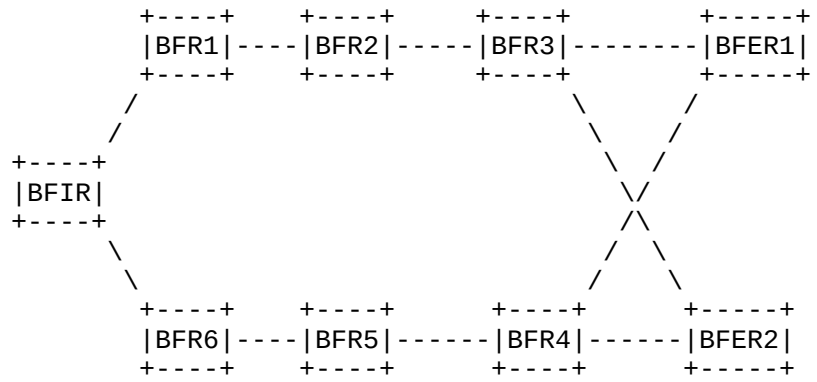


Figure 1: BIER End-to-End Protection

For 1+1 protection scenario, the multicast traffic MUST be sent across the network through both the working and backup paths. When the link or node failure occurs inf the working path, the BFERs need to switch to receiving the data flow from the protection path.

The failure detection mechanism for end-to-end 1+1 protection scenario MUST be able to monitor and detect multicast failures in working and protection paths. P2MP BFD [I-D.ietf-bfd-multipoint] MAY be used to verify multipoint connectivity between a BFIR and a set of BFERs. [I-D.hu-bier-bfd] describes the use of p2mp BFD in a BIER domain.

End-to-End 1+1 protection provides fast switch but low resource utilization. All BFERs MAY receive two copies from two paths in the no-failure scenario, and the receivers MUST be able to choose one of them and eliminate the duplication.

3.2. End-to-End 1:1 Protection

This section discusses the end-to-end 1:1 protection for BIER. If duplicate transmission is not desirable for some networks, end-to-end 1:1 protection mechanism may be taken into consideration where only one copy is sent to each receiver. The BFIR will send multicast flows onto the working path and switch to the backup path when a failure occurs.

The failure detection mechanism for end-to-end 1:1 protection scenario MUST be able to monitor and detect multicast failures in the receivers (tails) and notify the head node. BIER-specific extensions MAY be proposed based on [I-D.ietf-bfd-multipoint-active-tail]. The P2MP active tail detection method extends the mechanism defined in

[I-D.ietf-bfd-multipoint]. It allows tails to notify the head of the failure of the multicast path and can be used in multipoint and multicast networks, e.g., in BIER domain.

If P2MP BFD uses the active tail mode, then when one of the BFERs detects the failure of the working path, it will send a message to the BFIR. The BFIR will notify BFERs of switchover and start forwarding the multicast flows over the protection path.

3.3. BIER Link Protection

Local protection, i.e., link or node protection, MAY be considered for BIER domain as an alternative to end-to-end protection. The nodes which are the BFRs in BIER network and they exchange the information needed for them to forward packets to each other using BIER. The node protection MAY be provided by using mechanisms already existing in the underlay network, for example, described in [I-D.eckert-bier-te-frr].

A BFR MAY send BIER packets to directly connected BIER neighbors through a BIER link without requiring a routing underlay. Link protection SHOULD be considered in BIER domain. The detection of link failure MAY use the Point-to-Point BFD detection defined in [RFC5880]. A set of extension for BIER-specific P2P BFD SHOULD be proposed in further discussion.

As shown in Figure 2, the BIER path from BFIR to BFERs is BFIR->BFR4->BFR3 ->BFR2->BFER1 and BFIR->BFR4->BFR3->BFER2. If the BIER link from BFR4 to BFR3 fails, the failure can be protected by the backup paths over BFR4->BFR1->BFR2 ->BFR3.

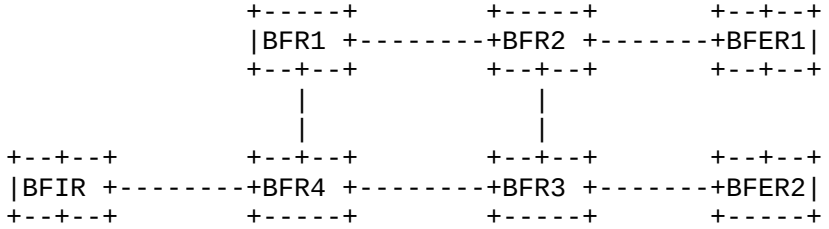


Figure 2: BIER Link Protection

As discussed in [I-D.eckert-bier-te-frr], the BIER link protection MAY use the existing RSVP-TE/P2MP or SR tunnel bypass. When a node detects a failure on a link, it MAY be assumed that the link has

failed and the traffic is switched onto the pre-established backup path to get packets to the downstream node.

Also, as discussed in [RFC7490], the Topology Independent Loop-free Alternate Fast Re-route (TI-LFA) Fast Reroute (FRR) approach that achieves guaranteed coverage against link or node failure in the Interior Gateway Protocol (IGP) network MAY be applied in BIER network.

4. Security Considerations

Security aspects of protection in BIER domain may be considered in relation to the data plane, and handling the dedicated OAM packets used to detect, signal a failure, coordinate the state in the BIER protection domain.

5. IANA Considerations

TBD

6. Acknowledgements

TBD

7. References

7.1. Normative References

[I-D.hu-bier-bfd]
hu, f., Mirsky, G., Xiong, Q., and C. Liu, "BIER BFD", draft-hu-bier-bfd-02 (work in progress), October 2018.

[I-D.ietf-bfd-multipoint]
Katz, D., Ward, D., Networks, J., and G. Mirsky, "BFD for Multipoint Networks", draft-ietf-bfd-multipoint-18 (work in progress), June 2018.

[I-D.ietf-bfd-multipoint-active-tail]
Katz, D., Ward, D., Networks, J., and G. Mirsky, "BFD Multipoint Active Tails.", draft-ietf-bfd-multipoint-active-tail-09 (work in progress), June 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>.

[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>.

[RFC7490] Bryant, S., Filsfils, C., Previdi, S., Shand, M., and N. So, "Remote Loop-Free Alternate (LFA) Fast Reroute (FRR)", RFC 7490, DOI 10.17487/RFC7490, April 2015, <https://www.rfc-editor.org/info/rfc7490>.

[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. Informational References

[I-D.eckert-bier-te-frr] Eckert, T., Cauchie, G., Braun, W., and M. Menth, "Protection Methods for BIER-TE", draft-eckert-bier-te-frr-03 (work in progress), March 2018.

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

Fangwei Hu
ZTE Corporation
No.889 Bibo Rd
Shanghai 201203
China

Phone: +86 21 68896273
Email: hu.fangwei@zte.com.cn

Greg Mirsky
ZTE Corporation
USA

Email: gregimirsky@gmail.com

BIER
Internet-Draft
Intended status: Standards Track
Expires: March 16, 2019

Zheng. Zhang
Bo. Wu
ZTE Corporation
Zhaohui. Zhang
Juniper Networks
IJsbrand. Wijnands
Cisco Systems, Inc.
September 12, 2018

BIER Prefix Redistribute
draft-zwzw-bier-prefix-redistribute-01

Abstract

This document defines a BIER proxy function to interconnect different underlay routing protocol areas in a hybrid network. And a new BIER proxy range sub-TLV is also defined to convey BIER BFR-id information across the routing areas.

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 March 16, 2019.

Table of Contents

1. Problem statement 2
2. Proposal 4
3. Advertisement 5
3.1. BIER proxy range sub-TLV 5
4. Example 6
5. IANA Considerations 7
6. Security Considerations 7
7. Normative References 7
Authors' Addresses 8

1. Problem statement

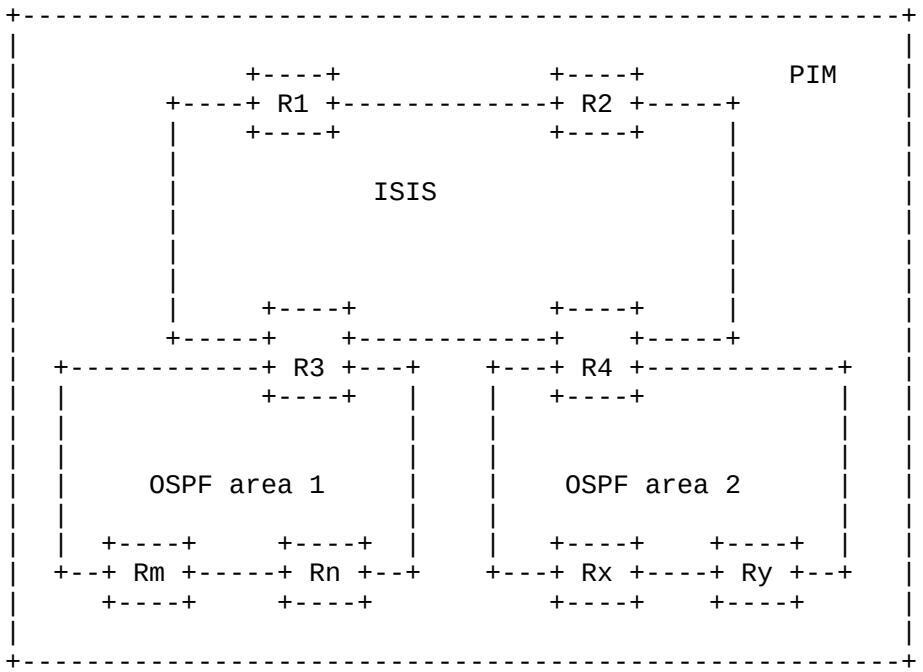


Figure 1

Figure 1 shows a hybrid network with different IGP routing protocols deployed. Hybrid network is used mostly for management consideration. There are just small number of routers in each area of the network. Currently, multicast services are provided in this hybrid network by using protocol independent feature of PIM.

BIER could be a candidate multicast protocol to replace PIM to reduce multicast states in the hybrid network. BIER [RFC8279] is a new architecture for the forwarding of multicast data packets. It does not require a protocol for explicitly building multicast distribution trees, nor does it require intermediate nodes to maintain any per-flow state. In order to build BIER forwarding plane, BIER key parameters must be flooded in one BIER domain such as BFR-prefix, BFR-id, subdomain-id, and so on. The routing protocols which are used to flood these BIER parameters are called BIER routing underlay. The associated routing protocol extensions are defined in documents such as [RFC8401], [I-D.ietf-bier-ospf-bier-extensions], [I-D.ietf-bier-idr-extensions], and so on.

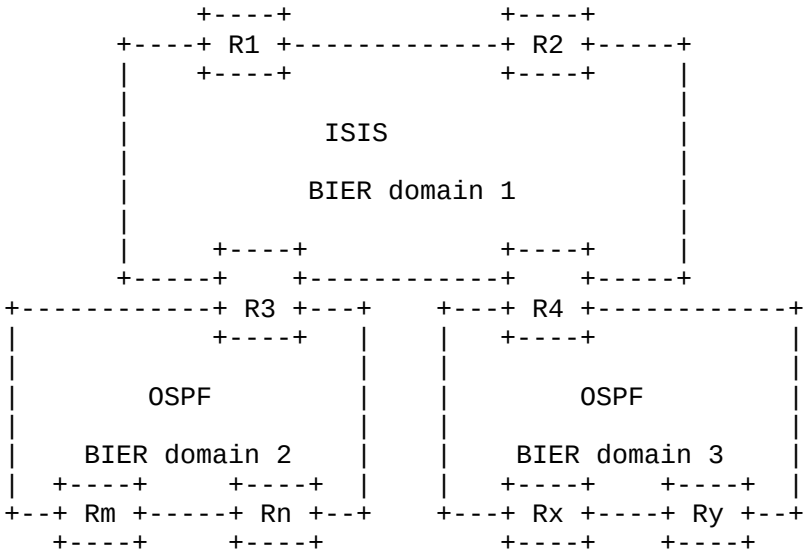


Figure 2

Based on the BIER design, a BIER domain is limited by the underlay routing protocols flooding scope. As in a hybrid network depicted in figure 2, in case we want deploy BIER instead of PIM, there are several BIER domains because of different underlay routing protocols limitation. Multiple encapsulating/ decapsulation executions are needed to across multiple BIER domains. These executions slow down the forwarding efficiency. The border routers also need to maintain overlay state, which is undesired.

Except the hybrid network, there is the situation that several areas formed by one same IGP protocol need to be merged into one BIER domain in existing network. The prefix redistribution method defined in this document can be used too.

2. Proposal

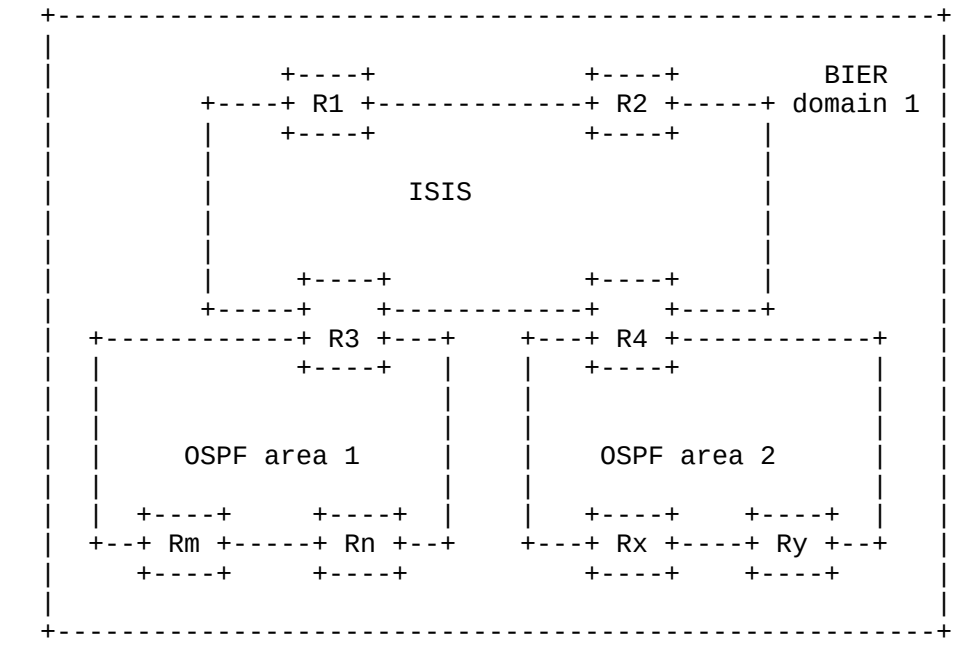


Figure 3

It is more efficient to deploy BIER by creating one BIER domain for the hybrid network to achieve forwarding benefit.

Since the limitation of the BIER routing protocol scope, BFR-id is confined to only one routing area. A BIER proxy function is introduced to transport BIER BFR-id information in a BIER domain across multiple routing protocol areas. So BIER forwarding tables can be built across multiple underlay routing protocols to replace encapsulation/decapsulation processing. In the current deployment, border router (ABR) has a similar role, ABR summaries unicast routing information from one routing protocol area and sends it to another routing area by new routing protocol messages. So ABR can implement BIER proxy function to summarize BIER BFR-id information from one routing protocol area and sends it to another routing area.

In figure 3, R3 and R4 connect two areas which running different routing protocols, they can be used as BIER proxies to transport BIER

information. For example, after R3 receives BFR-ids information from OSPF area 1 and sends it to ISIS routing area, the routers in ISIS routing area can generate BIER forwarding items toward the BFR-ids in OSPF area 1. Similarly, R3 receives BFR-ids information from ISIS area and sends it to OSPF area 1, the routers in OSPF area 1 can build BIER forwarding items toward the BFR-ids in ISIS area. R4 does the same function, the BIER forwarding plane is constructed accordingly.

3. Advertisement

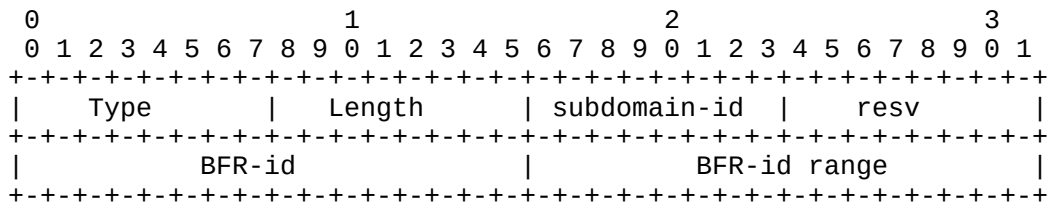
According to [RFC8279], each BFER needs to have a unique (in each sub-domain) BFR-id, and each BFR and BFER floods itself BIER info sub-TLV and associated sub-sub-TLVs in the BIER domain. To keep consistent with the definition in [I-D.ietf-bier-ospf-bier-extensions] and [RFC8401], BIER info sub-TLV defined in [RFC8401] and BIER sub-TLV defined in [I-D.ietf-bier-ospf-bier-extensions] is reused to convey the BFR-id information. OSPF extended Prefix Opaque LSA [RFC7684] and TLVs 235, 237 defined in [RFC5120] are still used to carry the BFR-id / BFR-prefix information.

The key parameters got from the original routing protocol should be adapted to the format of next routing protocol, such as BFR-prefix, BFR-id, subdomain-id, and so on. Some parameters like BAR, MT-ID has local significance, So they should be set to same values with BIER proxy own advertisement when BIER proxy advertise them to the next routing area.

And as the two BIER info sub-sub-TLVs (sub-TLVs) including MPLS encapsulation and BSL conversion also have local significance. The information carried in these two sub-sub-TLV need not, but MAY, be advertised to next routing area.

3.1. BIER proxy range sub-TLV

In case unicast default route and aggregated / summarized routes are used in some routing areas and routers in next area can not see the specific BFR-prefix routes from original area, the prefix advertised should be set to default route or aggregated / summarized routes. Like in figure 3, in case R3/R4 does not advertise specific ISIS unicast routes to OSPF area and only advertises unicast default route or aggregated / summarized route to OSPF area 1/2, when R3/R4 advertises BIER info sub-TLV to OSPF area 1/2, R3 MUST advertise the prefix with default route or aggregated / summarized route. In that case, multiple BFR-ids will be mapped to one prefix. In order to advertise BFR-ids optimally, we define a new BIER proxy range sub-TLV to advertise the information of BFR-ids.



- Type: TBD to indicate the BIER proxy range sub-TLV.
- Length: variable.
- Subdomain-id: The subdomain-id from original advertisement.
- resv: The reserved field.
- BFR-id: The first BFR-id from original advertisement.
- BFR-id range: The range of BFR-ids with one subdomain-id.

The BIER proxy range sub-TLV is attached to the aggregated / summarized route prefix or default route prefix. The summarized / aggregated / default prefix may need multiple BIER proxy range sub-TLVs if the BFR-ids covered by the prefix are allocated from different ranges (even if they’re from a single range but if some BFR-ids in the range map to some BIER prefixes that are covered by a different summarized / aggregated prefix, then that single large range needs to be broken into smaller ranges).

The BFR-ids associated with the summarized prefix can be advertised individually in the BIER range sub-TLV. Though BFR-id’s range can increase advertisement efficiency, necessary configuration / policy should be provided to guide the range generation of BFR-ids. Otherwise unwanted amount of updates may occur when a BFR-id is removed from the range.

Because a summarized / default prefix covers many BIER prefixes, the mapping between a BIER prefix and its BFR-id is no longer conveyed in the routing underlay. As a result, the mapping must be provided by other means, e.g. in the multicast overlay.

4. Example

As in figure 3, R3 and R4 as BIER proxy, R3 as an example should advertise the BIER BFR-ids information from ISIS area to OSPF area 1 with the advertiser set to R3 itself, and advertise BIER info from OSPF area 1 to ISIS area as well. In case R3 and R4 generates specific BFR-prefix and BFR-ids from the original area to the next

area, BIER info sub-TLV defined in [RFC8401] and BIER sub-TLV defined in [I-D.ietf-bier-ospf-bier-extensions] is reused to convey the BFR-id information. All the routers generate BIER forwarding items to other area toward BIER proxy according to [RFC8279].

In case BIER proxy can not advertise specific BFR-prefix but aggregated / summarized / default prefix from the original area to the next area, BIER proxy range sub-TLV is used to convey the information. Suppose that Rm is an ingress router, R1, R2, Rx and Ry is egress router, the BFR-ids of these egress router are 31, 55, 112, 157. The BFR prefixes of them are 10.1.1.5, 10.1.1.50, 203.1.1.10, 203.1.1.60. Suppose that summarized prefixes are advertised into OSPF area. The summarized prefixes are 10.1.1.0/24 and 203.1.1.0/24. All the routers in OSPF area 1 compute forwarding table for unicast / BIER according to the summarized prefixes, and they can get to these prefixes by routes toward proxy R3.

Rm encapsulate multicast flow with BIER header that with 31, 55, 112 and 157 bit set in the BIER header (Supposed that 256 BitStringLength is used). The routers in OSPF area 1 forward packet toward R3. R3 forwards packet according to the BFR-ids set in the BIER header normally. Later packet reaches R1, R2 and R4. Similarly, R4 forwards packet into OSPF area 2 normally. Finally packet reaches Rx and Ry.

5. IANA Considerations

IANA is requested to set up a new types of sub-TLV (TLV) registry value for BIER proxy range advertisement in OSPF, ISIS, BGP, etc.

6. Security Considerations

Implementations must assure that malformed TLV and Sub-TLV permutations do not result in errors which cause hard protocol failures.

7. 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-05 (work in progress), March 2018.

[I-D.ietf-bier-ospf-bier-extensions]
Psenak, P., Kumar, N., Wijnands, I., Dolganow, A., Przygienda, T., Zhang, Z., and S. Aldrin, "OSPFv2 Extensions for BIER", draft-ietf-bier-ospf-bier-extensions-18 (work in progress), June 2018.

[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>.

[RFC7684] Psenak, P., Gredler, H., Shakir, R., Henderickx, W., Tantsura, J., and A. Lindem, "OSPFv2 Prefix/Link Attribute Advertisement", RFC 7684, DOI 10.17487/RFC7684, November 2015, <https://www.rfc-editor.org/info/rfc7684>.

[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>.

Authors' Addresses

Zheng(Sandy) Zhang
ZTE Corporation

EMail: zzhang_ietf@hotmail.com

Bo Wu
ZTE Corporation

EMail: w1973941761@163.com

Zhaohui Zhang
Juniper Networks

EMail: zzhang@juniper.net

IJsbrand Wijnands
Cisco Systems, Inc.

EMail: ice@cisco.com

BIER
Internet-Draft
Intended status: Standards Track
Expires: November 25, 2020

Z. Zhang
Juniper Networks
E. Rosen
Individual
D. Awduche
Verizon
L. Geng
China Mobile
May 24, 2020

Multicast/BIER As A Service
draft-zzhang-bier-multicast-as-a-service-01

Abstract

This document describes a framework for providing multicast as a service via Bit Index Explicit Replication (BIER) [RFC7279], and specifies a few enhancements to [draft-ietf-bier-idr-extensions] [RFC8279] [draft-ietf-bier-ospf-bier-extensions] to enable multicast/BIER as a service.

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 November 25, 2020.

Table of Contents

1. Introduction 2
1.1. Terminologies 3
1.2. A CDN of A Single Provider 4
1.2.1. IGP/BGP Interworking 5
1.3. A CDN That Involves Another Providers 6
1.3.1. Providing Independent BAAS To Multiple Customers . . 6
1.3.2. Control and Accounting 7
1.4. Sets and Segmentation 8
1.4.1. Multiple Sets 8
1.4.2. Segmentation 8
2. Specifications for Enhancements to BIER Signaling with BGP/IGP 9
2.1. BGP Procedures 9
2.2. ISIS/OSPF Procedures 10
3. IANA Considerations 10
4. Security Considerations 11
5. Acknowledgements 11
6. References 11
6.1. Normative References 11
6.2. Informative References 12
Authors' Addresses 12

1. Introduction

Currently multicast is primarily used in the following scenarios:

- Enterprise Applications. For example, large scale financial data publishing.

- Provider/underlay tunnels for MVPN and for EVPN BUM.

- Real-time IPTV offered by a service provider to its customers.

Besides the above, large scale multicast services, especially transit multicast transport provided by large Internet Service Providers is virtually non-existent. This is mainly because of the following chicken and egg dilemma:

- Traditional multicast technologies are complicated and lack scalability. The revenue that multicast services bring in cannot offset the Capex and Opex that an operator has to invest, so provider networks typically do not enable multicast even though the deployed equipment does support multicast.

- As a result, Content Providers cannot take advantage of multicast and instead use less efficient methods like Ingress Replication, Peer2Peer, or multicast at application layer.

A recent multicast technology breakthrough, BIER, provides a simple and scalable solution for large scale multicast deployment, independent of number of multicast flows. In the meantime, large scale distribution of ultra high definition video content has become more and more popular and important. Service providers simply cannot keep on increasing their network capacity even if they could shift cost to Content Providers. With these developments, service providers now have both the need and means to provide scalable multicast service, potentially across multiple providers.

This document describes a framework for Multicast As A Service (MAAS) enabled by BIER. We use Content Delivery Network (CDN) as example, though it applies to any large scale multicast delivery service.

1.1. Terminologies

Readers are assumed to be familiar with multicast, BIER, BGP and ISIS/OSPF concepts and procedures. Some terminologies are listed here for convenience.

- BFR: BIER Forwarding Router.

- BFIR: BIER Forwarding Ingress Router.

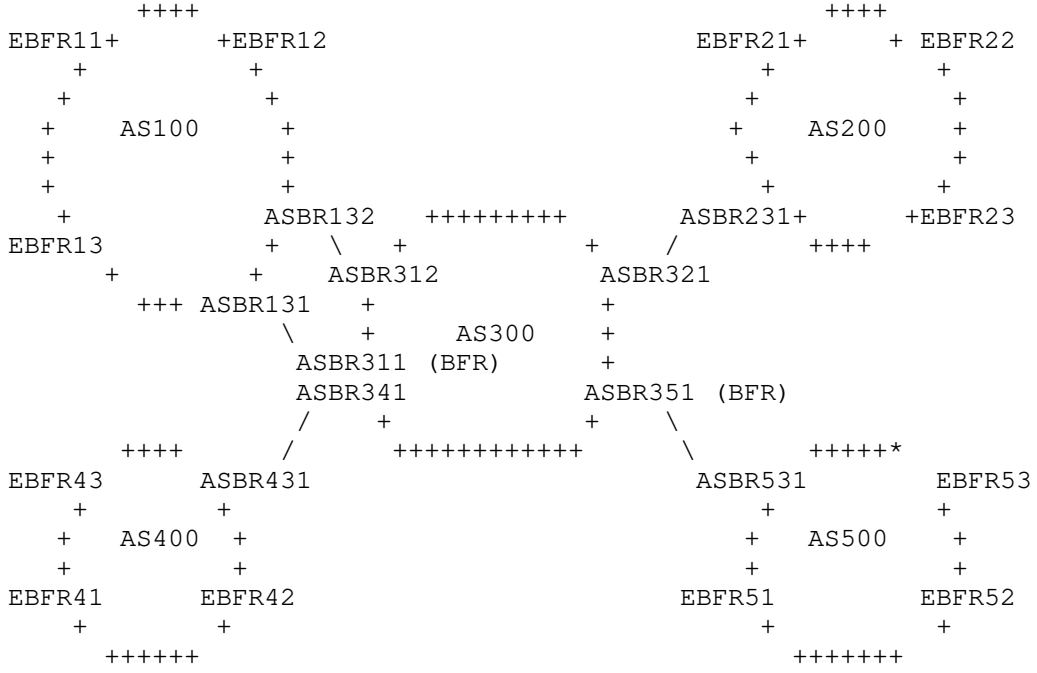
- BFER: BIER Forwarding Egress Router.

- EBFR: Edge BFR. Including BFIR and BFER.

- BSL: BitStrengLength. Number of bits in the BitString of a BIER header.

1.2. A CDN of A Single Provider

To make it easier to understand, we first consider a simple example: a CDN owned by a single operator, which could be a Content Provider itself. The network spans multiple ASes as shown in the following figure:



The CDN uses BIER for multicast transport and Edge BIER Forwarding Routers (EBFRs) are located throughout the network. Some of them are connected towards multicast content sources and are referred to as BIER Forwarding Ingress Routers (BFIRs) in BIER architecture. Most of them are connected towards multicast content receivers and are referred to as BIER Forwarding Egress Routers (BFERs). Notice that between content sources and BFIRs there may be Protocol Independent Multicast (PIM) in use, while between content receivers and BFERs there may be PIM and/or IGMP in use.

At the initial deployment stage, there might be only a few transit BIER Forwarding Routers (BFRs) at strategic points in the network (e.g. ASBR311 and ASBR351). BGP sessions are established among the EBFRs and BFRs, and BGP extensions as defined in

[I-D.ietf-bier-idr-extensions] are used to signal BIER information. All these are in a single BIER sub-domain.

In the example of initial stage with only ASBR311 and ASBR351 as BFRs, multicast traffic arriving at EBFR11 will be imposed with a BIER header and replicated to EBFR12/EBFR13/ASBR311 over tunnels. ASBR311 will further replicate traffic to ASBR351/EBFR41/EBFR42/EBFR43/EBFR21/EBFR22/EBFR23 over tunnels, and ASBR351 will further replicate traffic to EBFR51/EBFR52/EBFR53 over tunnels.

The BGP signaling and a necessary enhancement can be explained using the following example. EBFR43 advertises its BIER prefix (a loopback address) as /32 IPv4 or /128 IPv6 prefix in BGP with a BIER Path Attribute (BPA) [RFC8279] [I-D.ietf-bier-idr-extensions]. ASBR431 receives it and re-advertises it (with BGP Next Hop changed to itself) but does not do anything wrt BIER because it does not support BIER. Same happens on ASBR341. When ASBR311 and ASBR351 receive it from ASBR341, they create a BIFT entry corresponding to EBFR43's BFR-ID. The entry causes a BIER packet with corresponding bit set in its BitString to be tunneled to EBFR43. This cannot be based on BGP Next Hop in the advertisement because the BGP Next Hop is ASBR341. When eventually EBFR11 receives the re-advertised route, it creates a BIFT entry that causes corresponding packets to be tunneled to ASBR311 (but not to EBFR43 directly). Now it is clear that this cannot be based on either the BIER prefix itself or the BGP Next Hop. The solution is that the originating EBFR attaches a Tunnel Encap Attribute [I-D.ietf-idr-tunnel-encaps] with the tunnel destination set to itself, and whenever a BFR re-advertises the route it changes the tunnel destination to itself. When a BFR creates the BIFT entry, it uses the tunnels destination in the Tunnel Encapsulation Attribute to find out where to tunnel packets.

Over time, more routers in network may be upgraded to support BIER and become a BFR. For example, once ASBR431 is upgraded to a BFR, ASBR311 no longer needs to tunnel traffic to EBFR41/EBFR42/EBFR43 but only need to tunnel one copy to ASBR431, who will then replicate to EBFR41/EBFR42/EBFR43.

1.2.1. IGP/BGP Interworking

Additionally, if enough routers in an AS (or just one of its IGP areas) can be upgraded to run BIER, then hop-by-hop BIER forwarding can be utilized there, using IGP extensions for BIER signaling [RFC8401] [RFC8444].

Notice that even with this there is still only one BIER sub-domain, with mixed IGP and BGP signaling for BIER. To redistribute BIER

information between IGP and BGP, procedures specified in [I-D.zwzw-bier-prefix-redistribute] and detailed in Section 2.2 are followed.

1.3. A CDN That Involves Another Providers

In the above example, the CDN is providing multicast transport service, with simplicity and scalability provided by BIER (the per-flow state is confined to the edges). Now let us go one step further and consider that AS300 belongs to a different Internet Service Provider. Now the ISP is providing BIER As A Service (BAAS) to the CDN, by being part of the CDN's BIER sub-domain. Notice that, not only does the ISP not have per-tree state (it does not have EBFRs), but also its BFRs do not need BFR-ID assigned. The ISP does need to learn about all the EBFRs and their corresponding BFR-IDs (through signaling).

1.3.1. Providing Independent BAAS To Multiple Customers

Now consider that the ISP also provides BAAS for another CDN. Each of the two CDNs has its own BIER domain, with their own BFR-ID or even sub-domain ID assignment that could conflict between the two CDNs. For example, both have BFR-ID 100 and sub-domain ID 0 assigned but they are totally independent of each other. For an BFR in the ISP to support this, with BGP signaling it needs to advertise its own BFR prefix multiple times, each time with a different RD that is mapped to the corresponding CDN. A new SAFI BIER (to be allocated by IANA) is used.

In the above example, there are two paths between AS100 and AS300. It is possible that while ASBR311 is the BFR, ASBR312 is the unicast best path into AS300 and beyond from AS100. Advertising BIER prefixes using a different SAFI with a RD also has the side benefit of allowing incongruent topologies for unicast and BIER.

In the existing BIER architecture and IGP extensions for BIER a sub-domain is tied to a single topology (either the one and only topology if Multi-topology ISIS/OSPF is not used, or a topology as defined in Multi-topology ISIS/OSPF). In the BIER sub-TLV that ISIS/OSPF attaches to a BIER prefix, a Sub-domain-ID value can only appear once for a particular topology. In this document, a BFR in the BAAS provider may belong to different and independent BIER domains, and the same sub-domain ID needs to be signaled multiple times, once for each BIER domain (notice that the same sub-domain-ID actually identifies different sub-domains in different BIER domains, so this does not really change the architectural requirement that a sub-domain is tied to a single topology). To do so, a new "BIER Domain" sub-TLV is introduced, and its value field includes a RD (as in the

BGP signaling) and a BIER sub-sub-TLV that is the same as currently specified in ISIS/OSPF extensions for BIER.

This works very well because of the flexible BIER architecture - a BIER packet is forwarded based on a Bit Index Forwarding Table (BIFT) that is determined by a 20-bit BIFT ID in front of the BIER header, and each (subdomain, BSL, set) tuple has its own BIFT. Traditionally, a subdomain is identified by a sub-domain ID but in this document a subdomain is now identified by a (RD, sub-domain ID) tuple in the control plane.

With this, the scaling aspect on a BFR comes to how many BAAS customer the provider needs to support. For example, if it needs to support 16 BAAS customers, one BSL, and four sets (Section 1.4.1) for each customer, then the provider needs to support 64 BIFTs (16 x 1 x 4). If the BSL is 256, then each BIFT has 256 entries in it and the total number of BIFT entries (routes) is 4k (256 x 64). Notice that this 4k number is not related to the number of customers' multicast flows, but only related to the number of customers and number of customer EBFRs. The number of customers with their own independent BIER domains are likely not very large initially, but if multicast as a service gets more widely used, the protocol and procedures defined in this document can scale up to the extent of how many BIFTs (and BIFT entries) a BFR can support (notice that there is no real difference between a BIFT entry and a unicast RIB/FIB entry).

1.3.2. Control and Accounting

With BGP based signaling, internal routers of a BAAS provider does not need explicit configuration for the BIER transport services that it support. In the above example, the ASBRs (ASBR311, ASBR312, ASBR321, ASBR341, ASBR351) in AS300 only need to have BGP policy configured to allow certain received BIER prefix advertisements to trigger necessary BIER state and additional signaling of their own. For example, when ASBR351 receives the BIER prefix advertisement, if its local configuration allows it may create corresponding BIFTs and BIFT entries, and additionally originates or updates its own BIER prefix advertisement. An internal BFR inside AS300, upon receiving the BGP advertisements, may or may not need to go through the same policy check again (based on the providers operation model).

When the ASBRs (re-)advertise BIER prefixes toward their external peers, they could enable statistics counters for the corresponding BIER labels so that they can count incoming BIER packets from external peers specifically for this BAAS. Similarly, the ASBRs can enable statistics counters for BIER labels they receive from external peers, so that they can count outgoing BIER packets delivered to the

external peers. These incoming and outgoing counters can be used for accounting and billing purposes.

1.4. Sets and Segmentation

The number of EBFRs could very well be larger than the BSL. There are two ways to handle that - multiple sets or segmentation.

1.4.1. Multiple Sets

With this method the set of EBFRs are grouped into multiple sets, and the number of EBFRs in a set is smaller than the BSL. A BFIR may need to send multiple copies of a multicast packet to reach all BFERs, one copy for each set that covers one or more expecting BFERs. A separate BIFT is needed for each set (because the same bit in the BitString of packets for different sets maps to different BFERs). This not only leads to multiple copies to be sent over the same link, but also requires additional BIFTs. In the earlier example, 64 BIFTs are needed for 16 BAAS customers because each customer needs 4 BIFTs for the multiple sets.

1.4.2. Segmentation

With this method, a BIER network is segmented into multiple regions, each with its own BIER sub-domain. In the earlier example, each AS could be an independent sub-domain. A BIER packet from EBFR11 will be decapsulated by the segmentation border router ASBR311, and then sent into next sub-domain in AS300 with a new BIER header. The segmentation [RFC7524] involves Multicast Flow Overlay [RFC8279] [RFC8556] so that the segmentation border routers know what BitString to use when sending onto the next segment. The advantage of segmentation is that only a single copy needs to be sent, and the number of BIFTs is also reduced on all BFRs. The disadvantage is that the segmentation points need to run multicast flow overlay protocol and maintain related state in control plane and data plane.

A deployment may start without the need for either multiple sets or segmentation when the number of EBFRs is small. When the number of EBFRs grows, segmentation can be introduced incrementally. A new BFR can be added as, or an existing BFR could be converted to, a segmentation point, splitting the original sub-domain into two independent sub-domains. The segmentation point does not re-advertise BIER information from one sub-domain to another. Other BFRs/EBFRs do not need any configuration changes except to make sure that all BIER information exchange is restricted to a single sub-domain (for example, two BFRs were BGP peers before and were exchanging BIER information but now they belong to two sub-domains

and only exchange BIER information with the segmentation point and other BFRs in the same sub-domain).

In the earlier example of a CDN of a single provider, using segmentation may be acceptable, even though the overlay state needs to be kept by the segmentation points. A BAAS provider may need to carefully consider if it wants to keep a customer's overlay state on those segmentation points. On the other hand, the provider may consider hosting per-customer segmentation points. For example, tethering small or virtual BFRs to an ASBR and have those BFRs be the segmentation points [I-D.zzhang-bier-tether].

2. Specifications for Enhancements to BIER Signaling with BGP/IGP

2.1. BGP Procedures

When an EBFR advertises a BIER prefix with a BIER Path Attribute (BPA), it SHOULD attach a Tunnel Encap Attribute (TEA) with the tunnel destination set to itself.

A BFR receiving the advertisement MUST use the tunnel destination in the TEA to determine where to forward a BIER packet whose BitString has a set bit corresponding to the BIER prefix, unless the TEA does not exist, in which case the BIER prefix itself is used for the determination. When the BFR re-advertises the BIER prefixes, it MUST change the tunnel destination in the TEA to itself, or add a TEA with the tunnel destination set to itself if there was no TEA in the received advertisement.

The TEA SHOULD have a Protocol Sub-TLV with protocol type BIER (0xAB37).

A transit BFR that is allowed (by provisioning or based on policy) to participate in a BIER sub-domain MUST advertise its own BIER prefix with a BPA. The BFR-id in the BPA SHOULD be 0. Depending on the operational model of the operator, the advertisement MAY be based on received BIER prefixes (subject to certain BGP policy verification), or MAY do so only with explicit configuration.

If a provider provides independent BAAS services to multiple customers, when its BFR receives BIER prefixes from a customer it MUST re-advetise with a new BIER SAFI. For simplicity, all BFRs of the provider use the same RD that is specifically assigned for the customer. When a BFR re-advertises BIER prefixes to a customer, it MUST re-advertise with SAFI 1 or 2.

If multiple providers together provide BAAS to a customer, then the two providers may assign the same RD for the customer or do RD

rewriting when re-advertising BIER prefixes from one provider to another.

2.2. ISIS/OSPF Procedures

This document defines a new BIER Domain Sub-TLV of ISIS TLVs 135, 235, 236, and 237. The sub-TLV type is to be allocated.

This document also defines a new BIER Domain Sub-TLV of OSPF Extended Prefix TLV. The sub-TLV type is to be allocated.

The value part of the BIER Domain Sub-TLV includes a 64-bit Route Distinguisher followed by one or more BIER Info Sub-TLV (as defined in [RFC8401] and [RFC8444] respectively) as its sub-sub-TLVs .

When a BFR redistribute a BIER prefix from BGP into ISIS/OSPF, if the BGP advertisement is of BIER SAFI, a BIER Domain sub-TLV is attached, with the RD part of the sub-TLV copied from the BGP advertisement. For each BIER TLV in the BPA, a BIER Info sub-sub-TLV is added in the BIER Domain sub-TLV, with the subdomain-id and BFR-id copied from the corresponding BIER TLV in the BPA, and the Encapsulation sub-sub-sub-TLV omitted because it is not needed.

If the BGP advertisement is of SAFI 1 or 2, BIER Info Sub-TLVs are constructed as above directly, without using a BIER Domain sub-TLV.

When a BFR redistribute a BIER prefix from ISIS/OSPF into BGP, if there is a BIER Domain sub-TLV in the corresponding ISIS LSP or OSPF LSA, the BGP advertisement is of BIER SAFI and the RD part of the NLRI is set to the RD from the BIER Domain sub-TLV. For each BIER Info sub-sub-TLV in the BIER Domain sub-TLV, a BIER TLV is included in the BPA, with the subdomain-id and BFR-id copied from the corresponding BIER Info sub-sub-TLV. The MPLS Encapsulation sub-TLV is omitted. The tunnel destination in the TEA is set to the BFR's BIER prefix.

If there is no BIER Domain sub-TLV in the corresponding ISIS LSP or OSPF LSA for the BIER Prefix, the BGP advertisement is of SAFI 1 or 2, and the BPA is constructed similar to the above (the only difference is that in this case BIER Info sub-TLVs are not part of a BIER Domain sub-TLV).

3. IANA Considerations

This document requests the following IANA assignments:

- A sub-TLV type for BIER Domain Sub-TLV from ISIS "Sub-TLVs for TLVs 135, 235, 236, and 237" registry.

- o A sub-TLV type for BIER Domain Sub-TLV from OSPFv2 Extended Prefix Sub-TLV registry.

- o A BIER SAFI from Subsequent Address Family Identifiers (SAFI) registry.

4. Security Considerations

To be provided.

5. Acknowledgements

The authors thank Lenny Giuliano and Antoni Przygenda for their review and suggestions.

6. References

6.1. 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-07 (work in progress), September 2019.

[I-D.ietf-idr-tunnel-encaps]
Patel, K., Velde, G., and S. Ramachandra, "The BGP Tunnel Encapsulation Attribute", draft-ietf-idr-tunnel-encaps-15 (work in progress), December 2019.

[I-D.zwzw-bier-prefix-redistribute]
Zhang, Z., Bo, W., Zhang, Z., Wijnands, I., and Y. Liu, "BIER Prefix Redistribute", draft-zwzw-bier-prefix-redistribute-05 (work in progress), February 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>.

[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>.

[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>.

6.2. Informative References

[I-D.zzhang-bier-tether]
Zhang, Z., Warnke, N., Wijnands, I., and D. Awduche, "Tethering A BIER Router To A BIER incapable Router", draft-zzhang-bier-tether-05 (work in progress), April 2020.

[RFC7524] Rekhter, Y., Rosen, E., Aggarwal, R., Morin, T., Grosclaude, I., Leymann, N., and S. Saad, "Inter-Area Point-to-Multipoint (P2MP) Segmented Label Switched Paths (LSPs)", RFC 7524, DOI 10.17487/RFC7524, May 2015, <https://www.rfc-editor.org/info/rfc7524>.

[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

Zhaohui Zhang
Juniper Networks

EMail: zzhang@juniper.net

Eric Rosen
Individual

EMail: erosen52@gmail.com

Daniel Awduche
Verizon

EMail: daniel.awduche@verizon.com

Liang Geng
China Mobile

EMail: gengliang@chinamobile.com

BIER
Internet-Draft
Intended status: Standards Track
Expires: October 16, 2020

Z. Zhang
Juniper Networks
N. Warnke
Deutsche Telekom
I. Wijnands
Cisco Systems
D. Awduche
Verizon
April 14, 2020

Tethering A BIER Router To A BIER incapable Router
draft-zzhang-bier-tether-05

Abstract

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

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 October 16, 2020.

Table of Contents

1. Introduction 2
2. Additional Considerations 3
3. Specification 5
3.1. IGP Signaling 5
3.2. BGP Signaling 6
4. Security Considerations 7
5. IANA Considerations 7
6. Acknowledgements 8
7. Normative References 8
Authors' Addresses 9

1. Introduction

Consider the scenario in Figure 1 where router X does not support BIER.

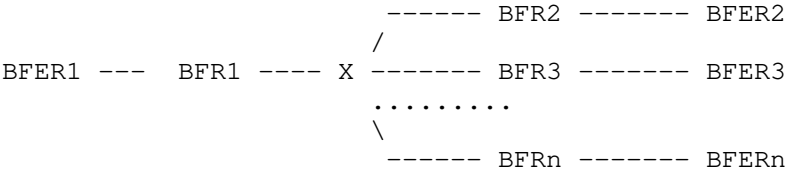


Figure 1: Deployment with BIER incapable routers

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 attach (tether) a BFRx to X as depicted in Figure 2:

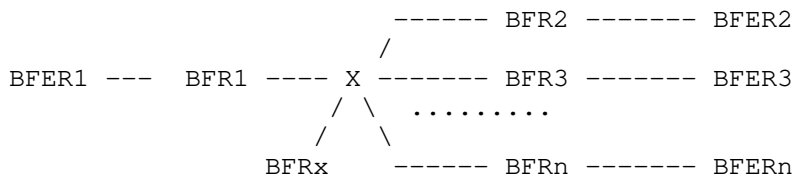


Figure 2: Tethered BFRx

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 Interial Gateway Protocol (IGP) as a forwarding adjacency so that BFRx will appear on the Shortest Path First (SPF) tree. This needs 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. 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 specification [RFC8279].

2. 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.

Additionally, the helper BRFx 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. Figure 3 below is a simple case:

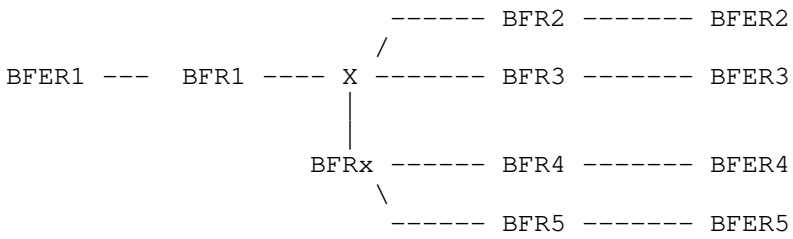


Figure 3: A Safe Transit Helper

In the example of Figure 4, 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.

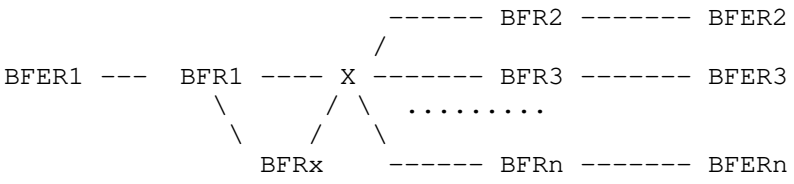


Figure 4: Potential looping situation

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 not different from the SPF done for a neighbor as part of Loop-Free Alternate (LFA) calculation. In fact, BFR1 tunneling packets to X's helper is not 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 situation in Figure 5 where a helper BFRxy helps two different non-BFRs X and Y also works. It's just a special situation of a transit helper.

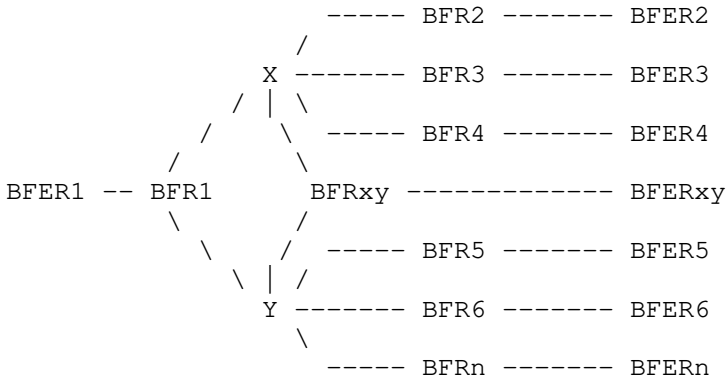


Figure 5: One Helper for multiple helped

3. 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.

3.1. IGP Signaling

Suppose that the BIER domain uses BIER signaling extensions to ISIS [RFC8401] or OSPF [RFC8444]. The helper node (BFRx) MUST advertise one or more BIER Helped Node sub-sub-TLVs (one for each helped node). The value is BIER prefix of the helped node (X) followed by a one-octet priority field, and one-octet reserved field. The length is 6 for IPv4 and 18 for IPv6 respectively.

The post-SPF processing procedures in Section 6.9 of the BIER architecture specification [RFC8279] are modified as following for BIER tethering purpose.

At step 2, the removed node is added to an ordered list maintained with each child that replaces the node. If the removed node already

has a non-empty list maintained with itself, add the removed node to the tail of the list and copy the list to each child.

At the end, the calculating node BFR-B would use a unicast tunnel to reach next hop BFRs for some BFERs. The next hop BFR has an ordered list created at step 2 above, recording each BIER incapable node replaced by their children along the way. For a particular BFER to be reached via a tunnel to the next hop BFR, additional procedures are performed as following.

- Starting with the first node in the ordered list of incapable nodes, say N1, check if there is one or more helper nodes for N1. If not, go the next node in the list.

- Order all the helper nodes of N1 based descending (priority, BIER prefix). Starting with the first one, say H1, check if BFR-B could use H1 as LFA next hop to reach the BFER. If yes, H1 is used as the next hop BFR for the BFER and the procedure stops. If not, go to the next helper in order.

- If none of the helper nodes of N1 can be used, go to the next node in the list of incapable nodes.

If the above procedure finishes without finding any helper, then the original BFR next hop via a tunnel is used to reach the BFER.

3.2. BGP Signaling

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

(1) X does not participate in BGP peering at all

(2) X re-advertises the BIER prefixes but does not do next-hop-self

(3) X re-advertises the BIER prefixes and does next-hop-self

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

To make tethering work well with BGP signaling, the following can be done:

- 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.

- 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.

- 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.

4. 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.

5. IANA Considerations

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

Type	Name
TBD1	BIER Helped Node

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

Type	Name
TBD2	BIER Helped Node

6. Acknowledgements

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

7. 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-07 (work in progress), September 2019.

[I-D.ietf-idr-tunnel-encaps]
Patel, K., Velde, G., and S. Ramachandra, "The BGP Tunnel Encapsulation Attribute", draft-ietf-idr-tunnel-encaps-15 (work in progress), December 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>.

[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>.

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