

Networking Working Group
Internet-Draft
Intended status: Standards Track
Expires: May 19, 2019

Ran. Chen
Zheng. Zhang
ZTE Corporation
November 15, 2018

PCEP Extensions for BIER
draft-chen-pce-bier-04

Abstract

Bit Index Explicit Replication (BIER)-TE shares architecture and packet formats with BIER as described in [I-D.ietf-bier-architecture]. BIER-TE forwards and replicates packets based on a BitString in the packet header, but every BitPosition of the BitString of a BIER-TE packet indicates one or more adjacencies. BIER-TE Path can be derived from a Path Computation Element (PCE).

This document specifies extensions to the Path Computation Element Protocol (PCEP) to handle requests and responses for the computation of paths for BIER-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 May 19, 2019.

Copyright Notice

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

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of

publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	2
2. Conventions used in this document	3
3. Overview of PCEP Operation in BIER Networks	3
4. BIER PCEP Message Extensions	3
4.1. BIER Capability Advertisement	3
4.1.1. The OPEN Object	3
4.1.1.1. The BIER PCE Capability TLV	3
4.2. Path Computation Request/Reply Message Extensions	4
4.2.1. The RP/SPR Object	4
4.2.2. The New BIER END-POINT Object	5
4.2.3. ERO Object	5
4.2.3.1. BIER-ERO Subobject	6
4.2.4. RRO Object	7
4.2.4.1. RRO Processing	7
5. Security Considerations	7
6. IANA Considerations	7
6.1. PCEP Objects	7
6.2. PCEP-Error Objects and Types	8
6.3. PCEP TLV Type Indicators	8
6.4. New Path Setup Type	8
7. References	8
7.1. Normative references	9
7.2. Informative references	9
Authors' Addresses	9

1. Introduction

Bit Index Explicit Replication (BIER)-TE shares architecture and packet formats with BIER as described in [I-D.ietf-bier-architecture]. BIER-TE forwards and replicates packets based on a BitString in the packet header, but every BitPosition of the BitString of a BIER-TE packet indicates one or more adjacencies. BIER-TE Path can be derived from a Path Computation Element (PCE).

This document specifies extensions to the Path Computation Element Protocol (PCEP) to handle requests and responses for the computation of paths for BIER-TE.

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. Overview of PCEP Operation in BIER Networks

BIER-TE forwards and replicates packets based on a BitString in the packet header. In a PCEP session, An ERO object specified in [RFC5440] can be extended to carry a BIER-TE path consists of one or more BIER-ERO subobject(s). BIER-TE computed by a PCE can be represented in the following forms:

- o An ordered set of adjacencies BitString(s) in which each bit represents that the adjacencies to which the BFR should replicate packets to in the domain.

In this document, we define a set of PCEP protocol extensions, including a new PCEP capability, a new Path Setup Type (PST), a new BIER END-POINT Object, new ERO subobjects, new RRO subobjects, new PCEP error codes and procedures.

4. BIER PCEP Message Extensions

The following section describes the protocol extensions required to support BIER-TE path.

4.1. BIER Capability Advertisement

4.1.1. The OPEN Object

This document defines a new optional TLV for use in the OPEN Object.

4.1.1.1. The BIER PCE Capability TLV

The BIER-PCE-CAPABILITY TLV is an optional TLV associated with the OPEN Object to exchange BIER capability of PCEP speakers. The format of the BIER-PCE-CAPABILITY TLV is shown in the following figure:

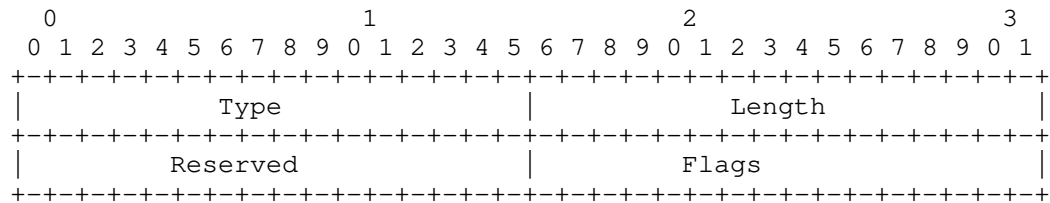


Figure 1

The code point for the TLV type is to be defined by IANA.

Length: 4 bytes.

The "Reserved" (2 octet) and "Flags" (2 octet) fields are currently unused, and MUST be set to zero on transmission and ignored on reception.

4.1.1.1. Exchanging BIER Capability

This document defines a new optional BIER-PCE-CAPABILITY TLV for use in the OPEN object to negotiate the BIER capability. The inclusion of this TLV in the OPEN message destined to a PCC indicates the PCE's capability to perform BIER-TE path computations, and the inclusion of this TLV in the OPEN message destined to a PCE indicates the PCC's capability to support BIER-TE Path.

A PCE that is able to support the BIER extensions defined in this document SHOULD include the BIER-PCE-CAPABILITY TLV on the OPEN message. If the PCE does not include the BIER-PCE-CAPABILITY TLV in the OPEN message and PCC does include the TLV, it is RECOMMENDED that the PCC indicates a mismatch of capabilities.

4.2. Path Computation Request/Reply Message Extensions

4.2.1. The RP/SPR Object

In order to setup an BIER-TE, a new PATH-SETUP-TYPE TLV[I-D.ietf-pce-lsp-setup-type] MUST be contained in RP or SRP object. This document defines a new Path Setup Type (PST) for BIER as follows:

- o PST = 2: Path is setup using BIER Traffic Engineering technique.

If a PCEP speaker does not recognize the PATH-SETUP-TYPE TLV, it MUST ignore the TLV in accordance with [RFC5440]. If a PCEP speaker recognizes the TLV but does not support the TLV, it MUST send PCErr with Error-Type = 2 (Capability not supported).

4.2.2. The New BIER END-POINT Object

The END-POINTS object is used in a PCReq message to specify the BIER information of the path for which a path computation is requested. To represent the end points for a BIER path efficiently, we define a new END-POINT Object for the BIER path:

The format of the new END-POINTS Object is as follows:

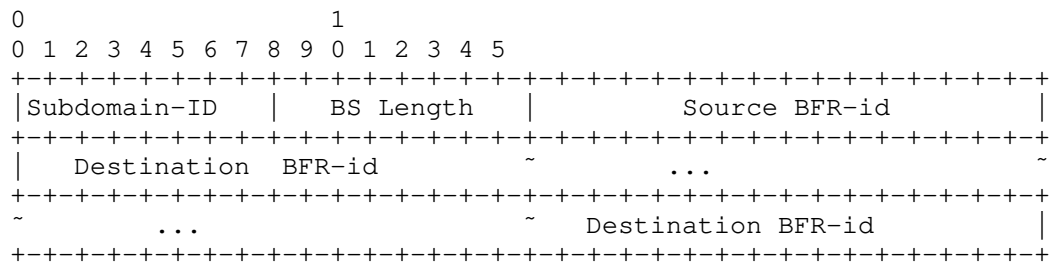


Figure 2

Subdomain-id: Unique value identifying the BIER sub-domain. 1 octet

BS Length: A 1 octet field encoding the supported BitString length.

Source BFR-id: A 2 octet field encoding the source BFR-id.

Destnation BFR-id: A 2 octet field encoding the destnation BFR-id.

4.2.3. ERO Object

BIER-TE consists of one or more adjacencies BitStrings where every BitPosition of the BitString indicates one or more adjacencies, as described in([I-D.eckert-bier-te-arch]).

The ERO object specified in [RFC5440] is used to encode the path of a TE LSP through the network. The ERO is carried within a PCRep message to provide the computed TE LSP if the path computation was successful. In order to carry BIER-TE explicit paths, this document defines a new ERO subobjects referred to as "BIER-ERO subobjects" whose formats are specified in the following section. An BIER-ERO subobjects carrying a adjacencies BitStrings consists of one or more BIER-ERO subobject(s).

4.2.3.1. BIER-ERO Subobject

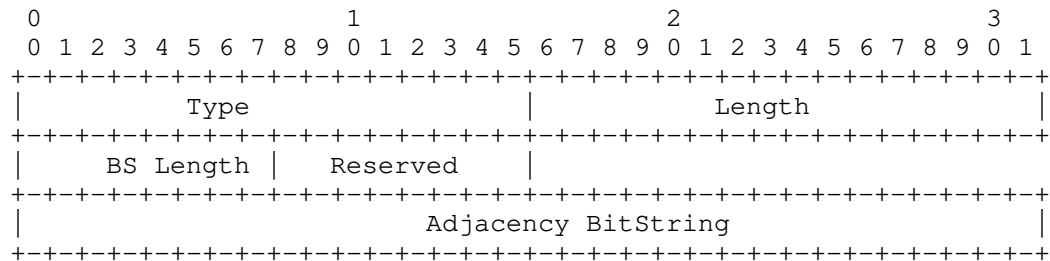


Figure 3

Type: TBD

Length: 4 bytes

BS Length: A 1 octet field encoding the supported BitString length.

The "Reserved" (1 octets) fields are currently unused, and MUST be set to zero on transmission and ignored on reception.

Adjacency BitString: A 4 octet field encoding the Adjacency BitString where every BitPosition of the BitString indicates one or more adjacencies.

4.2.3.1.1. BIER-ERO Processing

If a PCC finds a non-recognize the BIER-ERO subobject, the PCC MUST respond with a PCErr message with Error-Type=3 ("Unknown Object") and Error-Value=2 ("Unrecognized object Type") or Error-Type=4 ("Not supported object") and Error-Value=2 ("Not supported object Type") as described in [RFC5440] .

If a PCC receives an BIER-ERO subobject in which either BitStringLength or Adjacency BitString is absent, it MUST consider the entire BIER-ERO subobject invalid and send a PCErr message with Error-Type = 10 ("Reception of an invalid object") and Error-Value = TBD ("BitStringLength is absent ") and Error-Value = TBD ("Adjacency BitString is absent ")

If a PCC detects that all subobjects of BIER-ERO are not identical, it MUST send a PCErr message with Error-Type = 10 ("Reception of an invalid object") and Error-Value = TBD ("Non-identical BIER-ERO subobjects").

If a PCC receives an BIER-ERO subobject in which BitStringLength values are not chosen from: 64, 128, 256, 512, 1024, 2048, and 4096, as it described in ([I-D.ietf-bier-architecture]). The PCC MUST send a PCErr message with Error-Type = 10 ("Reception of an invalid object") and Error-Value = TBD ("Invalid BitStringLength").

4.2.4. RRO Object

A PCC can record BIER-ERO explicit paths and report the paths to a PCE via RRO. An RRO object contains one or more subobjects called "BIER-RRO subobjects" whose formats are the same as that of BIER-ERO subobject.

4.2.4.1. RRO Processing

Processing rules of BIER-RRO subobject are identical to those of BIER-ERO subobject defined in section 4.2.3.1 in this document.

5. Security Considerations

TBD.

6. IANA Considerations

6.1. PCEP Objects

As discussed in Section 4.2.2, a new END-POINTS Object-Type is defined. IANA has made the following Object-Type allocations from the "PCEP Objects" sub-registry:

Object	Object-Class Value
BIER END-POINT Object	TBD

As discussed in Section 4.2.3 and 4.2.4, a new sub-object type for the PCEP explicit route object (ERO), and a new sub-object type for the PCEP record route object (RRO) are defined.

IANA has made the following sub-objects allocation from the RSVP Parameters registry:

Object	Sub-Object	Sub-Object Type

EXPLICIT_ROUTE	BIER-ERO (PCEP-specific)	TBD
ROUTE_RECORD	BIER-RRO (PCEP-specific)	TBD

6.2. PCEP-Error Objects and Types

As described in Section 4.2.3.1.1, a number of new PCEP-ERROR Object Error Values have been defined.

Error-Type	Meaning	Reference

10	Reception of an invalid object.	RFC5
540	Error-value = TBD: BitStringLength is absent	This document
	Error-value = TBD: BitString is absent	This document
	Error-value = TBD: Invalid BitStringLength	This document

6.3. PCEP TLV Type Indicators

IANA is requested to allocate a new code point in the PCEP TLV Type Indicators registry, as follows:

Value	Meaning	Reference

TBD	BIER-PCE-CAPABILITY TLV	This document

6.4. New Path Setup Type

IANA is requested to allocate a new code point in the PCEP PATH_SETUP_TYPE TLV PST field registry, as follows:

Value	Description	Reference

2	Path is setup using BIER Traffic Engineering technique	This document

7. References

7.1. Normative references

- [I-D.ietf-bier-architecture]
Wijnands, I., Rosen, E., Dolganow, A., Przygienda, T., and S. Aldrin, "Multicast using Bit Index Explicit Replication", draft-ietf-bier-architecture-08 (work in progress), September 2017.
- [I-D.ietf-pce-lsp-setup-type]
Sivabalan, S., Tantsura, J., Minei, I., Varga, R., and J. Hardwick, "Conveying path setup type in PCEP messages", draft-ietf-pce-lsp-setup-type-10 (work in progress), May 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>>.
- [RFC5440] Vasseur, JP., Ed. and JL. Le Roux, Ed., "Path Computation Element (PCE) Communication Protocol (PCEP)", RFC 5440, DOI 10.17487/RFC5440, March 2009, <<https://www.rfc-editor.org/info/rfc5440>>.

7.2. Informative references

- [I-D.eckert-bier-te-arch]
Eckert, T., Cauchie, G., Braun, W., and M. Menth, "Traffic Engineering for Bit Index Explicit Replication BIER-TE", draft-eckert-bier-te-arch-06 (work in progress), November 2017.

Authors' Addresses

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

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

Zheng Zhang
ZTE Corporation
No.50 Software Avenue, Yuhuatai District
Nanjing, Jiangsu Province 210012
China

Email: zhang.zheng@zte.com.cn

Internet Engineering Task Force
Internet-Draft
Updates: 8296 (if approved)
Intended status: Standards Track
Expires: August 2, 2019

S. Dhanaraj, Ed.
Huawei
IJ. Wijnands
P. Psenak
Cisco Systems, Inc.
Z. Zhang
Juniper Networks.
G. Yan
J. Xie
Huawei
January 29, 2019

LSR Extensions for BIER over Ethernet
draft-dhanaraj-bier-lsr-ethernet-extensions-00

Abstract

Bit Index Explicit Replication (BIER) [RFC8279] is an architecture that provides multicast forwarding through a "BIER domain" without requiring intermediate routers to maintain multicast related per-flow state. BIER can be supported in MPLS and non-MPLS networks. The common BIER header format and encapsulation for MPLS and non-MPLS networks is specified in [RFC8296].

This document specifies the required extensions to the IS-IS [RFC1195] and OSPFv2 [RFC2328] protocol for supporting BIER in non-MPLS networks using BIER in Ethernet encapsulation.

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 August 2, 2019.

Copyright Notice

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

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

Table of Contents

1. Introduction	2
2. Terminology	4
2.1. Requirements Language	5
3. Specification	5
3.1. IS-IS BIER Ethernet Encapsulation Sub-sub TLV	5
3.2. OSPFv2 BIER Ethernet Encapsulation Sub-TLV	6
4. Security Considerations	8
5. IANA Considerations	8
5.1. IS-IS sub-sub-TLVs for BIER Info sub-TLV Registry	8
5.2. OSPFv2 Extended Prefix TLV Sub-TLVs Registry	8
6. Acknowledgments	9
7. References	9
7.1. Normative References	9
7.2. Informative References	9
Authors' Addresses	11

1. Introduction

Bit Index Explicit Replication (BIER) [RFC8279] is an architecture that provides multicast forwarding through a "BIER domain" without requiring intermediate routers to maintain multicast related per-flow state. BIER can be supported in MPLS and non-MPLS networks.

[RFC8296] specifies a common BIER header format for both MPLS and non-MPLS networks, though the first 20-bits of the BIER header (referred as BIFT-id) is a "MPLS Label" in case of MPLS networks and is a "domain-wide-unique-value" representing the combination of SD-BSL-SI in case of non-MPLS networks.

[I-D.ietf-bier-non-mpls-bift-encoding] specifies two optional ways of statically assigning domain-wide-unique mapping between BIFT-IDs and SD-BSL-SI combination.

However, BIER architecture [RFC8279] does NOT require domain-wide-unique BIFT-IDs to be used (even for non-MPLS encapsulation). As discussed in [I-D.zzhang-bier-rift], the BIFT-ID in case of non-MPLS encapsulation can also just be a local 20-bit opaque value and signaled just like in MPLS case.

As an example, suppose a particular BIER domain contains a SD (SD 0), supports two BSLs (256 and 512), and contains 1024 BFRs. A BFR that is provisioned for above SD, and that supports both BSLs, could advertise the following set of BIFT-id's:

BIFT-id 1: corresponding to SD 0, BSL 256, SI 0.

BIFT-id 2: corresponding to SD 0, BSL 256, SI 1.

BIFT-id 3: corresponding to SD 0, BSL 256, SI 2.

BIFT-id 4: corresponding to SD 0, BSL 256, SI 3.

BIFT-id 5: corresponding to SD 0, BSL 512, SI 0.

BIFT-id 6: corresponding to SD 0, BSL 512, SI 1.

Notice that the example uses ranges of continuous BIFT-ids:

BIFT-id range [1 to 4] correspond to <SD 0, BSL 256>. The first BIFT-id in the range correspond to SI=0, the second correspond to SI=1, and so on.

BIFT-id range [5 to 6] correspond to <SD 0, BSL 512>. The first BIFT-id in the range correspond to SI=0, the second correspond to SI=1.

Strictly speaking, using contiguous range is not required, but it is done for the purpose of simplified signaling similar to MPLS label blocks (notice that locally assigning BIFT-ID ranges requires no manual processing just like in the case of MPLS label block allocation).

Processing and forwarding of BIER packets requires special software and hardware capabilities. The BFRs supporting a BIER encapsulation type MUST advertise this capability (along with the other required parameters specific to the encapsulation) to the other routers in BIER domain. This advertisement, for example, will enable the other BFRs in the BIER domain in deciding, whether to include or exclude the advertising router from the BAR and/or IPA algorithm while computing the multicast path for a specific encapsulation type.

[RFC8401] and [RFC8444] specifies the required extensions to the IS-IS [RFC1195] and OSPFv2 [RFC2328] protocol respectively for the distribution of BIER sub-domain information including the Sub-sub-TLV required to support BIER in MPLS encapsulation for MPLS networks.

This document specifies the required extensions to the IS-IS [RFC1195] and OSPFv2 [RFC2328] protocol for supporting BIER using BIER in Ethernet encapsulation with dynamically and locally assigned BIFT-IDs.

Support for other encapsulation types are outside the scope of this document.

2. Terminology

Some of the terminology specified in [RFC8279] is replicated here and extended by necessary definitions:

BIER: Bit Index Explicit Replication

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

BIER-MPLS: BIER in MPLS encapsulation.

(Encapsulation of BIER header inside MPLS header in MPLS networks).

BIER-ETH: BIER in Ethernet encapsulation.

(Encapsulation of BIER header inside Ethernet header (EtherType=0xAB37) in non-MPLS networks).

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

BIFT: Bit Index Forwarding Table used to forward the BIER packets in a domain.

BAR: BIER Algorithm. Used to calculate underlay nexthops as defined by the BAR value.

IPA: IGP Algorithm. May be used to modify, enhance or replace the calculation of underlay paths as defined by the BAR value

2.1. Requirements Language

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

3. Specification

A BIER sub-domain MAY support multiple BIER encapsulation types like BIER-MPLS, BIER-ETH. The different encapsulation types supported by a BFR in a sub-domain MUST share the same BFR-id. This would allow the BFR's in transit to translate the encapsulation from one type to the other while forwarding the packet in the BIER sub-domain.

When a BFIR/BFR supports multiple BIER encapsulation types, when sending to a BIER neighbor it MUST use a type that the neighbor also supports. If the neighbor also supports more than one encapsulation type that this BFIR/BFR supports, the type selection could be a matter of local policy and is outside the scope of this document.

3.1. IS-IS BIER Ethernet Encapsulation Sub-sub TLV

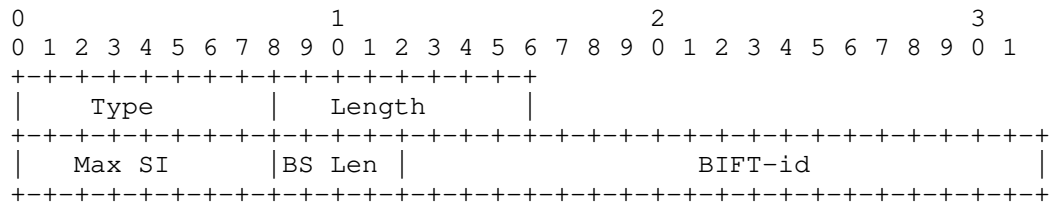
BIER Info sub-TLV defined in [RFC8401] is used to advertise the sub-domain id, and other associated parameters of the sub-domain like BFR-id, MT, BAR, IPA.

This document introduces new sub-sub-TLV under BIER Info sub-TLV to advertise the ethernet encapsulation capability and other associated parameters of the encapsulation.

This sub-sub-TLV carries the information for the BIER Ethernet encapsulation including the BitString length supported for a certain <MT,SD> pair.

It is advertised within the BIER Info sub-TLV defined in [RFC8401] which in-turn is carried within the TLVs 235, 237 [RFC5120] or TLVs 135 [RFC5305], or TLV 236 [RFC5308].

This sub-sub-TLV MAY appear multiple times within a single BIER Info sub-TLV. If the same BitString length is repeated in multiple BIER Ethernet encapsulation sub-sub-TLVs inside the same BIER Info sub-TLV, the BIER Info sub-TLV MUST be ignored.



Type: 2 (suggested value - To be assigned by IANA).

Length: 4

Max SI: A 1 octet field encoding the Maximum Set Identifier (Section 1 of [RFC8279]) used in the encapsulation for this BIER subdomain for this BitString length. The first BIFT-id is for SI=0, the second BIFT-id is for SI=1, etc. If the BIFT-id associated with the Maximum Set Identifier exceeds the 20-bit range, the sub-sub-TLV MUST be ignored.

Local BitString Length (BS Len): A 4 bit field encoding the bitstring length (as per [RFC8296]) supported for the encapsulation.

BIFT-id: A 20 bit field encoding the first BIFT-id of the BIFT-id range.

The "BIFT-id range" is the set of 20-bit values beginning with the BIFT-id and ending with (BIFT-id + (Max SI)). A unique BIFT-id range is allocated for each BitString length and sub-domain-id. These BIFT-id's are used for BIER forwarding as described in [RFC8279] and [RFC8296].

The size of the BIFT-id range is determined by the number of SI's (Section 1 of [RFC8279]) that are used in the network. Each SI maps to a single BIFT-id in the BIFT-id range: the first BIFT-id is for SI=0, the second BIFT-id is for SI=1, etc.

If the BIFT-id associated with the Maximum Set Identifier exceeds the 20-bit range, the BIER Ethernet Encapsulation Sub-sub-TLV containing the error MUST be ignored.

3.2. OSPFv2 BIER Ethernet Encapsulation Sub-TLV

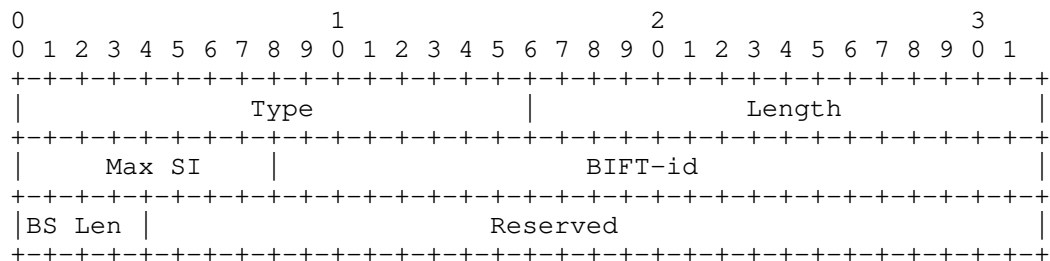
BIER Sub-TLV defined in [RFC8444] is used to advertise the sub-domain id, and other associated parameters of the sub-domain like BFR-id, MT, BAR, IPA.

This document introduces new Sub-TLV under BIER Sub-TLV to advertise the ethernet encapsulation capability and other associated parameters of the encapsulation.

This Sub-TLV carries the information for the BIER Ethernet encapsulation including the BitString length supported for a certain <MT,SD> pair.

It is advertised within the BIER Sub-TLV defined in [RFC8444] which in-turn is carried within the OSPFv2 Extended Prefix TLV defined in [RFC7684].

This Sub-TLV MAY appear multiple times within a single BIER Sub-TLV. If the same BitString length is repeated in multiple BIER Ethernet encapsulation Sub-TLVs inside the same BIER Sub-TLV, the BIER Sub-TLV MUST be ignored.



Type: 11 (suggested value - To be assigned by IANA).

Length: 8

Max SI: A 1 octet field encoding the Maximum Set Identifier (Section 1 of [RFC8279]) used in the encapsulation for this BIER subdomain for this BitString length. The first BIFT-id is for SI=0, the second BIFT-id is for SI=1, etc. If the BIFT-id associated with the Maximum Set Identifier exceeds the 20-bit range, the sub-sub-TLV MUST be ignored.

BIFT-id: A 3-octet field, where the 20 rightmost bits represent the first BIFT-id in the BIFT-id range. The 4 leftmost bits MUST be ignored.

The "BIFT-id range" is the set of 20-bit values beginning with the BIFT-id and ending with (BIFT-id + (Max SI)). A unique BIFT-id range is allocated for each BitString length and sub-domain-id. These BIFT-id's are used for BIER forwarding as described in [RFC8279] and [RFC8296].

The size of the BIFT-id range is determined by the number of SI's (Section 1 of [RFC8279]) that are used in the network. Each SI maps to a single BIFT-id in the BIFT-id range: the first BIFT-id is for SI=0, the second BIFT-id is for SI=1, etc.

If the BIFT-id associated with the Maximum Set Identifier exceeds the 20-bit range, the BIER Ethernet Encapsulation Sub-sub-TLV containing the error MUST be ignored.

Local BitString Length (BS Len): A 4 bit field encoding the bitstring length (as per [RFC8296]) supported for the encapsulation.

Reserved: SHOULD be set to 0 on transmission and MUST be ignored on reception.

4. Security Considerations

Security concerns for IS-IS are addressed in [RFC5304] and [RFC5310] and the security concerns for IS-IS extensions for BIER are addressed in [RFC8401]. This document introduces new sub-sub-TLV for the already existing IS-IS TLVs defined for distributing the BIER sub-domain information in [RFC8401]. It does not introduce any new security risks to IS-IS.

Security concerns and required extensions for OSPFv2 are addressed in [RFC2328] and [RFC7684] and the security concerns for OSPFv2 extensions for BIER are addressed in [RFC8444]. This document introduces new Sub-TLV for the already existing OSPFv2 TLV defined for distributing the BIER sub-domain information in [RFC8444]. It does not introduce any new security risks to OSPFv2.

5. IANA Considerations

The document requests new allocations from the IANA registries as follows

5.1. IS-IS sub-sub-TLVs for BIER Info sub-TLV Registry

BIER Ethernet Encapsulation sub-sub-TLV: 2 (suggested)

5.2. OSPFv2 Extended Prefix TLV Sub-TLVs Registry

BIER Ethernet Encapsulation Sub-TLV: 11 (suggested)

6. Acknowledgments

The author wants to thank Antonie Przygienda for his comments and suggestions.

7. References

7.1. 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>>.
- [RFC8279] Wijnands, IJ., Ed., Rosen, E., Ed., Dolganow, A., Przygienda, T., and S. Aldrin, "Multicast Using Bit Index Explicit Replication (BIER)", RFC 8279, DOI 10.17487/RFC8279, November 2017, <<https://www.rfc-editor.org/info/rfc8279>>.
- [RFC8296] Wijnands, IJ., Ed., Rosen, E., Ed., Dolganow, A., Tantsura, J., Aldrin, S., and I. Meilik, "Encapsulation for Bit Index Explicit Replication (BIER) in MPLS and Non-MPLS Networks", RFC 8296, DOI 10.17487/RFC8296, January 2018, <<https://www.rfc-editor.org/info/rfc8296>>.
- [RFC8401] Ginsberg, L., Ed., Przygienda, T., Aldrin, S., and Z. Zhang, "Bit Index Explicit Replication (BIER) Support via IS-IS", RFC 8401, DOI 10.17487/RFC8401, June 2018, <<https://www.rfc-editor.org/info/rfc8401>>.
- [RFC8444] Psenak, P., Ed., Kumar, N., Wijnands, IJ., Dolganow, A., Przygienda, T., Zhang, J., and S. Aldrin, "OSPFv2 Extensions for Bit Index Explicit Replication (BIER)", RFC 8444, DOI 10.17487/RFC8444, November 2018, <<https://www.rfc-editor.org/info/rfc8444>>.

7.2. Informative References

- [I-D.ietf-bier-non-mpls-bift-encoding] Wijnands, I., Xu, X., and H. Bidgoli, "An Optional Encoding of the BIFT-id Field in the non-MPLS BIER Encapsulation", draft-ietf-bier-non-mpls-bift-encoding-01 (work in progress), October 2018.

- [I-D.zzhang-bier-rift]
Zhang, Z., Ma, S., and Z. Zhang, "Supporting BIER with RIFT", draft-zzhang-bier-rift-00 (work in progress), March 2018.
- [RFC1195] Callon, R., "Use of OSI IS-IS for routing in TCP/IP and dual environments", RFC 1195, DOI 10.17487/RFC1195, December 1990, <<https://www.rfc-editor.org/info/rfc1195>>.
- [RFC2328] Moy, J., "OSPF Version 2", STD 54, RFC 2328, DOI 10.17487/RFC2328, April 1998, <<https://www.rfc-editor.org/info/rfc2328>>.
- [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>>.
- [RFC5304] Li, T. and R. Atkinson, "IS-IS Cryptographic Authentication", RFC 5304, DOI 10.17487/RFC5304, October 2008, <<https://www.rfc-editor.org/info/rfc5304>>.
- [RFC5305] Li, T. and H. Smit, "IS-IS Extensions for Traffic Engineering", RFC 5305, DOI 10.17487/RFC5305, October 2008, <<https://www.rfc-editor.org/info/rfc5305>>.
- [RFC5308] Hopps, C., "Routing IPv6 with IS-IS", RFC 5308, DOI 10.17487/RFC5308, October 2008, <<https://www.rfc-editor.org/info/rfc5308>>.
- [RFC5310] Bhatia, M., Manral, V., Li, T., Atkinson, R., White, R., and M. Fanto, "IS-IS Generic Cryptographic Authentication", RFC 5310, DOI 10.17487/RFC5310, February 2009, <<https://www.rfc-editor.org/info/rfc5310>>.
- [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>>.

Authors' Addresses

Senthil Dhanaraj (editor)
Huawei

Email: senthil.dhanaraj.ietf@gmail.com

IJsbrand Wijnands
Cisco Systems, Inc.

Email: ice@cisco.com

Peter Psenak
Cisco Systems, Inc.

Email: ppsenak@cisco.com

Zhaohui Zhang
Juniper Networks.

Email: zzhang@juniper.net

Gang Yan
Huawei

Email: yangang@huawei.com

Jingrong Xie
Huawei

Email: xiejingrong@huawei.com

BIER WG
Internet-Draft
Intended status: Standards Track
Expires: September 2, 2019

Quan Xiong
Greg Mirsky
Fangwei Hu
ZTE Corporation
Chang Liu
China Unicom
March 1, 2019

BIER BFD
draft-hu-bier-bfd-03.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 September 2, 2019.

Copyright Notice

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

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of

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

Table of Contents

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

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. BIER resilience use cases are described in [I-D.xiong-bier-resilience] including End-to-End 1+1 and 1:1 Protection and the failure detection mechanisms MAY use P2MP BFD and P2MP active tail detection method respectively.

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

[I-D.ietf-bfd-multipoint-active-tail] describes active tail extensions to the BFD protocol for multipoint 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 P2MP BFD as per [I-D.ietf-bfd-multipoint], and active tail as per [I-D.ietf-bfd-multipoint-active-tail] for BIER-specific domain.

2. Conventions used in this document

2.1. Terminology

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

BFD: Bidirectional Forwarding Detection.

OAM: Operations, Administration, and Maintenance.

P2MP: Point to Multi-Point.

2.2. Requirements Language

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

3. BIER BFD Encapsulation

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

4. Bootstrapping BIER BFD

4.1. BIER OAM Ping Bootstrap

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

The BFD discriminator TLV is a new TLV for BIER OAM TLV with the type (TBD2 by IANA) and the length of 4. The value contains the 4-byte local discriminator generated by BFIR for this session.

4.2. IGP protocol Bootstrap

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

4.2.1. IS-IS extension for BIER BFD

The new BIER BFD Sub-TLV is carried in the ISIS router capability TLV. The format is as follows.

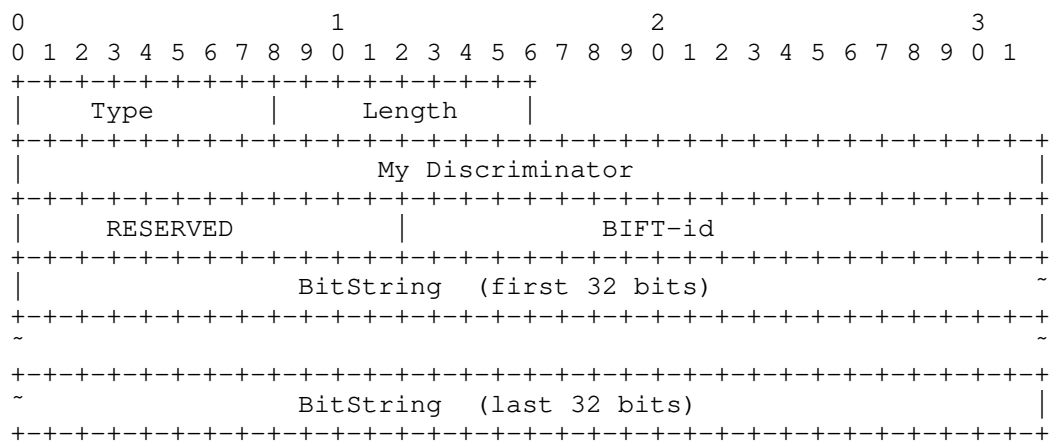


Figure 1: BIER BFD Sub-TLV for IS-IS extension

Type : TBD3 by IANA.

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

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

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

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

4.2.2. OSPF extension for BIER BFD

The new BIER BFD Sub-TLV is carried in the Router Information Link State Advertisement (LSA). The format is as follows.

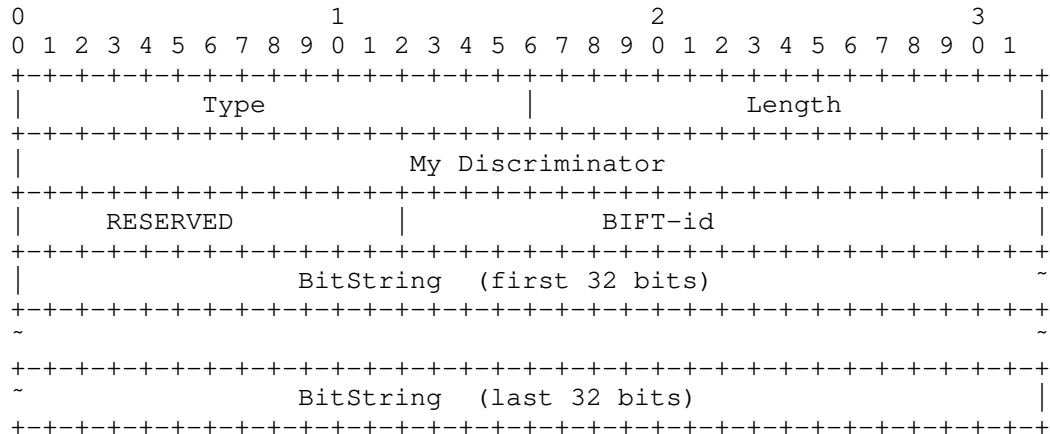


Figure 2: BIER BFD Sub-TLV for OSPF extension

Type : TBD4 by IANA.

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

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

5. Discriminators and Packet Demultiplexing

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. Active Tail in BIER BFD

[I-D.ietf-bfd-multipoint-active-tail] defined an extension for Multipoint BFD, which allows tails to notify the head of the lack of multipoint connectivity. For BIER BFD in active tail mode, the BFIR may learn the state and connectivity of the BFERs. As per [I-D.ietf-bfd-multipoint-active-tail], the BFIR can send the Poll sequence messages in combination with the unicast BFD over the monitored BFERs.

7. Security Considerations

For BIER OAM packet processing 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.

8. Acknowledgements

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

9. IANA Considerations

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

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

Table 1

10. References

10.1. Normative References

[I-D.ietf-bfd-multipoint]

Katz, D., Ward, D., Networks, J., and G. Mirsky, "BFD for Multipoint Networks", draft-ietf-bfd-multipoint-19 (work in progress), December 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-04 (work in progress), October 2018.

- [I-D.xiong-bier-resilience]
Xiong, Q., hu, f., and G. Mirsky, "The Resilience for BIER", draft-xiong-bier-resilience-01 (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>>.
- [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>>.

10.2. 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-10 (work in progress), November 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

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

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

Greg Mirsky
ZTE Corporation
USA

Email: gregimirsky@gmail.com

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

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

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

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

BIER WG
Internet-Draft
Intended status: Standards Track
Expires: April 2, 2019

Ran. Chen
Fangwei. Hu
Zheng. Zhang
Xianxia. Dai
ZTE Corporation
Mahesh. Sivakumar
Cisco Systems, Inc.
September 29, 2018

YANG Data Model for BIER Protocol
draft-ietf-bier-bier-yang-04.txt

Abstract

This document defines a YANG data model for BIER configuration and operation.

Status of This Memo

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

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <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 2, 2019.

Copyright Notice

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

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of

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

Table of Contents

1. Introduction	2
2. Design of the Data Model	2
3. Configuration	4
4. Control plane configuration	4
5. States	5
6. Notification	5
7. BIER YANG Data Model	5
8. Security Considerations	17
9. Acknowledgements	18
10. IANA Considerations	18
11. Normative references	18
Authors' Addresses	19

1. Introduction

This document defines a YANG data model for BIER configuration and operation.

2. Design of the Data Model

```

module: ietf-bier
augment /rt:routing:
  +--rw bier
    |
    |   +--rw bier-global
    |   |
    |   |   +--rw encapsulation-type?          identityref
    |   |   +--rw bitstringlength?            bsl
    |   |   +--rw bfr-id?                      bfr-id
    |   |   +--rw ipv4-bfr-prefix?             inet:ipv4-prefix
    |   |   +--rw ipv6-bfr-prefix?             inet:ipv6-prefix
    |   |   +--rw sub-domain* [sub-domain-id]
    |   |   |
    |   |   |   +--rw sub-domain-id            sub-domain-id
    |   |   |   +--rw underlay-protocol-type?  underlay-protocol-type
    |   |   |   +--rw mt-id?                   mt-id
    |   |   |   +--rw bfr-id?                   bfr-id
    |   |   |   +--rw bitstringlength?         bsl
    |   |   |   +--rw igp-algorithm?           ipa
    |   |   |   +--rw bier-algorithm?          bar
    |   |   +--rw af
    |   |   |
    |   |   |   +--rw ipv4* [bitstringlength bier-mpls-label-base]
    |   |   |   |
    |   |   |   |   +--rw bitstringlength      uint16
    |   |   |   |   +--rw bier-mpls-label-base  rt-types:mpls-label
    |   |   |   |   +--rw max-si?              max-si
    |   |   +--rw ipv6* [bitstringlength bier-mpls-label-base]

```

```

|          +---rw bitstringlength?      uint16
|          +---rw bier-mpls-label-base   rt-types:mpls-label
|          +---rw max-si?                max-si
+---ro bier-state
+---bier-global-state
|   +---ro encapsulation-type?   identityref
|   +---ro bitstringlength?      bsl
|   +---ro bfr-id?              bfr-id
|   +---ro ipv4-bfr-prefix?      inet:ipv4-prefix
|   +---ro ipv6-bfr-prefix?      inet:ipv6-prefix
|   +---ro sub-domain* [sub-domain-id]
|   +---ro sub-domain-id        sub-domain-id
|   +---ro underlay-protocol-type underlay-protocol-type
|   +---ro mt-id?               mt-id
|   +---ro bfr-id?              bfr-id
|   +---rw bitstringlength?      bsl
|   |   +---rw igp-algorithm?     ipa
|   +---rw bier-algorithm?       bar
|   +---ro ipv4* [bitstringlength bier-mpls-label-base]
|   |   +---ro bitstringlength    uint16
|   |   +---ro bier-mpls-label-base rt-types:mpls-label
|   |   +---ro max-si?            max-si
|   +---ro ipv6* [bitstringlength bier-mpls-label-base]
|   |   +---ro bitstringlength    uint16
|   |   +---ro bier-mpls-label-base rt-types:mpls-label
|   |   +---ro max-si?            max-si
+---ro birts-state
+---ro birt* [sub-domain-id]
|   +---ro sub-domain-id        sub-domain-id
|   +---ro birt-bitstringlength* [bitstringlength]
|   |   +---ro bitstringlength    uint16
|   +---ro birt-si* [si]
|   |   +---ro si                si
|   |   +---ro f-bm?             uint16
|   +---ro bier-mpls-in-label?   rt-types:mpls-label
|   +---ro bfr-nbr?              inet:ip-address
|   +---ro bier-mpls-out-label?  rt-types:mpls-label

augment /rt:routing/rt:control-plane-protocols/rt:control-plane-protocol/ospf:
ospf:
+---rw bier-ospf-cfg
|   +---rw mt-id                mt-id
|   +---rw bier-global
|   |   +---rw enable?          boolean
|   |   +---rw advertise?       boolean
|   |   +---rw receive?         boolean

augment /rt:routing/rt:control-plane-protocols/rt:control-plane-protocol/isis:
isis:
+---rw bier-isis-cfg

```



```

+--rw mt-id          mt-id
+--rw bier-global
  +--rw enable?      boolean
  +--rw advertise?   boolean
  +--rw receive?     boolean

```

notifications:

```

+---n bfr-id-collision
|   +--ro bfr-id?    bfr-id
+---n bfr-zero
|   +--ro ipv4-bfr-prefix?  inet:ipv4-prefix
|   +--ro ipv6-bfr-prefix?  inet:ipv6-prefix
+---n sub-domain-id-collision
  +--ro sub-domain-id?  sub-domain-id
  +--ro mt-id?          uint16

```

3. Configuration

This Module augments the `"/rt:routing:"` with a BIER container. This Container defines all the configuration parameters related to BIER for this particular routing.

The BIER configuration contains global configuration.

The global configuration includes BIER encapsulation type, imposition BitStringLengths, BFR-id, BFR-prefixes, and parameters associated with bier sub-domain.

In this document, we contains two types of BitStringLengths: Imposition and Disposition BitStringLengths, as defined in ([I-D.ietf-bier-architecture]). The imposition BitStringLengths is defined under bier-global container, and the disposition BitStringLengths is defined under the sub-domain.

4. Control plane configuration

This Module augments the `"/rt:routing/rt:routing-protocols/rt:routing-protocol/ospf:ospf:"` and `"/rt:routing/rt:routing-protocols/rt:routing-protocol/isis:isis:"` configuration with BIER.

This Module supports ISIS ([I-D.ietf-bier-isis-extensions]) and OSPF ([I-D.ietf-bier-ospf-bier-extensions]) as control plane for BIER.

5. States

The operational states contains basic parameters associated with bier, such as BIER encapsulation type, BitStringLengths, BFR-id, BFR-prefixes, and parameters associated with bier sub-domain.

It also includes the Bit Index Routing Table(BIRT).

6. Notification

This Module includes bfr-id-collision, bfr-zero, and sub-domain-id-collision.

7. BIER YANG Data Model

```
<CODE BEGINS> file "ietf-bier@2018-09-29.yang"
module ietf-bier {

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

    prefix "bier";

    import ietf-routing {
        prefix "rt";
    }

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

    import ietf-routing-types {
        prefix "rt-types";
    }

    import ietf-isis{
        prefix "isis";
    }

    import ietf-ospf {
        prefix "ospf";
    }

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

    contact
        "WG List: <mailto:bier@ietf.org>
```

WG Chair: Tony Przygienda
<mailto:tonysietf@gmail.com>

WG Chair: Greg Shepherd
<mailto:gjshep@gmail.com>

Editor: Ran Chen
<mailto:chen.ran@zte.com.cn>

Editor: Fangwei Hu
<mailto:hu.fangwei@zte.com.cn>

Editor: Zheng Zhang
<mailto:zhang.zheng@zte.com.cn>

Editor: Xianxian Dai
<mailto:dai.xianxian@zte.com.cn>

Editor: Mahesh Sivakumar
<mailto:masivaku@cisco.com>

";

description

"The YANG module defines a generic configuration
model for BIER.";

revision 2018-09-29{

description

"latest revision";

reference "RFC XXXX: YANG Data Model for BIER Protocol.";

}

revision 2018-02-07{

description

"03 revision";

reference "RFC XXXX: YANG Data Model for BIER Protocol.";

}

revision 2017-08-10{

description

"02 revision";

reference "RFC XXXX: YANG Data Model for BIER Protocol.";

}

revision 2017-01-20{

description

"01 revision";

reference "RFC XXXX: YANG Data Model for BIER Protocol.";

}

```
        revision 2016-07-23{
        description
            "00 revision";
        reference "RFC XXXX: YANG Data Model for BIER Protocol.";
    }

        revision 2016-05-12{
        description
            "04 revision";
        reference "RFC XXXX: YANG Data Model for BIER Protocol.";
    }

    revision 2016-03-16 {
        description
            "03 revision";
        reference "RFC XXXX: YANG Data Model for BIER Protocol.";
    }

    revision 2015-12-03 {
        description
            "02 revision";
        reference "RFC XXXX: YANG Data Model for BIER Protocol.";
    }

    revision 2015-10-16 {
        description
            "01 revision.";
        reference "RFC XXXX: YANG Data Model for BIER Protocol.";
    }

    revision 2015-06-22 {
        description
            "Initial revision.";
        reference "RFC XXXX: YANG Data Model for BIER Protocol.";
    }

    /* Identities */
    identity bier-encapsulation{
        description
            "Base identity for BIER encapsulation.";
    }
    identity bier-encapsulation-mpls {
        base bier-encapsulation;
        description
            "This identity represents MPLS encapsulation for bier.";
    }
    identity bier-encapsulation-ipv6 {
        base bier-encapsulation;
```

```
    description
      "This identity represents ipv6 encapsulation for bier.";
  }
  identity bier-encapsulation-ethernet {
    base bier-encapsulation;
    description
      "This identity represents ethernet encapsulation for bier.";
  }

  /*Typedefs*/

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

  typedef si {
    type uint16;
    description
      "The type for set identifier";
  }

  typedef bfr-id {
    type uint16;
    description
      "The type for bfr identifier";
  }

  typedef ipa{
    type uint8;
    description
      "The type for igp algorithm";
  }
  typedef bar{
    type uint8;
    description
      "The type for bier algorithm";
  }
  typedef mt-id {
    type uint16;
    description
      "The type for multi-topology identifier";
  }

  typedef max-si{
    type uint8;
    description
      "Maximum Set Identifier .";
```

```
    }

typedef bsl {
    type enumeration{
        enum 64-bit{
            value 1;
            description
                "bitstringlength is 64";
        }
        enum 128-bit{
            value 2;
            description
                "bitstringlength is 128";
        }
        enum 256-bit{
            value 3;
            description
                "bitstringlength is 256";
        }
        enum 512-bit{
            value 4;
            description
                "bitstringlength is 512";
        }
        enum 1024-bit{
            value 5;
            description
                "bitstringlength is 1024";
        }
        enum 2048-bit{
            value 6;
            description
                "bitstringlength is 2048";
        }
        enum 4096-bit{
            value 7;
            description
                "bitstringlength is 4096";
        }
    }
    description
        "The bitstringlength type for imposition mode";
}

typedef underlay-protocol-type {
    type enumeration{
        enum ISIS{
            value 1;

```

```

        description
            "isis protocol";
        }
    enum OSPF{
        value 2;
        description
            "ospf protocol";
        }
    enum BGP{
        value 3;
        description
            "bgp protocol";
        }
    }
    description
        "the underlay protocol type";
    }

/*grouping*/
    grouping bier-protocol-extensions{
        leaf mt-id{
            type mt-id;
            description
                "Multi-topology associated with bier sub-domain.";
        }
        container bier-global {
            leaf enable {
                type boolean;
                default false;
                description
                    "Enables bier protocol extensions.";
            }
            leaf advertise {
                type boolean;
                default true;
                description
                    "Enable to advertise the parameters associated with bi
er.";
            }
            leaf receive {
                type boolean;
                default true;
                description
                    "Enable to receive the parameters associated with bier
.";
            }
            description
                "BIER global config.";
        }
    }
    description

```

```
        "Defines protocol extensions.";
    }

    grouping bier-parameters{
        leaf encapsulation-type {
            type identityref {
                base bier-encapsulation;
            }
            default "bier-encapsulation-mpls";
            description
                "Dataplane to be used.";
        }
        leaf bitstringlength{
            type bsl;
            description
                "imposition bitstringlength.";
        }
        leaf bfr-id {
            type bfr-id;
            description
                "BIER bfr identifier.";
        }
        leaf ipv4-bfr-prefix {
            type inet:ipv4-prefix;
            description
                "BIER IPv4 prefix.";
        }
        leaf ipv6-bfr-prefix {
            type inet:ipv6-prefix;
            description
                "BIER IPv6 prefix.";
        }
    }
    description
        " BIER parameters.";
}

grouping bier-mpls-cfg{
    leaf bitstringlength {
        type uint16;
        description
            "BIER bitstringlength.";
    }
    leaf bier-mpls-label-base{
        type rt-types:mpls-label;
        description
            "BIER label base.";
    }
    leaf max-si{
```



```

        type max-si;
        description
            "Maximum Set Identifier.";
    }
    description
        "Defines the necessary label ranges per bitstring length.";
}

augment "/rt:routing" {
    description
        "This augments routing-instance configuration with bier.";
    container bier{
        container bier-global {
            uses bier-parameters;
            list sub-domain{
            key "sub-domain-id";
                leaf sub-domain-id{
                    type sub-domain-id;
                    description
                        "sub-domain ID.";
                }
                leaf underlay-protocol-type {
                    type underlay-protocol-type;
                    description
                        " the underlay protocol type.";
                }
            }
            leaf mt-id {
                type mt-id;
                description
                    "multi-topology ID.";
            }
            leaf bfr-id{
                type bfr-id;
                description
                    "BIER bfr identifier.";
            }
        }

        leaf bitstringlength{
            type bsl;
            description
                "Disposition bitstringlength.";
        }

        leaf igp-algorithm{
            type ipa;
            description
                "IGP Algorithm, the values are from the
IGP Algorithm registry.";
        }
    }
}

```

```

        leaf bier-algorithm{
            type bar;
            description
                "bier Algorithm.Specifies a BIER-specifi
c algorithm used to calculate underlay paths to reach BFERs";
        }
        container af {
            list ipv4 {
                key "bitstringlength bier-mpls-label-base";
                uses bier-mpls-cfg;

                description
                    "Defines the necessary label ranges per
bitstring length in ipv4.";
            }
            list ipv6 {
                key "bitstringlength bier-mpls-label-base";
                uses bier-mpls-cfg;

                description
                    "Defines the necessary label ranges per bits
tring length in ipv6.";
            }
            description
                "Bier mapping entries.";
        }
        description
            "Denfines subdomain configuration";
    }
    description
        "BIER global config.";
    }
    description "BIER config.";
}
container bier-state{
    config false;
    description
        "BIER operational state.";
    container bier-global-state{
        config false;
        uses bier-parameters;
    list sub-domain{
        key "sub-domain-id";
        leaf sub-domain-id{
            type sub-domain-id;
            description
                "sub-domain ID.";
        }
        leaf underlay-protocol-type {
            type underlay-protocol-type;
            description

```

```

        "the underlay protocol type.";
    }
    leaf mt-id {
        type mt-id;
        description
            "multi-topology ID.";
    }
    leaf bfr-id{
        type bfr-id;
        description
            "BIER bfr identifier.";
    }
    leaf bitstringlength{
        type bsl;
        description
            "Disposition bitstringlength.";
    }
    leaf igp-algorithm{
        type ipa;
        description
            "IGP Algorithm, the values are from the
IGP Algorithm registry.";
    }
    leaf bier-algorithm{
        type bar;
        description
            "bier Algorithm.Specifies a BIER-specifi
c algorithm used to calculate underlay paths to reach BFERs";
    }
    list ipv4 {
        key "bitstringlength bier-mpls-label-base";
        uses bier-mpls-cfg;
        description
            "Show the necessary label ranges per bitstring
length in ipv4.";
    }
    list ipv6 {
        key "bitstringlength bier-mpls-label-base";
        uses bier-mpls-cfg;
        description
            "Show the necessary label ranges per bit
string length in ipv6.";
    }
    description
        "Denfines subdomain configuration";
    }
    description
        "Parameters associated with bier.";
    }

    container birts-state{

```

```

list birt{
  key "sub-domain-id";
  leaf sub-domain-id{
    type sub-domain-id;
    description
      "BIER sub domain ID";
  }
  list birt-bitstringlength {
    key "bitstringlength";
    leaf bitstringlength{
      type uint16;
      description
        "BIER bitstringlength.";
    }
  }
  list birt-si {
    key "si";
    leaf si{
      type si;
      description
        "BIER set identifier.";
    }
    leaf f-bm{
      type uint16;
      description
        "BIER Forwarding Bit Mask.";
    }
    leaf bier-mpls-in-label{
      type rt-types:mpls-label;
      description
        "BIER in-label.";
    }
    leaf bfr-nbr{
      type inet:ip-address;
      description
        "BIER BFR Neighbors.";
    }
    leaf bier-mpls-out-label{
      type rt-types:mpls-label;
      description
        "BIER out-label.";
    }
  }
  description
    "Query the BIRT based on the key set identifier
    & bitstringlength & sub-domain-id.";
}
description
  "Query the BIRT based on the key bitstringlength & sub-
  domain-id.";
}
description

```

```
        "Query the BIRT based on the key sub-domain.";
    }
    description
        "Shows Bit Index Routing Table.";
    }
}

augment "/rt:routing/rt:control-plane-protocols/"
+ "rt:control-plane-protocol/ospf:ospf" {
    when "../rt:type = 'ospf:ospfv2' or
        ../rt:type = 'ospf:ospfv3'" {
        description
            "This augments the ospf routing protocol when used";
    }
    description
        "This augments ospf protocol configuration with bier.";
        container bier-ospf-cfg{
            uses bier-protocol-extensions;
            description
                "Control of bier advertisement and reception.";
        }
    }

augment "/rt:routing/rt:control-plane-protocols/"
+ "rt:control-plane-protocol/isis:isis"{
    when "/rt:routing/rt:control-plane-protocols/" +
        "rt:control-plane-protocol/rt:type = 'isis:isis'" {
        description
            "This augment ISIS routing protocol when used";
    }
    description
        "This augments ISIS protocol configuration with bier.";
        container bier-isis-cfg{
            uses bier-protocol-extensions;
            description
                "Control of bier advertisement and reception.";
        }
    }
}

/* Notifications */
notification bfr-id-collision{
    leaf bfr-id{
        type bfr-id;
    }
}
```

```

        description
            "BIER BFR ID.";
    }
    description
        "BFR ID received in the controlplane that caused BFR ID collision.";
    }

    notification bfr-zero{
        leaf ipv4-bfr-prefix{
            type inet:ipv4-prefix;
            description
                "BIER ipv4 bfr prefix";
        }

        leaf ipv6-bfr-prefix{
            type inet:ipv6-prefix;
            description
                "BIER ipv6 bfr prefix";
        }

        description
            "Invalid value associated with prefix";
    }

    notification sub-domain-id-collision{
        leaf sub-domain-id{
            type sub-domain-id;
            description
                "BIER sub domain ID";
        }

        leaf mt-id{
            type uint16;
            description
                "Multi-topology ID";
        }

        description
            "Sub domain ID received in the controlplane that caused Sub domain ID collision";
    }
}

```

<CODE ENDS>

8. Security Considerations

TBD.

9. Acknowledgements

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

10. IANA Considerations

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

11. Normative references

[I-D.ietf-bier-architecture]

Wijnands, I., Rosen, E., Dolganow, A., Przygienda, T., and S. Aldrin, "Multicast using Bit Index Explicit Replication", draft-ietf-bier-architecture-08 (work in progress), September 2017.

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

Ginsberg, L., Przygienda, T., Aldrin, S., and Z. Zhang, "BIER support via ISIS", draft-ietf-bier-isis-extensions-11 (work in progress), March 2018.

[I-D.ietf-bier-mpls-encapsulation]

Wijnands, I., Rosen, E., Dolganow, A., Tantsura, J., Aldrin, S., and I. Meilik, "Encapsulation for Bit Index Explicit Replication in MPLS and non-MPLS Networks", draft-ietf-bier-mpls-encapsulation-12 (work in progress), October 2017.

[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.

[I-D.ietf-isis-yang-isis-cfg]

Litkowski, S., Yeung, D., Lindem, A., Zhang, Z., and L. Lhotka, "YANG Data Model for IS-IS protocol", draft-ietf-isis-yang-isis-cfg-24 (work in progress), August 2018.

[I-D.ietf-mpls-base-yang]

Saad, T., Raza, K., Gandhi, R., Liu, X., and V. Beeram, "A YANG Data Model for MPLS Base", draft-ietf-mpls-base-yang-06 (work in progress), February 2018.

- [I-D.ietf-mpls-static-yang]
Saad, T., Raza, K., Gandhi, R., Liu, X., and V. Beeram, "A YANG Data Model for MPLS Static LSPs", draft-ietf-mpls-static-yang-05 (work in progress), February 2018.
- [I-D.ietf-netmod-routing-cfg]
Lhotka, L. and A. Lindem, "A YANG Data Model for Routing Management", draft-ietf-netmod-routing-cfg-25 (work in progress), November 2016.
- [I-D.ietf-ospf-yang]
Yeung, D., Qu, Y., Zhang, Z., Chen, I., and A. Lindem, "YANG Data Model for OSPF Protocol", draft-ietf-ospf-yang-17 (work in progress), September 2018.
- [RFC6020] Bjorklund, M., Ed., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, DOI 10.17487/RFC6020, October 2010, <<https://www.rfc-editor.org/info/rfc6020>>.
- [RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011, <<https://www.rfc-editor.org/info/rfc6241>>.
- [RFC6991] Schoenwaelder, J., Ed., "Common YANG Data Types", RFC 6991, DOI 10.17487/RFC6991, July 2013, <<https://www.rfc-editor.org/info/rfc6991>>.
- [RFC7223] Bjorklund, M., "A YANG Data Model for Interface Management", RFC 7223, DOI 10.17487/RFC7223, May 2014, <<https://www.rfc-editor.org/info/rfc7223>>.

Authors' Addresses

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

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

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

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

Zheng Zhang
ZTE Corporation
No.50 Software Avenue,Yuhuatai District
Nanjing, Jiangsu Province 210012
China

Email: zhang.zheng@zte.com.cn

Xianxian Dai
ZTE Corporation
No.50 Software Avenue,Yuhuatai District
Nanjing, Jiangsu Province 210012
China

Email: Dai.xianxian@zte.com.cn

Mahesh Sivakumar
Cisco Systems, Inc.
510 McCarthy Blvd
Milpitas,California 95035
United States

Email: masivaku@cisco.com

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: 9 July 2022

P. Pfister
I.J. Wijnands
S. Venaas
Cisco Systems
C. Wang

Z. Zhang
ZTE Corporation
M. Stenberg
5 January 2022

BIER Ingress Multicast Flow Overlay using Multicast Listener Discovery
Protocols
draft-ietf-bier-mld-06

Abstract

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

Status of This Memo

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

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

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

This Internet-Draft will expire on 9 July 2022.

Copyright Notice

Copyright (c) 2022 IETF Trust and the persons identified as the document authors. All rights reserved.

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

Table of Contents

1. Introduction	2
2. Terminology	3
3. Overview	4
4. Applicability Statement	4
5. Querier and Listener Specifications	5
5.1. Configuration Parameters	5
5.2. MLDv2 instances.	6
5.2.1. Sending Queries	6
5.2.2. Sending Reports	7
5.2.3. Receiving Queries	8
5.2.4. Receiving Reports	8
5.3. Packet Forwarding	8
6. BIER MLD/IGMP Extension Type	9
7. Security Considerations	10
8. IANA Considerations	10
9. Acknowledgements	11
10. References	11
10.1. Normative References	11
10.2. Informative References	11
Appendix A. BIER Use Case in Data Centers	13
A.1. Convention and Terminology	14
A.2. BIER in data centers	15
A.3. A BIER MLD solution for Virtual Network information	15
Authors' Addresses	16

1. Introduction

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

ingress routers.

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

This document defines an MLDv2 and IGMPv3 extension type, using the extension scheme defined in [I-D.ietf-pim-igmp-mld-extension], that is used to provide BIER specific information about the message originator.

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

2. Terminology

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

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

Additionally, the following definitions are used:

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

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

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

3. Overview

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

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

4. Applicability Statement

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

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

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

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

If multiple BFIRs have connectivity to the same source, a mechanism is needed to determine which BFIR should be the forwarder, that is not specified in this document. As a special case, if BIER is used end-to-end such that sources would be directly connected to the BFIRs, then an election mechanism is needed if there are multiple BFIRs on the same link as the source. One option is to utilize PIM DR Election where the DR is the BIER forwarder, but other election mechanisms could be used. In order to allow quick failover, the BFIRs that are not forwarders should still track BFER interest so that they have the correct state in case they become forwarders.

5. Querier and Listener Specifications

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

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

5.1. Configuration Parameters

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

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

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

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

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

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

5.2. MLDv2 instances.

BMLD Queriers MUST run a MLDv2 Querier instance with per-host tracking, which means they keep track of the MLDv2 state associated with each BMLD Listener. For that purpose, Listeners are identified by their respective BFR-Prefix, used as IP source address in all BMLD reports.

BMLD Listeners MUST run a MLDv2 Listener instance expressing their interest in the multicast traffic they are supposed to receive for local use or forwarding.

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

5.2.1. Sending Queries

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

- * Toward all BMLD Nodes (i.e., providing to the BIER Layer the BFR-ids of all BMLD Nodes).
- * Without the IPv6 router alert option [RFC2711] in the hop-by-hop extension header [RFC8200] (or the IPv4 router alert option [RFC2113] for IPv4).
- * With the IP destination address set to the 'all BMLD Nodes' group address.
- * With a deterministic IP source address. It is RECOMMENDED that the address is a BFR-Prefix of the sender, but it MAY be another value. This address is only used for querier election.

- * With a TTL value large enough such that the packet can be received by all BMLD Nodes, depending on the underlying BIER layer (whether it decrements the IP TTL or not) and the size of the network. The default value is 64.
- * The extension type defined in Section 6 MUST be included once, specifying the Sub-domain-id, BFR-id and BFR-Prefix of the sender. This information may be useful for logging and debugging.

5.2.2. Sending Reports

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

- * Toward all BMLD Queriers (i.e., providing to the BIER layer the BFR-ids of all BMLD Queriers).
- * Without the IPv6 router alert option [RFC2711] in the hop-by-hop extension header [RFC8200] (or the IPv4 router alert option [RFC2113] for IPv4).
- * With the IP destination address set to the 'all BMLD Queriers' group address.
- * With a deterministic IP source address. It is RECOMMENDED that the address is a BFR-Prefix of the sender.
- * With a TTL value large enough such that the packet can be received by all BMLD Queriers, depending on the underlying BIER layer (whether it decrements the IP TTL or not) and the size of the network. The default value is 64.
- * The extension type defined in Section 6 MUST be included once, specifying the Sub-domain-id, BFR-id and BFR-Prefix of the sender. This information is used to create the necessary forwarding state for requested flows, and may be useful for logging and debugging.

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

5.2.3. Receiving Queries

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

If the IPv6 (resp. IPv4) packet contains an ICMPv6 (resp. IGMP) message of type 'Multicast Listener Query' (resp. of type 'Membership Query'), and include the extension defined in Section 6), it is processed by the MLDv2 (resp. IGMPv3) instance run by the BMLD Querier. It MUST be dropped otherwise.

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

5.2.4. Receiving Reports

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

If the IPv6 (resp. IPv4) packet contains an ICMPv6 (resp. IGMP) message of type 'Multicast Listener Report Message v2' (resp. 'Version 3 Membership Report'), and include the extension defined in Section 6), it is processed by the MLDv2 (resp. IGMPv3) instance run by the BMLD Querier. It MUST be dropped otherwise.

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

5.3. Packet Forwarding

BMLD Queriers configure the BIER Layer using the information obtained using BMLD, and the extension Section 6), to track membership state, including the Sub-domain-id, BFR-id and BFR-Prefix of the members.

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

6. BIER MLD/IGMP Extension Type

A new MLD/IGMP extension type adds BIER specific information to IGMP/MLD messages, using the extension scheme defined in [I-D.ietf-pim-igmp-mld-extension]). The BIER specific information is the same as the PTA tunnel identifier in [RFC8556] and is shown in Figure 1. Note that, as defined in the MLD (resp. IGMP), existing implementations are supposed to ignore this additional data.

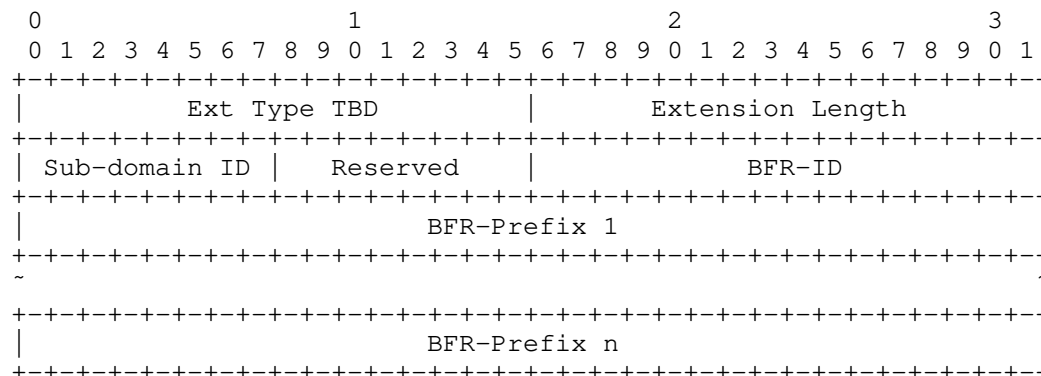


Figure 1: MLD/IGMP Extension Type for BIER

- * Ext Type: Assigned by IANA, identifying this BIER extension.
- * Extension Length: The length in octets of the data after this field. If there are n IPv4 prefixes, the length would be $4 + 4 * n$, if there are n IPv6 prefixes, the length would be $4 + 16 * n$.
- * Sub-domain-id: A single octet containing a BIER sub-domain-id (see [[RFC8279]]). This indicates the BIER sub-domain of the router originating the message.
- * Reserved: A single octet, MUST be set to 0 when sending and ignored when receiving.

- * BFR-id: A two-octet field containing the BFR-id, in the specified sub-domain, of the router originating the message.
- * BFR-prefix: The BFR-prefix (see [[RFC8279]]) of the router that is originating the message. The BFR-prefix will either be a /32 IPv4 address or a /128 IPv6 address.

This extension type MUST be present once in all IGMP and MLD messages when originated with a BIER header to identify the BIER originator. It is expected that any BIER router originating IGMP/MLD messages in BIER supports this specification. Any IGMP/MLD messages that do not contain the extension Section 6) MUST be dropped by the decapsulating router with no processing other than potentially logging or debugging. It is expected that any BIER router processing IGMP/MLD messages with BIER encapsulation supports this specification. If they do not, they will likely ignore the report since they cannot identify the BIER receiver, but they may be able to derive some of the receiver information from the BIER header.

7. Security Considerations

BMLD makes use of IGMPv3/MLDv2 messages transported over BIER in order to configure the BIER Layer of BFIRs. BMLD messages MUST be secured, either by relying on physical or link-layer security, by securing the IP packets (e.g., using IPsec [RFC4301]), or by relying on security features provided by the BIER Layer.

By spoofing the IP source address, an attacker could become the IGMP/MLD querier. Once one becomes the querier, several attack vectors are possible. This is similar to regular IGMP/MLD without BIER encapsulation.

An attacker could send reports with the BIER IGMP/MLD extension Section 6) specifying a BFR-ID and BIER prefix identifying another router. This would allow the attacker to:

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

8. IANA Considerations

This document requests that IANA assigns a new type called BIER information in the registry defined in [I-D.ietf-pim-igmp-mld-extension].

9. Acknowledgements

Comments concerning this document are very welcome.

10. References

10.1. Normative References

- [RFC2113] Katz, D., "IP Router Alert Option", RFC 2113, DOI 10.17487/RFC2113, February 1997, <<https://www.rfc-editor.org/info/rfc2113>>.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC3376] Cain, B., Deering, S., Kouvelas, I., Fenner, B., and A. Thyagarajan, "Internet Group Management Protocol, Version 3", RFC 3376, DOI 10.17487/RFC3376, October 2002, <<https://www.rfc-editor.org/info/rfc3376>>.
- [RFC3810] Vida, R., Ed. and L. Costa, Ed., "Multicast Listener Discovery Version 2 (MLDv2) for IPv6", RFC 3810, DOI 10.17487/RFC3810, June 2004, <<https://www.rfc-editor.org/info/rfc3810>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.
- [RFC8279] Wijnands, IJ., Ed., Rosen, E., Ed., Dolganow, A., Przygienda, T., and S. Aldrin, "Multicast Using Bit Index Explicit Replication (BIER)", RFC 8279, DOI 10.17487/RFC8279, November 2017, <<https://www.rfc-editor.org/info/rfc8279>>.
- [I-D.ietf-pim-igmp-mld-extension] Sivakumar, M., Venaas, S., Zhang, Z., and H. Asaeda, "Internet Group Management Protocol version 3 (IGMPv3) and Multicast Listener Discovery version 2 (MLDv2) Message Extension", Work in Progress, Internet-Draft, draft-ietf-pim-igmp-mld-extension-05, 7 November 2021, <<https://www.ietf.org/archive/id/draft-ietf-pim-igmp-mld-extension-05.txt>>.

10.2. Informative References

- [RFC2711] Partridge, C. and A. Jackson, "IPv6 Router Alert Option", RFC 2711, DOI 10.17487/RFC2711, October 1999, <<https://www.rfc-editor.org/info/rfc2711>>.
- [RFC3306] Haberman, B. and D. Thaler, "Unicast-Prefix-based IPv6 Multicast Addresses", RFC 3306, DOI 10.17487/RFC3306, August 2002, <<https://www.rfc-editor.org/info/rfc3306>>.
- [RFC4301] Kent, S. and K. Seo, "Security Architecture for the Internet Protocol", RFC 4301, DOI 10.17487/RFC4301, December 2005, <<https://www.rfc-editor.org/info/rfc4301>>.
- [RFC5015] Handley, M., Kouvelas, I., Speakman, T., and L. Vicisano, "Bidirectional Protocol Independent Multicast (BIDIR-PIM)", RFC 5015, DOI 10.17487/RFC5015, October 2007, <<https://www.rfc-editor.org/info/rfc5015>>.
- [RFC7348] Mahalingam, M., Dutt, D., Duda, K., Agarwal, P., Kreeger, L., Sridhar, T., Bursell, M., and C. Wright, "Virtual eXtensible Local Area Network (VXLAN): A Framework for Overlaying Virtualized Layer 2 Networks over Layer 3 Networks", RFC 7348, DOI 10.17487/RFC7348, August 2014, <<https://www.rfc-editor.org/info/rfc7348>>.
- [RFC7365] Lasserre, M., Balus, F., Morin, T., Bitar, N., and Y. Rekhter, "Framework for Data Center (DC) Network Virtualization", RFC 7365, DOI 10.17487/RFC7365, October 2014, <<https://www.rfc-editor.org/info/rfc7365>>.
- [RFC7761] Fenner, B., Handley, M., Holbrook, H., Kouvelas, I., Parekh, R., Zhang, Z., and L. Zheng, "Protocol Independent Multicast - Sparse Mode (PIM-SM): Protocol Specification (Revised)", STD 83, RFC 7761, DOI 10.17487/RFC7761, March 2016, <<https://www.rfc-editor.org/info/rfc7761>>.
- [RFC8200] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", STD 86, RFC 8200, DOI 10.17487/RFC8200, July 2017, <<https://www.rfc-editor.org/info/rfc8200>>.
- [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>>.
- [I-D.ietf-bier-mtud]
Venaas, S., Wijnands, I., Ginsberg, L., and M. Sivakumar, "BIER MTU Discovery", Work in Progress, Internet-Draft,

draft-ietf-bier-mtud-00, 27 February 2019,
 <<https://www.ietf.org/archive/id/draft-ietf-bier-mtud-00.txt>>.

[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", Work in Progress, Internet-Draft, draft-ietf-bier-path-mtu-discovery-11, 4 October 2021, <<https://www.ietf.org/archive/id/draft-ietf-bier-path-mtu-discovery-11.txt>>.

Appendix A. BIER Use Case in Data Centers

In current data center virtualization, virtual eXtensible Local Area Network (VXLAN) [RFC7348] is a kind of network virtualization overlay technology which is overlaid between NVEs and is intended for multi-tenancy data center networks, whose reference architecture is illustrated as per Figure 2.

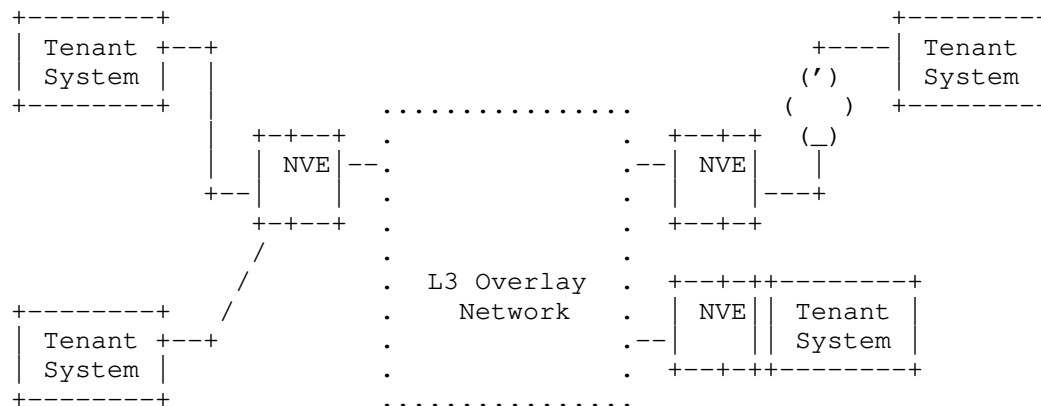


Figure 2: NVO3 Architecture

And there are two kinds of most common methods about how to forward BUM packets in this virtualization overlay network. One is using PIM as underlay multicast routing protocol to build explicit multicast distribution tree, such as PIM-SM [RFC7761] or PIM-BIDIR [RFC5015] multicast routing protocol. Then, when BUM packets arrive at NVE, it requires NVE to have a mapping between the VXLAN Network Identifier and the IP multicast group. According to the mapping, NVE can encapsulate BUM packets in a multicast packet which group address is the mapping IP multicast group address and steer them through explicit multicast distribution tree to the destination NVEs. This method has two serious drawbacks. It need the underlay network

supports complicated multicast routing protocol and maintains multicast related per-flow state in every transit nodes. What is more, how to configure the ratio of the mapping between VNI and IP multicast group is also an issue. If the ratio is 1:1, there should be 16M multicast groups in the underlay network at maximum to map to the 16M VNIs, which is really a significant challenge for the data center devices. If the ratio is n:1, it would result in inefficiency bandwidth utilization which is not optimal in data center networks.

The other method is using ingress replication to require each NVE to create a mapping between the VXLAN Network Identifier and the remote addresses of NVEs which belong to the same virtual network. When NVE receives BUM traffic from the attached tenant, NVE can encapsulate these BUM packets in unicast packets and replicate them and tunnel them to different remote NVEs respectively. Although this method can eliminate the burden of running multicast protocol in the underlay network, it has a significant disadvantage: large waste of bandwidth, especially in big-sized data center where there are many receivers.

BIER [RFC8279] is an architecture that provides optimal multicast forwarding through a "BIER domain" without requiring intermediate routers to maintain any multicast related per-flow state. BIER also does not require any explicit tree-building protocol for its operation. A multicast data packet enters a BIER domain at a "Bit-Forwarding Ingress Router" (BFIR), and leaves the BIER domain at one or more "Bit-Forwarding Egress Routers" (BFERs). The BFIR router adds a BIER header to the packet. The BIER header contains a bit-string in which each bit represents exactly one BFER to forward the packet to. The set of BFERs to which the multicast packet needs to be forwarded is expressed by setting the bits that correspond to those routers in the BIER header. Specifically, for BIER-TE, the BIER header may also contain a bit-string in which each bit indicates the link the flow passes through.

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

A.1. Convention and Terminology

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

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

VXLAN: Virtual eXtensible Local Area Network

VNI: VXLAN Network Identifier

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

A.2. BIER in data centers

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

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

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

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

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

Each NVE acquires the Virtual Network information, and advertises this Virtual Network information to other NVEs through the MLD messages. For a given VNI it sends BMLD reports to the BMLD nodes

address used for that VNI, for the group used for delivering BUM traffic for that VNI. This allows all NVE routers to know which other NVE routers have interest in BUM traffic for a particular VNI. If one attached virtual network is migrated, the NVE will withdraw the Virtual Network information by sending an unsolicited BMLD report. Note that NVEs also respond to periodic queries to BMLD Nodes addresses corresponding to VNIs for which they have interest.

When ingress NVE receives the Virtual Network information advertisement message, it builds a mapping between the receiving Virtual Network Context Identifier in this message and the bit-string in which each bit represents one egress NVE who sends the same Virtual Network information. Subsequently, once this ingress NVE receives some other MLD advertisements which include the same Virtual Network information from some other NVEs, it updates the bit-string in the mapping and adds the corresponding sending NVE to the updated bit-string. Once the ingress NVE removes one virtual network, it will delete the mapping corresponding to this virtual network as well as send withdraw message to other NVEs.

After finishing the above interaction of MLD messages, each ingress NVE knows where the other egress NVEs are in the same virtual network. When receiving BUM traffic from the attached virtual network, each ingress NVE knows exactly how to encapsulate this traffic and where to forward them to.

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

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

Authors' Addresses

Pierre Pfister
Cisco Systems
Paris
France

Email: pierre.pfister@darou.fr

IJsbrand Wijnands
Cisco Systems
De Kleetlaan 6a
1831 Diegem
Belgium

Email: ice@cisco.com

Stig Venaas
Cisco Systems
Tasman Drive
San Jose, CA 95134
United States of America

Email: stig@cisco.com

Cui(Linda) Wang

Email: lindawangjoy@gmail.com

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

Email: zhang.zheng@zte.com.cn

Markus Stenberg
FI-00930 Helsinki
Finland

Email: markus.stenberg@iki.fi

Network Working Group
Internet-Draft
Intended status: Experimental
Expires: August 18, 2019

S. Venaas
IJ. Wijnands
L. Ginsberg
Cisco Systems, Inc.
M. Sivakumar
Juniper Networks
February 14, 2019

BIER MTU Discovery
draft-ietf-bier-mtud-00

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 August 18, 2019.

Copyright Notice

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

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents

carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	2
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 [RFC8444]. 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.

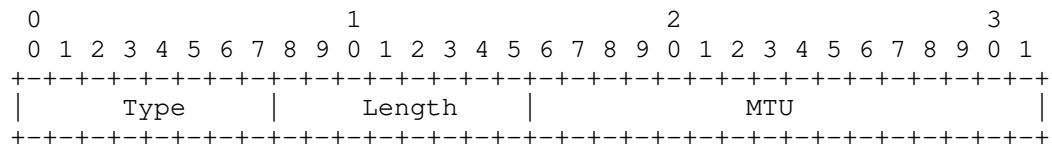
Note that a router should announce a new local MTU for a sub-domain immediately if the value becomes smaller than what it currently announces. This would happen if the MTU of an interface is configured to a smaller value, or the first BIER neighbor for a sub-domain is detected on an interface, and the 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.



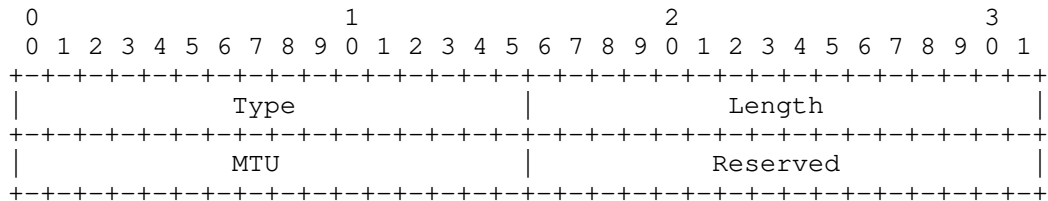
Type: TBD

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

- [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>>.
- [RFC8444] Psenak, P., Ed., Kumar, N., Wijnands, IJ., Dolganow, A., Przygienda, T., Zhang, J., and S. Aldrin, "OSPFv2 Extensions for Bit Index Explicit Replication (BIER)", RFC 8444, DOI 10.17487/RFC8444, November 2018, <<https://www.rfc-editor.org/info/rfc8444>>.

8.2. Informative References

- [I-D.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-05 (work in progress), December 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

BIER
Internet-Draft
Intended status: Standards Track
Expires: September 9, 2019

M. McBride
J. Xie
S. Dhanaraj
Huawei
March 8, 2019

Problem Statement of BIER IPv6 Encapsulation
draft-mcbride-bier-ipv6-problem-statement-01

Abstract

The BIER WG has a charter item to work on mechanisms which use BIER natively in IPv6. This document is intended to help the WG with this effort by describing the problem space of transporting packets, with Bit Index Explicit Replication (BIER) headers, in an IPv6 environment. There will be a need to send IPv6 payloads, to multiple IPv6 destinations, using BIER. There have been several proposed solutions in this area. But there hasn't been a document which describes the problem and why this may be necessary. The goal of this document is to describe the BIER IPv6 problem space, basic use cases, why new solutions may be needed and briefly summarize some of the proposed solutions.

Status of This Memo

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

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

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

This Internet-Draft will expire on September 9, 2019.

Copyright Notice

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

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents

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

Table of Contents

1. Introduction	2
1.1. Requirements Language	3
1.2. Terminology	3
2. Problem Statement	3
3. BIER IPv6 encapsulation Scenario's	3
3.1. BIERv6 for Access Network	4
3.2. BIERv6 for Data Center	4
3.3. BIERv6 for Core Networks	4
3.4. Implications for BIER in SRv6	5
4. Example Proposed Solutions	5
4.1. BIER-ETH encapsulation in IPv6 networks	5
4.2. Encode Bitstring in IPv6 destination address	6
4.3. Add BIER header into IPv6 Extension Header	7
4.4. Transport BIER as IPv6 payload	8
4.5. Tunneling BIER in a IPv6 tunnel	8
5. Suggested Requirements	9
6. IANA Considerations	10
7. Security Considerations	10
8. Acknowledgement	10
9. Normative References	10
Authors' Addresses	11

1. Introduction

Bit Index Explicit Replication (BIER) [RFC8279] is an architecture that provides optimal multicast forwarding, without requiring intermediate routers to maintain per-flow state, through the use of a multicast-specific BIER header. [RFC8296] defines two types of BIER encapsulation to run on physical links: one is BIER MPLS encapsulation to run on various physical links that support MPLS, the other is BIER Ethernet encapsulation to run on ethernet links, with an ethertype 0xAB37. This document describes using BIER in non-MPLS IPv6 environments. We explain the problem space of transporting IPv4/IPv6 multicast payloads, from an IPv6 router (BFIR) to multicast IPv6 destinations (BFERs), using BIER. This can include native IPv6 encapsulation and generic tunneling. There have been several proposed solutions in this area. But there hasn't been a document which describes the problem and why this may be necessary. The goal

of this document is to describe the BIER v6 problem space, use cases, encapsulations, existing solutions and why new solutions may be needed. This draft is intended to help the BIER WG evaluate the need for an encapsulation that is IPv6-specific through describing the problem and summarizing BIERV6 related solutions.

1.1. Requirements Language

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

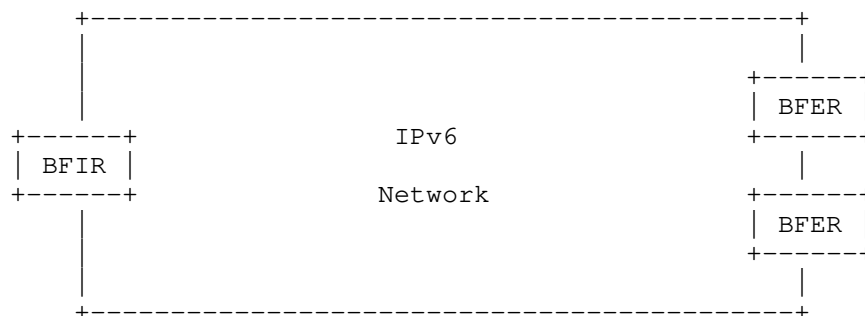
1.2. Terminology

- o BIER: Bit Index Explicit Replication. Provides optimal multicast forwarding through adding a BIER header and removing state in intermediate routers.
- o BUM: Broadcast, Unknown Unicast, Multicast. Term used to describe the three types of Ethernet modes that will be forwarded to multiple destinations

2. Problem Statement

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

3. BIER IPv6 encapsulation Scenario's



This basic scenario depicts the need to replicate bier packets from a BFIR to BFERs across an IPv6 core. The IPv6 environment may include

a variety of link types, may be entirely IPv6, may be dual stack or any type of combination which includes IPv6. Regardless of the environment, there are times when a BIER header, including the BIER bitstring used to determine the set of BIER forwarding egress routers, will need to traverse a IPv6 domain. The ways in which BIER will function in an IPv6 environment is the problem that needs to be solved. [RFC8354] lists some good IPv6 related use cases which we will similarly reference in this document.

3.1. BIERv6 for Access Network

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

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

3.2. BIERv6 for Data Center

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

3.3. BIERv6 for Core Networks

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

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

3.4. Implications for BIER in SRv6

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

4. Example Proposed Solutions

Although this is not a solutions document it should be helpful to list the various proposed solutions, without addressing the benefits of one over another, to help understand the problem more clearly. The following are solutions that have been proposed to solve BIER in v6 environments.

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

4.1. BIER-ETH encapsulation in IPv6 networks

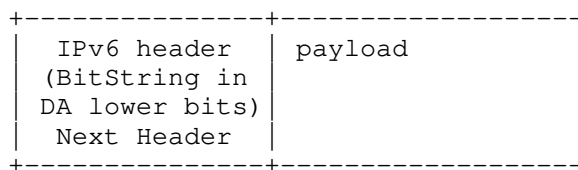
Ethernet (ethType = 0xAB37)	BIER header (BIFT-id, ...) Next Header	payload
-----------------------------------	--	---------

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

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

- o BIER-ETH in IPv6 networks is considered similar to 6PE solution where-in the multicast user data packet is encapsulated with-in the BIER-MPLS header.
- * It is worth noting that the only major difference between BIER-MPLS and BIER-Non-MPLS header is that BIER-MPLS uses downstream assigned MPLS label while BIER-Non-MPLS header uses a domain-wide-unique BIFT-id. While the use of domain-wide-unique BIFT-id in BIER-ETH header takes away the complexity of allocation and state maintenance from the network, it still requires some sort of ID (similar to label) to identify the application context after the decapsulation of BIER header (example: MVPN VRF Label). Encoding of such an ID/LABEL before encapsulating the multicast user data payload with BIER-ETH header cannot be avoided.
- * The absence of an IPv6 header, and the optional IPv6 extension headers, deprives BIER of some of the useful cases (ex: Use of IPv6 address for identification of network function or service mapping) that is otherwise possible in native IPv6 encapsulation which utilizes a IPv6 header.
- * Tunneling of BIER packets is one common technique used for FRR, to tunnel over BIER incapable nodes etc. While it is possible for the BIER-ETH encapsulated packet to be further encapsulated within a GRE6 or SRv6, etc tunnel, it might not be possible to parse and decapsulate different types of tunnel headers and forward the BIER packet completely in hardware fast path similar to the label stack processing in BIER-MPLS networks. It would be useful to select an encapsulation which could help in processing the tunnel and BIER header and make the forwarding decision completely in hardware fast path, which is lacking in BIER-ETH encapsulation if chosen to be deployed in pure IPv6 networks.

4.2. Encode Bitstring in IPv6 destination address



As described in [I-D.pfister-bier-over-ipv6], The information required by BIER is stored in the destination IPv6 address. The BIER BitString is encoded in the low-order bits of the IPv6 destination address of each packet. The high-order bits of the IPv6 destination

address are used by intermediate routers for unicast forwarding, deciding whether a packet is a BIER packet, and if so, to identify the BIER Sub-Domain, Set Identifier and BitString length. No additional extension or encapsulation header is required. Instead of encapsulating the packet in IPv6, the payload is attached to the BIER IPv6 header and the IPv6 protocol number is set to the type of the payload. If the payload is UDP, the UDP checksum needs to change when the BitString in the IPv6 destination address changes.

4.3. Add BIER header into IPv6 Extension Header

IPv6 header (Multicast DA)	IPv6 Ext header (BIER header in TLV Type = X)	payload
Next Header	Next Header	

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

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

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

BIER IPv6 encapsulation to include the BIER Header ([RFC8296]) as an IPv6 Extension Header, to pilot the hop-by-hop BIER replication.

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

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

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

4.4. Transport BIER as IPv6 payload

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

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

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

4.5. Tunneling BIER in a IPv6 tunnel

IPv6 header	IPv6 Ext header (optional)	GRE header	BIER Hdr + payload as GRE Payload
Next Header	Next Header	Proto = X	

A generic IPv6 Tunnel could be used to encapsulate the bier packet within an IPv6 domain.

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

IPv6 header	IPv6 Ext header (optional)	UDP header	BIER Hdr + payload as UDP Payload
Next Header	Next Header	DPort = X	

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

5. Suggested Requirements

This is not a requirements document and we may eventually remove this section. We will, however, summarize some of the "requirements", that have been suggested on the BIER email list, to help further understand the problem. At a minimum, this may serve as a kick start to a requirements draft if one is deemed necessary by the WG:

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

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

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

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

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

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

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

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

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

6. IANA Considerations

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

7. Security Considerations

There are no security issues introduced by this draft.

8. Acknowledgement

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

9. Normative References

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

Pfister, P. and I. Wijnands, "An IPv6 based BIER Encapsulation and Encoding", draft-pfister-bier-over-ipv6-01 (work in progress), October 2016.

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

Xie, J., Geng, L., McBride, M., Dhanaraj, S., Yan, G., and Y. Xia, "Encapsulation for BIER in Non-MPLS IPv6 Networks", draft-xie-bier-ipv6-encapsulation-00 (work in progress), March 2019.

[I-D.xu-bier-encapsulation]

Xu, X., somasundaram.s@alcatel-lucent.com, s., Jacquenet, C., Raszuk, R., and Z. Zhang, "A Transport-Independent Bit Index Explicit Replication (BIER) Encapsulation Header", draft-xu-bier-encapsulation-06 (work in progress), September 2016.

[I-D.zhang-bier-bierin6]

Zhang, Z. and T. Przygienda, "BIER in IPv6", draft-zhang-bier-bierin6-02 (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>>.
- [RFC2473] Conta, A. and S. Deering, "Generic Packet Tunneling in IPv6 Specification", RFC 2473, DOI 10.17487/RFC2473, December 1998, <<https://www.rfc-editor.org/info/rfc2473>>.
- [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>>.
- [RFC8354] Brzozowski, J., Leddy, J., Filsfils, C., Maglione, R., Ed., and M. Townsley, "Use Cases for IPv6 Source Packet Routing in Networking (SPRING)", RFC 8354, DOI 10.17487/RFC8354, March 2018, <<https://www.rfc-editor.org/info/rfc8354>>.

Authors' Addresses

Mike McBride
Huawei

Email: michael.mcbride@huawei.com

Jingrong Xie
Huawei

Email: xiejingrong@huawei.com

Senthil Dhanaraj
Huawei

Email: senthil.dhanaraj@huawei.com

BIER Working Group
Internet-Draft
Intended status: Standards Track
Expires: September 6, 2019

D. Merling
M. Menth
University of Tuebingen
March 05, 2019

BIER Fast Reroute
draft-merling-bier-frr-00

Abstract

BIER is a scalable multicast overlay [RFC8279] that utilizes some routing underlay, e.g., IP, to build up its Bit Index Forwarding Tables (BIFTs). This document proposes a Fast Reroute Extension for BIER (BIER-FRR). In case of a link or node failure, the routing underlay may first utilize FRR techniques to restore connectivity and then its forwarding tables converge. After that, BIER can update its BIFTs, which requires time. BIER packets may not be delivered until the last procedure has finished. With BIER-FRR, a BIER Forwarding Router (BFR) can deliver BIER packets again after a link or node failures as soon as the connectivity within the routing underlay is restored and the BFR is informed about a next-hop (NH) that is unreachable on a lower layer. BIER-FRR provides a mode for link protection and node protection. For link protection, it tunnels traffic to the next-hop using the underlying routing. For node protection, it forwards BIER packets to their specific next-hop and next-next-hops using tunnels in the underlying routing after applying a suitable backup bitmask to the bitstring in the BIER header of each packet. This procedure prevents duplicates. If topology allows, BIER-FRR achieves full protection against any single component failure. For link protection, BIER-FRR requires only a minor change to the forwarding logic. For node protection, BIER-FRR also requires backup entries in the BIFT.

This document describes the concept and operating principles of BIER-FRR. It defines the necessary modifications to the BIER forwarding Procedure and the BIFT, and explains how backup entries are computed.

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 September 6, 2019.

Copyright Notice

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

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

Table of Contents

1. Introduction	3
2. Terminology	4
2.1. Requirements Language	6
3. Problem Statement	6
3.1. Link Failures	6
3.1.1. BIER Encapsulation within a Lower-Layer Technology with Protection	6
3.1.2. BIER Encapsulation within the Routing Underlay . . .	6
3.1.3. BIER Encapsulation within a Lower-Layer Technology without Protection	7
3.2. Node Failures	7
4. Fast Reroute Extension for BIER (BIER-FRR)	7
4.1. BIER-FRR Link Protection	7
4.1.1. Mechanism	8
4.1.2. Example	8
4.2. BIER-FRR Node Protection	8
4.2.1. Mechanism	8
4.2.2. Example	10
4.2.3. Computation of Backup BIFT Entries	11
4.2.3.1. Computation	11
4.2.3.2. Example	12
4.3. Protection Level	12
5. Necessary Changes to the BIER Architecture	13
5.1. Unicast Tunneling	13

5.2. Detecting Unreachable N(N)Hs	13
5.3. BIFT with backup entries	13
6. Security Considerations	13
7. IANA Considerations	13
8. References	13
8.1. Normative References	14
8.2. Informative References	14
Authors' Addresses	14

1. Introduction

With BIER [RFC8279], Bit-Forwarding Routers (BFRs) forward packets based on a bitstring in the BIER header using the information in their Bit Index Forwarding Tables (BIFTs). In case of a persistent link or node failure, BIER traffic may not be delivered until the BIFT has been updated based on the re-converged routing underlay. The routing underlay restores connectivity more quickly than BIER, in particular if the routing underlay leverages fast reroute (FRR) mechanisms because then the forwarding ability is retained before forwarding tables of the routing underlay have converged. In this document we propose Fast Reroute Extension for BIER (BIER-FRR). It enables a BFR to quickly reroute BIER packets as soon as the underlying routing works again and it is informed about a next-hop (NH) that is unreachable on a lower layer.

We first explain the problem and distinguish link and node failures for that purpose. In case of a persistent link failure, a BFR cannot deliver BIER traffic until the NH in the BIFT is updated with an appropriate node. In case of a node failure, the entire multicast subtree behind the failed node is not reachable until the BIFT is updated. Thus, in either case, BIER's connectivity is restored only after the underlying routing has converged and the BIFTs have been updated. This may require substantial time during which BIER traffic is dropped. An exception are unreachable NHs to which BIER traffic is sent through tunnels in the routing underlay. They are reachable again as soon as the routing underlay works again.

BIER-FRR tackles this problems with two different modes that have different implementation complexity: link protection and node protection. In any case, a BFR with an unreachable NH needs to be informed about the failure and acts as a point of local repair (PLR). E.g., BFD mechanisms may be used [I-D.hu-bier-bfd] to detect failed NHs. With BIER-FRR link protection, a BFR tunnels affected BIER packets towards the NH using a tunnel in the underlying routing. Then, this traffic can be delivered as soon as the underlying routing works again. With BIER-FRR node protection, a BFR tunnels affected BIER packets to the NH and all relevant next-next-hops (NNHs) in the underlying routing after applying suitable backup bitmasks to the

bitstring in the BIER header. This procedure ensures that both the NH and all potential multicast subtrees receive the traffic if possible, and it prevents potential duplicates and loops on the BIER layer. Thus, BIER-FRR basically implements 1:1 protection. The latter is discussed in [I-D.xiong-bier-resilience] without proposing a specific mechanism.

This document describes the concept of BIER-FRR, protection properties, the computation of backup bitmasks, and gives a detailed BIER-FRR forwarding example.

2. Terminology

The following sections require the understanding of certain abbreviations and definitions that were defined within other documents, especially [RFC8279]. To facilitate the reading of this document, they are shortly explained in this section.

- o BFR: Bit-Forwarding Router, Section 1 of [RFC8279]

A device that acts as a BIER forwarding device in the BIER domain.

- o BFIR: Bit-Forwarding Ingress Router, Section 1 of [RFC8279]

Entry point to the BIER domain. A BFIR adds a BIER header to a multicast packet.

- o BFER: Bit-Forwarding Egress Router, Section 1 of [RFC8279]

Exit point of the BIER domain. A BFER removes the BIER header of a multicast packet.

- o NH: Next-hop

The next downstream BFR to which a packet is forwarded.

- o BIFT: Bit Index Forwarding Table, Section 6.4 of [RFC8279]

A BFR uses its BIFT to determine the NH(s) of a BIER packet. The BIFT maps a F-BM to each NH of a BFR.

- o F-BM: Forwarding bitmask Section 6.4 of [RFC8279]

A F-BM is a bitmask that indicates which destinations are reached via the subtree of the corresponding NH. A F-BM is applied to the BIER header by bitwise AND'ing the F-BM with the bitstring in the BIER header. This prevents duplicates at BFERs.

- o Routing underlay: Section 4.1 of [RFC8279]

The routing underlay connects pairs of BFR. If a typical Interior Gateway Protocol (IGP) like OSPF is used, the multicast packets will be forwarded on shortest paths. Other routing underlays with different path layouts are possible. The routing underlay is used to determine the NH entries of the BIFT.

- o BIER Layer: Section 4.2 of [RFC8279]

Conceptually the BIER layer is placed above the routing underlay. The BIER layer can be understood as a transport layer for multicast packets. It consists of all necessary protocols and mechanisms to forward a BIER packet from a BFIR, over potentially multiple BFRs, to a set of BFERs.

- o Multicast Overlay: Section 4.3 of [RFC8279]

Conceptually the multicast overlay is placed above the BIER layer. It is used to maintain information about egress points of multicast groups. The multicast overlay passes those information to the BIER layer which then determines the corresponding BIER headers.

- o PLR: Point of Local Repair

A node that cannot forward a packet due to an unreachable NH.

- o BIER-FRR: Fast Reroute Extension for BIER (BIER-FRR)

A mechanism to restore connectivity on the BIER layer as soon as BFRs are informed about non-reachable NHs and the underlying routing works again.

- o BIER-FRR link protection

A mode of BIER-FRR that can handle only link failures.

- o BIER-FRR node protection

A mode of BIER-FRR that can handle both link and node failures. It is more complex than BIER-FRR link protection.

- o NNH: Next-next-hop

Next downstream BFR after the NH.

2.1. 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].

3. Problem Statement

We first consider the impact of link failures and then the one of node failures on the behavior of a BIER network without BIER-FRR.

3.1. Link Failures

The effect of a link failure depends on the technology used for encapsulation of BIER packets. We distinguish three cases. (1) BIER packets are carried over some lower-layer technology with protection. (2) BIER packets are tunneled through the routing underlay. (3) BIER packets are carried over some lower-layer technology without protection.

3.1.1. BIER Encapsulation within a Lower-Layer Technology with Protection

MPLS is an example for a lower-layer technology with protection capabilities. In case of a link failure, first packets are lost, then the protection mechanism of the lower-layer technology quickly restores the link. From then on, packets are no longer lost.

3.1.2. BIER Encapsulation within the Routing Underlay

IP is an example for a routing underlay. The routing underlay is expected to deal with failures of lower-layer technologies. In case of a link failure, packets are lost. If the failure persists due to a lower-layer technology without protection, the routing underlay is informed about the failure. The routing underlay may leverage FRR techniques, e.g., Loop-Free Alternates (LFAs) [RFC5286], to quickly restore reachability so that packets are delivered again which are sent encapsulated the within routing underlay. From then on, also BIER packets encapsulated within the routing underlay are delivered again.

At the same time, routing reconvergence is triggered. When the routing has converged after some time, forwarding tables of the routing underlay are updated. Based on them, new NHs for BIFTs are computed and installed so that the PLR delivers BIER packets to a different NH than the one that is still unreachable via the lower-layer technology.

3.1.3. BIER Encapsulation within a Lower-Layer Technology without Protection

Ethernet is an example for a lower-layer technology without protection. In case of a link failure, the failure persists from the perspective of BIER and the routing underlay unless the failure is repaired. As a consequence, packets are lost. Then, the routing underlay acts as described above to restore reachability and finally updates its forwarding tables. By that time, BIER packets encapsulated within the lower-layer technology are still dropped. Then, new NHs for BIFTs are computed based on the new forwarding tables of the routing underlay and are installed. From then on, BIER can deliver packets again over a different NH.

When BIER-FRR is used, BIER packets can be delivered again as soon as the BFR is informed about the unreachable NH and routing underlay works again.

3.2. Node Failures

The effect of node failures is more severe. First, the packets cannot be delivered to the failed node. This, however, cannot be repaired. Second, multicast distribution trees downstream of a failed NH cannot receive traffic as the failed NH replicate traffic towards relevant NNHs. This problem is solved neither by lower-layer technologies with link protection nor by BIER encapsulation within the routing underlay. Therefore, BIER packets sent to the failed NH are dropped until BIFTs are updated based on reconverged forwarding tables of the routing underlay. This may require quite some time.

When BIER-FRR node protection is used, BIER packets can be delivered along the affected multicast tree as soon as the BFR is informed about the unreachable NH and the routing underlay works again.

4. Fast Reroute Extension for BIER (BIER-FRR)

This section describes the concept of BIER-FRR. In case of a link or node failure, it reroutes BIER packets until the BIFTs are updated or the failure is repaired. BIER-FRR offers two modes: link protection and node protection. Their protection level is summarized.

4.1. BIER-FRR Link Protection

We introduce the mechanism and illustrate it by an example.

4.1.1. Mechanism

When a BFR is informed about an unreachable NH, it tunnels all BIER packets towards that NH through the routing underlay. As soon as the routing underlay works again, the BIER packets are delivered to the NH if the NH still works. Then, the NH forwards the BIER packets further along the multicast distribution tree.

4.1.2. Example

Figure 1 shows an example topology for the routing underlay and Figure 2 the multicast distribution tree for BFR 1 on the BIER Layer which is computed based on shortest paths.

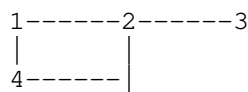


Figure 1: Example topology for the routing underlay.

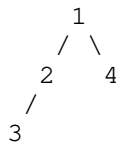


Figure 2: Multicast distribution tree for BFR 1 on the BIER Layer.

When BFR 1 sends a packet to BFR 3, the NH is BFR 2. If link 1<->2 fails, packets encapsulated within a lower-layer technology can no longer be delivered from BFR 1 to BFR 2. As soon as BFR 1 is informed that BFR 2 is no longer reachable, it encapsulates the BIER packets to BFR 3 within the routing underlay towards BFR 2. When the routing underlay has restored connectivity, the BIER packets are tunneled from BFR 1 via BFR 4 to BFR 2 which decapsulates them. Then BFR further forwards the BIER packets to BFR 3.

4.2. BIER-FRR Node Protection

We introduce the mechanism, illustrate it by an example, and explain how needed backup BIFT entries are computed.

4.2.1. Mechanism

When a BFR is informed about an unreachable NH, it tunnels all affected BIER packets to that NH if the NH receives a copy of the BIER packet, and to all NNHs over which copies of the BIER packet are

to be delivered. The latter are the relevant NNHs. Before tunneling the packets, their bitstrings are modified using backup F-BMs to avoid forwarding loops and duplicates.

The BIFT and its operation are explained in detail in Section 6.4 of [RFC8279]. We briefly revise them to facilitate further reading before introducing backup BIFT entries to support the solution presented above. Figure 3 illustrates the structure of the BIFT. The BIFT contains for each BFR-id a F-BM and the next hop (BFR-NBR). The BFR-id is mapped to a position in the bitstring of the BIER header; for this purpose, the bits within a bitstring are counted from right to left starting with 1. The F-BM is also a bitstring and indicates all BFRs that are reachable through the multicast distribution subtree via BFR-NBR. As a result, the F-BMs in a BIFT are either identical (same BFR-NBR) or disjoint with regard to activated bits. For transmission of BIER packets, the BIFT is used together with a copy of the bitstring of the BIER header. Processing is performed by the following loop. An entry from the BIFT is selected that holds a BFR-id which is set in the copy of the bitstring. Then, the F-BM of that entry is applied to the bitstring of the BIER packet and then the BIER packet is sent to the indicated BFR-NBR. The bits of the F-BM are cleared in the bitstring copy and the loop restarts. It ends when all bits in the bitstring copy are zero.

BFR-id	F-BM	BFR-NBR
1		

Figure 3: Structure of the BIFT according to [RFC8279].

BFR-id	F-BM	BFR-NBR
1	primary F-BM backup F-BM	primary NH backup NH
...

Figure 4: Structure of a BIFT with primary and backup entries.

To support BIER-FRR node protection, backup BIFT entries for protected BFR-NBRs are added to the BIFT. That structure is illustrated in Figure 4. We call the normal BIFT entries primary entries. Backup BIFT entries have the same structure as primary BIFT entries and are used for forwarding in the same way. The set of

active bits in a primary BIFT entry must equal the set of active bits in its corresponding backup entries to guarantee that all destinations in the multicast distribution subtree via BFR-NBR are protected.

If the BFR-NBR of a primary BIFT entry is reachable, the corresponding backup BIFT entries are ignored in the forwarding process. If the BFR-NBR of a primary BIFT entry is unreachable, the BIER packet is processed using the corresponding backup BIFT entries instead of the primary BIFT entry. BIER packets sent by a backup BIFT entry MUST be tunneled through the routing underlay to the backup BFR-NBR after application of the backup F-BM.

There are other options to organize the backup entries just as there are options for more scalable BIFT organization.

4.2.2. Example

Figure 5 shows an example topology for the routing underlay and Figure 6 the multicast distribution trees for BFR 1 and BFR 2 on the BIER Layer which are computed based on shortest paths.

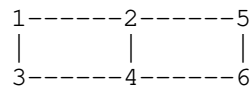


Figure 5: Example topology for the routing underlay.

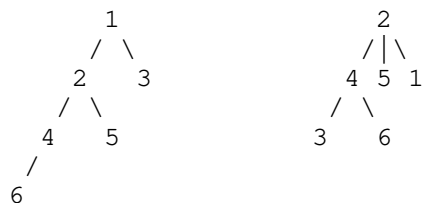


Figure 6: Multicast distribution trees for BFR 1 and BFR 2 on the BIER Layer.

Figure 7 gives the BIFT for BFR 1 with backup entries.

FRR-BIFT BFR 1		
BFR-id	F-BM	BFR-NBR
1	000001 -	- -
2	111010 000010	2 2
3	000100 000100	3 3
4	111010 101000	2 4
5	111010 010000	2 5
6	111010 101000	2 4

Figure 7: BIFT of BFR 1 with backup entries.

When BFR 1 sends a BIER packet to BFR 6, the NH is BFR 2. If link 1<->2 fails, BIER packets encapsulated within a lower-layer technology can no longer be delivered from BFR 1 to BFR 2. As soon as BFR 1 is informed that BFR 2 is no longer reachable, it applies backup BIFT entries to forward affected BIER packets. That means, it modifies the bitstring of BIER packets towards BFR 6 with the appropriate backup F-BM and sends them to backup NH BFR 4 after encapsulation within the routing underlay. Therefore, the packets are tunneled from BFR 1 via BFR 3 to BFR 4. BFR 4 decapsulates the packet and a copy of the BIER packet is delivered to BFR 6.

4.2.3. Computation of Backup BIFT Entries

We explain the computation and give an example.

4.2.3.1. Computation

BIER-FRR node protection ensures that a PLR can send BIER packets in case of an unreachable NH to all BFRs in the downstream multicast subtree of the unreachable NH. For this purpose, backup entries for these BFRs need to be provided in the BIFT of the PLR. We compute them differently for the NHs of PLRs and for all other BFRs which

belong to multicast subtrees starting with a NNH. This leads to two computation rules:

1. BIER packets for NHs are sent to the NHs (backup-NH = NH) via a tunnel and the backup F-BM must ensure that these BIER packets are not forwarded any further. That means, the backup F-BM contains only the BFR-id of the NH.
2. BIER packets for other BFRs are sent via a tunnel to the NNH in the multicast subtree they belong to. Also all other BFRs in the same multicast subtree should be reached with the same BIER packet. Therefore, the backup F-BM for a BFR contains the BFR-ids for all BFRs in its multicast subtree starting with the respective NNH. Thus, the corresponding backup F-BM can be computed by ANDing the PLR's F-BM for the NH and the NH's F-BM for the specific NNH.

4.2.3.2. Example

We consider the BIFT of BFR 1 in Figure 7.

Example for rule (1): The backup BIFT entry for BFR 2 has a F-BM that just contains BFR 2 (000010).

Example for rule (2): The backup BIFT entry for BFR 4 has a F-BM that contains BFR 4 and BFR 6 (101000). It is computed ANDing the F-BM of BFR 1 for BFR 2 (111010) and the F-BM of BFR 2 for BFR 4 (101100). The latter has been derived from the multicast distribution tree of BFR 2 in Figure 6.

4.3. Protection Level

BIER-FRR is a protection scheme that reacts when a NH is no longer reachable. It is a local mechanism that does not require signaling or cooperation with other nodes, possibly except for the detection of locally unreachable NHs.

The protection ensures that BIER multicast traffic is forwarded to all destinations that are reachable over the routing underlay and that no duplicates occur. The protection is fast as it works as soon as the BFR is informed about a unreachable NH and the underlying routing works again after the failure occurred.

BIER-FRR link protection is able to protect single link failures within a network provided that the underlying routing can restore full connectivity. Multiple link failures within a network are not necessarily protected.

BIER-FRR node protection protects both single link and single node failures within a network provided that the underlying routing can restore full connectivity. Multiple link and node failures within a network are not necessarily protected.

The design of BIER-FRR guarantees loop-freeness on the BIER layer. Since the BIER packet is tunneled, the BIER is header is only used for forwarding if the tunneled packet arrives at the designated BFR. Loop-freeness on the routing underlay is out of the scope of this document.

5. Necessary Changes to the BIER Architecture

This section serves as an overview to list the necessary conceptual features or changes that are required for BIER-FRR.

5.1. Unicast Tunneling

Unicast tunnels to connect two not directly adjacent BFRs are already available. This feature is described in Section 6.9 of [RFC8279].

5.2. Detecting Unreachable N(N)Hs

A liveness component (e.g. BFD) has to be added to enable the detection of unreachable NHs. This feature has been proposed in [I-D.hu-bier-bfd].

5.3. BIFT with backup entries

The BIFT has to be extended with backup entries as described in Section XXX. When the regular BIER forwarding procedure yields an unreachable NH, the backup entry contains a backup F-BM for header modification and a NNH to which the BIER packet should be tunneled to.

6. Security Considerations

This memo does not extend the security considerations for BIER.

7. IANA Considerations

This document requests no action by IANA.

8. References

8.1. 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.
- [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>>.

8.2. Informative References

- [I-D.xiong-bier-resilience]
Xiong, Q., hu, f., and G. Mirsky, "The Resilience for
BIER", draft-xiong-bier-resilience-01 (work in progress),
October 2018.
- [RFC5286] Atlas, A., Ed. and A. Zinin, Ed., "Basic Specification for
IP Fast Reroute: Loop-Free Alternates", RFC 5286,
DOI 10.17487/RFC5286, September 2008,
<<https://www.rfc-editor.org/info/rfc5286>>.

Authors' Addresses

Daniel Merling
University of Tuebingen
Sand 13
Tuebingen 72076
Germany

Phone: +49 7071 29-70507
Email: daniel.merling@uni-tuebingen.de

Michael Menth
University of Tuebingen
Sand 13
Tuebingen 72076
Germany

Phone: +49 7071 29-70505
Email: menth@uni-tuebingen.de

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.

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.

Copyright Notice

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

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

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

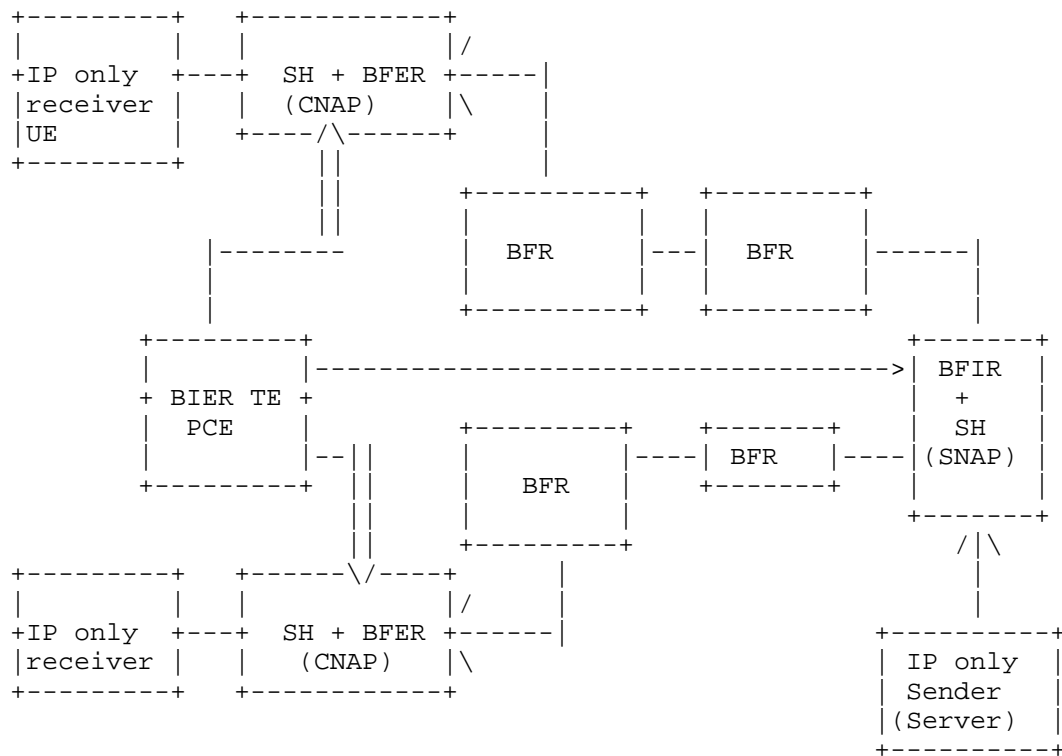
client requests are converted into the BIER or BIER-TE bitstring and used to send the HTTP response.

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 send and receive HTTP transactions through CNAP.

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

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

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

As part of 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:

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

5. Realization over IP Multicast

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

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

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

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

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

5.1. Mapping to Requirements

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

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

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

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

5.2. Problems

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

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

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

6. Realization over BIER

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

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

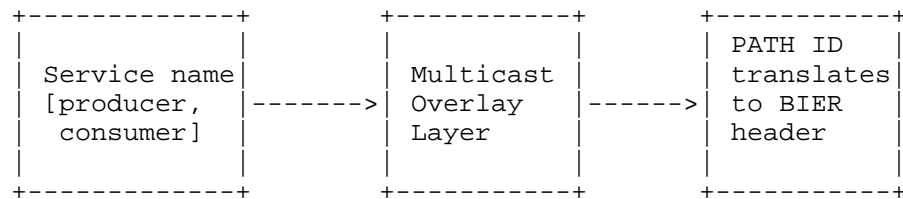


Figure 2: Service name to Path ID translation

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

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

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

6.1.1. BIER Multicast Overlay Components

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

- o Service Handler (SH): The Service handler terminates transport level protocols, such as TCP, and extracts the URI. It processes the URI in order to determine the PATH ID by contacting the PCE for a suitable path resolution, which in turn is used to send the HTTP Request.
- o Optional PCE : Path Computation Element keeps track of all service execution end points through a registration process. SH interacts with the PCE to obtain PATH information by resolving the FQDN from the incoming URI at the ingress SH to a suitable PATH ID.
- o Interface functions to BFIR where the PATH ID is mapped to BIER header. An Interface to the BFER is likely not required because the BFER will only receive the traffic that they need and should be able to derive from the BIER payload which subset of its receivers need to get an HTTP encapsulated version of a particular reply.

6.1.2. 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.

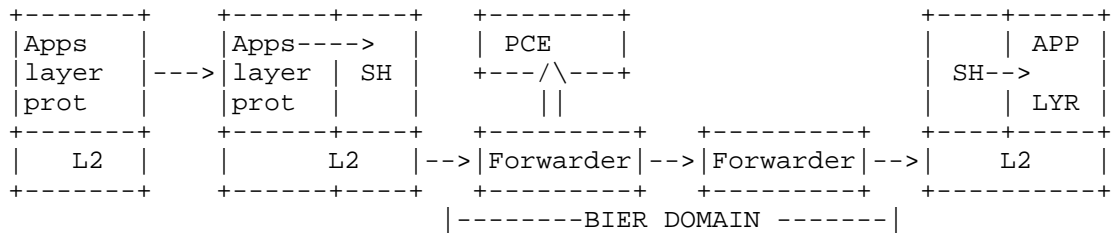


Figure 3: Protocol for Multicast Overlay Layer

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.

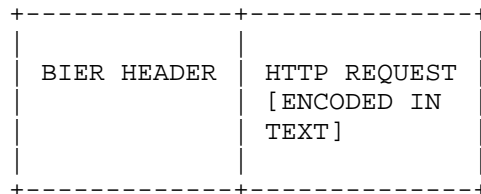


Figure 4: Encapsulation of Service Request

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.

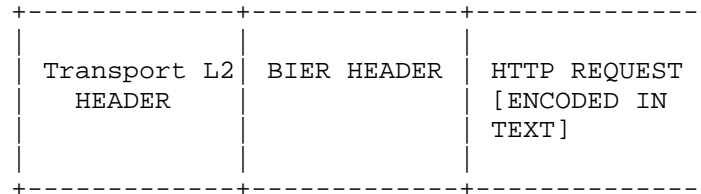


Figure 5: Transport Encapsulation of BIER payload

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_bluebook.pdf>.

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

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

[I-D.ietf-bier-use-cases]

Kumar, N., Asati, R., Chen, M., Xu, X., Dolganow, A., Przygienda, T., Gulko, A., Robinson, D., Arya, V., and C. Bestler, "BIER Use Cases", draft-ietf-bier-use-cases-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: Standards Track
Expires: September 12, 2019

J. Xie
M. McBride
S. Dhanaraj
Huawei Technologies
L. Geng
China Mobile
March 11, 2019

Use of BIER IPv6 Encapsulation (BIERv6) for Multicast VPN in Non-MPLS
IPv6 networks
draft-xie-bier-ipv6-mvpn-00

Abstract

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

Requirements Language

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

Status of This Memo

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

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

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

This Internet-Draft will expire on September 12, 2019.

Copyright Notice

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

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

Table of Contents

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

1. Introduction

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

SRv6 explained in [I-D.ietf-spring-segment-routing] refers to the deployment of segment routing in Non-MPLS IPv6 networks. [I-D.filsfils-spring-srv6-network-programming] introduces the Network programming concepts in SRv6 networks and explains how the 128-bit IPv6 address can be used as SRv6 SID in the format LOC:FUNCT, where

LOC part of the SID is routable, while FUNCT part of the SID is an opaque identification of a local function bound to the SID. It has also defined some well known standard functions like End.DT4 - Endpoint with decaps and IPv4 table lookup for L3VPN (equivalent to per-VRF VPN label). [I-D.dawra-idr-srv6-vpn] defines the TLVs to associate a function like End.DT4 with the L3VPN Unicast routes advertised via BGP. It also details how the functions of End.DT4, End.DT6, End.DT46 (End.DTx) can be used to identify a L3VPN/EVPN instead of using a VPN Label in MPLS-VPN [RFC4364] of the received data packet and thereby realize the L3VPN Services in the SRv6 Networks. However, it covers unicast services exclusively.

This document describes a method to realize MVPN services using BIER as a P-tunnel in the BIERv6 Networks. It defines a method to use an SRv6 Service SID like End.DTx as source address to identify the MVPN instance at the Egress PE. While the End.DTx is used as IPv6 destination address in unicast L3VPN scenarios as defined in [I-D.dawra-idr-srv6-vpn], this document extends the use of End.DTx as IPv6 source address. The LOC part and FUNCT part of this SRv6 Service SID represent the context and the upstream-assigned VPN Label respectively in MVPN scenario's as defined in [I-D.ietf-bier-mvpn].

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

GTM defined in [RFC7716] is also covered in this document, as GTM shares the same BGP-MVPN signaling, while providing an approach of Non-VPN multicast over a service provider core with various P-tunnel type. For the same reason of UMH selection, and the requirement of basic operation like ping (e.g, to the multicast source address), the Global IPv4/IPv6 over SRv6 Core as described in [I-D.dawra-idr-srv6-vpn] is also required, and the [RFC5549] may be required further.

2. Terminology

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

- o BIERv6 - BIER in Non-MPLS IPv6 networks using the BIER IPv6 encapsulation format defined in [I-D.xie-bier-ipv6-encapsulation].

- o SRv6 - Segment Routing instantiated on the IPv6 dataplane as defined in [I-D.filsfils-spring-srv6-network-programming].
- o SRv6 SID - SRv6 Segment Identifier as defined in [I-D.filsfils-spring-srv6-network-programming].
- o End.DTx - Refers to the functions End.DT6, End.DT4, End.DT46 defined in [I-D.filsfils-spring-srv6-network-programming].
- o SRv6 L3 Service - L3VPN/Global-L3 service in Non-MPLS SRv6 network defined in [I-D.dawra-idr-srv6-vpn], or MVPN/GTM service in Non-MPLS BIERv6 network defined in this document.

3. Use of PTA and Prefix-SID Attribute in x-PMSI A-D Routes

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

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

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

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

4. MVPN over BIERv6 Core

[I-D.ietf-bier-mvpn] specifies the protocol and procedures to be followed by the Ingress and Egress PEs to use BIER as a P-tunnel for MVPN in MPLS networks. This section specifies the required changes and procedures in addition to support BIER as a P-tunnel in Non-MPLS IPv6 networks.

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

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

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

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

If the VPN is IPv4 VPN, the End.DTx can be either End.DT4 or End.DT46. If the VPN is IPv6 VPN, the End.DTx can be either End.DT6 or End.DT46. By default, the distribution of the x-PMSI A-D routes uses the same End.DTx as the ones used for the distribution of VPN-IP unicast routes. That is, by default, the x-PMSI A-D route MUST carry the same SRv6-Service-SID used by the unicast routing for L3VPN. The

default could be modified via configuration by having a End.DTx used for the BGP-MVPN x-PMSI A-D routes being distinct from the ones used for the VPN-IP unicast routes.

BFIR MAY carry the BGP Prefix-SID attribute only in I-PMSI A-D route when I-PMSI A-D route is used, while other S-PMSI A-D routes do not carry the BGP Prefix-SID attribute.

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

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

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

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

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

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

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

5. GTM over BIERv6 Core

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

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

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

- o Route Distinguishers - the RD field of a BGP-MVPN route's NLRI MUST be set to zero (i.e., to 64 bits of zero) to represent a Non-VPN GTM. See section 2.2 of [RFC7716].
- o Route Targets Extended Community (EC) - The RT EC carried by the BGP-MVPN C-multicast (Type 6 or 7) route or Leaf A-D (Type 4) route MUST be an IPv6-address-specific Extended Community (EC). The Global Administrator field identifies the Upstream PE or the UMH, and the Local Administrator field MUST always be set to zero in GTM case.
- o VRF Route Import Extended Community (EC) - The VRF Route Import EC used in BIERv6 core MUST be an IPv6-address-specific EC if used, either used in UMH-eligible unicast routes having a SAFI of 1, 2, or 4, or used in the MVPN (SAFI of 5) Source Active A-D route.

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

6. Data Plane

6.1. Encapsulation of Multicast Traffic

BIER IPv6 encapsulation (BIERv6) [I-D.xie-bier-ipv6-encapsulation] is used for forwarding the c-multicast traffic through an IPv6 core. The following diagram shows the progression of an MVPN c-multicast packet as it enters and leaves the intra-AS service-provider network.

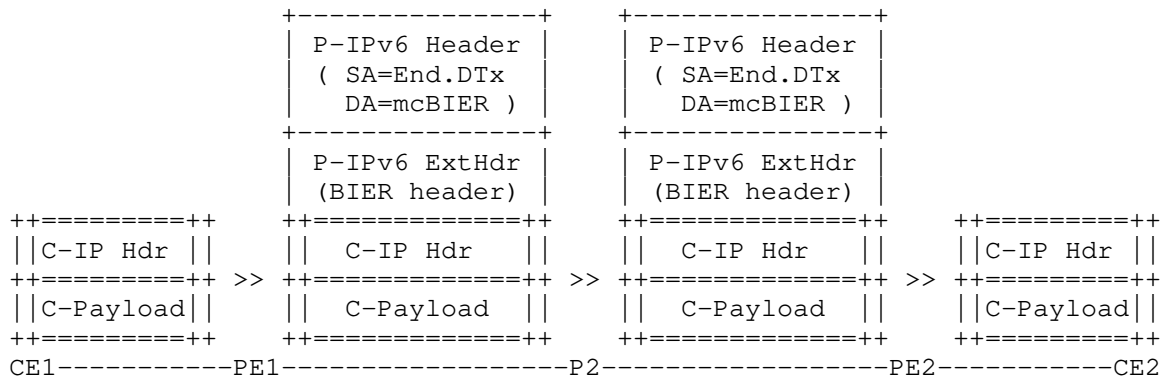


Figure 1: BIERv6 MVPN/GTM Intra-AS

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

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

6.2. MTU

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

6.3. TTL

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

7. Security Considerations

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

8. IANA Considerations

No IANA allocation is required.

9. Acknowledgements

TBD.

10. References

10.1. Normative References

[I-D.dawra-idr-srv6-vpn]

Dawra, G., Filsfils, C., Dukes, D., Brissette, P., Camarillo, P., Leddy, J., daniel.voyer@bell.ca, d., daniel.bernier@bell.ca, d., Steinberg, D., Raszuk, R., Decraene, B., Matsushima, S., and S. Zhuang, "BGP Signaling for SRv6 based Services.", draft-dawra-idr-srv6-vpn-05 (work in progress), October 2018.

[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-07 (work in progress), February 2019.

[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-spring-segment-routing]

Filsfils, C., Previdi, S., Ginsberg, L., Decraene, B., Litkowski, S., and R. Shakir, "Segment Routing Architecture", draft-ietf-spring-segment-routing-15 (work in progress), January 2018.

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

Xie, J., Geng, L., McBride, M., Dhanaraj, S., Yan, G., and Y. Xia, "Encapsulation for BIER in Non-MPLS IPv6 Networks", draft-xie-bier-ipv6-encapsulation-00 (work in progress), March 2019.

- [RFC5549] Le Faucheur, F. and E. Rosen, "Advertising IPv4 Network Layer Reachability Information with an IPv6 Next Hop", RFC 5549, DOI 10.17487/RFC5549, May 2009, <<https://www.rfc-editor.org/info/rfc5549>>.
- [RFC6515] Aggarwal, R. and E. Rosen, "IPv4 and IPv6 Infrastructure Addresses in BGP Updates for Multicast VPN", RFC 6515, DOI 10.17487/RFC6515, February 2012, <<https://www.rfc-editor.org/info/rfc6515>>.
- [RFC6625] Rosen, E., Ed., Rekhter, Y., Ed., Hendrickx, W., and R. Qiu, "Wildcards in Multicast VPN Auto-Discovery Routes", RFC 6625, DOI 10.17487/RFC6625, May 2012, <<https://www.rfc-editor.org/info/rfc6625>>.
- [RFC7716] Zhang, J., Giuliano, L., Rosen, E., Ed., Subramanian, K., and D. Pacella, "Global Table Multicast with BGP Multicast VPN (BGP-MVPN) Procedures", RFC 7716, DOI 10.17487/RFC7716, December 2015, <<https://www.rfc-editor.org/info/rfc7716>>.
- [RFC8279] Wijnands, IJ., Ed., Rosen, E., Ed., Dolganow, A., Przygienda, T., and S. Aldrin, "Multicast Using Bit Index Explicit Replication (BIER)", RFC 8279, DOI 10.17487/RFC8279, November 2017, <<https://www.rfc-editor.org/info/rfc8279>>.
- [RFC8296] Wijnands, IJ., Ed., Rosen, E., Ed., Dolganow, A., Tantsura, J., Aldrin, S., and I. Meilik, "Encapsulation for Bit Index Explicit Replication (BIER) in MPLS and Non-MPLS Networks", RFC 8296, DOI 10.17487/RFC8296, January 2018, <<https://www.rfc-editor.org/info/rfc8296>>.

10.2. Informative References

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

Authors' Addresses

Jingrong Xie
Huawei Technologies

Email: xiejingrong@huawei.com

Mike McBride
Huawei Technologies

Email: michael.mcbride@huawei.com

Senthil Dhanaraj
Huawei Technologies

Email: senthil.dhanaraj@huawei.com

Liang Geng
China Mobile

Email: gengliang@chinamobile.com

BIER WG
Internet-Draft
Intended status: Standards Track
Expires: September 25, 2019

Zheng. Zhang
Cui. Wang
Ran. Chen
Fangwei. Hu
ZTE Corporation
Mahesh. Sivakumar
Cisco Systems, Inc.
Huanan. Chen
China Telecom
March 24, 2019

BIER TE YANG model
draft-zhang-bier-te-yang-07

Abstract

This document defines a YANG data model for BIER TE configuration and operation.

Status of This Memo

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

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <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 September 25, 2019.

Copyright Notice

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

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must

include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	2
2. Design of the Data Model	2
3. BIER-TE configuration	3
4. Notifications	4
5. RPCs	4
6. BIER TE YANG model	4
7. IANA Considerations	13
8. Acknowledgement	13
9. Normative References	13
Authors' Addresses	15

1. Introduction

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

2. Design of the Data Model

The BIER TE YANG model includes BIER TE adjacency configuration and forwarding items configuration. Some features can also be used to enhance BIER TE function, like ECMP and FRR.

```

module: ietf-bier-te
  augment /rt:routing:
    +--rw bier-te
      +--rw subdomain* [subdomain-id]
        +--rw subdomain-id    uint16
      +--rw te-adj-id
        +--rw si* [si]
          +--rw si            uint16
          +--rw adj* [adj-id]
            +--rw adj-id      uint16
            +--rw adj-if      if:interface-ref
            +--rw bp-type?    enumeration
      +--rw bsl* [fwd-bsl]
        +--rw fwd-bsl        uint16
        +--rw si* [si]
          +--rw si            uint16
          +--rw te-bift-id

```



```

|--rw type?      enumeration
+--rw value      rt-types:mpls-label
+--rw fwd-items* [te-bp]
    +--rw te-bp          uint16
    +--rw bp-type?       enumeration
    +--rw (fwd-type)
        |--:(connected)
        |--:(routed)
        |--:(local-decap)
        |--:(other)
    +--rw dnr-flag?      boolean
    +--rw out-info
        +--rw fwd-intf          if:interface-ref
        +--rw te-out-bift-id
            +--rw type?          enumeration
            +--rw value          rt-types:mpls-label
    +--rw te-frr {bier-te-frr}?
        +--rw frr-index?         uint16
        +--rw resetbitmask* [bitmask]
            +--rw bitmask        bit-string
    +--rw te-ecmp* [out-if] {bier-te-ecmp}?
        +--rw out-if              if:interface-ref
        +--rw te-out-bift-id
            +--rw type?           enumeration
            +--rw value           rt-types:mpls-label
+--rw te-frr-items {bier-te-frr}?
    +--rw btaft* [frr-index]
        +--rw frr-index          uint16
        +--rw frr-si             uint16
        +--rw frr-bsl            uint16
        +--rw addbitmask* [bitmask]
            +--rw bitmask        bit-string

notifications:
+---n bier-te-notification
    +--ro bp-is-zero* [if-index]
        +--ro if-index          if:interface-ref
        +--ro bp-type?          enumeration

```

3. BIER-TE configuration

The BIER-TE forwarding item is indexed by the combination of sub-domain-id, BitStringLength and set identifier.

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

Because the BIER-TE is controlled by controller now, the information about IGP is not defined. If in the future the IGP is used to carry the information about BIER-TE, the IGP extension will be added in this document.

4. Notifications

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

5. RPCs

TBD.

6. BIER TE YANG model

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

```
module ietf-bier-te {  
  yang-version 1.1;  
  namespace "urn:ietf:params:xml:ns:yang:ietf-bier-te";  
  prefix bier-te;  
  import ietf-routing {  
    prefix "rt";  
    reference "RFC8022";  
  }  
  import ietf-interfaces {  
    prefix "if";  
    reference "RFC7223";  
  }  
  import ietf-routing-types {  
    prefix "rt-types";  
    reference "RFC8294";  
  }  
  organization " IETF BIER(Bit Indexed Explicit Replication)  
                Working Group";  
  contact  
    "WG Web:  <http://tools.ietf.org/wg/bier/>  
    WG List:  <mailto:bier@ietf.org>
```

```
Editor: Zheng Zhang
      <mailto:zhang.zheng@zte.com.cn>
Editor: Cui Wang
      <mailto:wang.cuil@zte.com.cn>
Editor: Ran Chen
      <mailto:chen.ran@zte.com.cn>
Editor: Fangwei Hu
      <mailto:hu.fangwei@zte.com.cn>
Editor: Mahesh Sivakumar
      <mailto:masivaku@cisco.com>
";

description
  " The module defines the YANG definitions for BIER TE.

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

  Redistribution and use in source and binary forms, with or
  without modification, is permitted pursuant to, and subject
  to the license terms contained in, the Simplified BSD License
  set forth in Section 4.c of the IETF Trust's Legal Provisions
  Relating to IETF Documents
  (http://trustee.ietf.org/license-info).
  This version of this YANG module is part of RFC 3618; see
  the RFC itself for full legal notices.";

revision 2019-03-23 {
  description
    "Initial revision.";
  reference
    "draft-ietf-bier-te-arch: Traffic Engineering for Bit Index
    Explicit Replication (BIER-TE)";
}

/*
 * Features
 */
feature bier-te-frr {
  description
    "Support Fast Re-route feature in BIER TE.";
}
feature bier-te-ecmp {
  description
    "Support ECMP feature in BIER TE.";
}

typedef bit-string {
```

```
    type uint16;
    description "The bit mask of one bit-string.";
}

grouping te-frr {
    description "The TE fast re-route information.";
    list btaft {
        key "frr-index";
        description "The index of the frr paths. This item can be
            used for multiple links protection in
            different SI.";
        leaf frr-index {
            type uint16;
            mandatory true;
            description "The frr item index.";
        }
        leaf frr-si {
            type uint16;
            mandatory true;
            description "The set identifier of this forwarding
                item.";
        }
        leaf frr-bsl {
            type uint16;
            mandatory true;
            description "The value of bitstringlength.";
        }
        list addbitmask {
            key "bitmask";
            description "The adding bitmask of the forwarding
                item.";
            leaf bitmask {
                type bit-string;
                description "The adding bitmask of the forwarding
                    item. This item should be merged
                    into the packet's bit-string.";
            }
        }
    }
}

grouping fwd-type {
    description "The collection of all possible forwarding types.";
    choice fwd-type {
        mandatory true;
        case connected {
            description "The forwarding type is connected.
                Mostly connected interfaces.";
        }
    }
}
```

```
    }
    case routed {
        description "The forwarding type is routed.
                    Mostly not connected interfaces.";
    }
    case local-decap {
        description "Means that the packet should be
                    decapsulated and forward out
                    of BIER domain.";
    }
    case other {
        description "Means that the packet should be
                    discarded.";
    }
    description "The collection of all possible forwarding
                types.";
}

}

grouping bp-type {
    description "The collection of all possible adjacency type.";

    leaf bp-type {
        type enumeration {
            enum p2p {
                description "Describes p2p adjacency.";
            }
            enum bfer {
                description "Describes bfer adjacency.";
            }
            enum leaf-bfer {
                description "Describes leaf-bfer adjacency.
                            There is no next BFR that the packet
                            should be forwarded.";
            }
            enum lan {
                description "Describes lan adjacency.";
            }
            enum spoke {
                description "Describes spoke adjacency of
                            hub-and-spoke.";
            }
            enum ring-clockwise {
                description "Describes clockwise adjacency in
                            ring.";
            }
            enum ring-counterclockwise {
                description "Describes counterclockwise adjacency in
```

```
        ring.";
    }
    enum ecmp {
        description "Describes ecmp adjacency.
                    When the type is set to ecmp,
                    the corresponding ecmp entry
                    should be used to balance the load.";
    }
    enum virtual-link {
        description "Describes virtual adjacency
                    between two indirect connect
                    nodes.";
    }
    enum other {
        description "Describes other id type of
                    adjacency.";
    }
    }
    description "The collection of all possible adjacency
                type.";
}

grouping te-bift-id {
    description "The index of BIER forwarding items. It usually
                represents the combination of [SD, BSL, SI].";
    leaf type {
        type enumeration {
            enum mpls {
                description "The bift-id value is represent the
                            BIER TE mpls forwarding plane. It
                            is a mpls label.";
            }
            enum eth {
                description "The bift-id value is represent the
                            BIER TE ethernet forwarding plane.
                            It is an index of ethernet
                            encapsulation.";
            }
            enum other {
                description "Describes other type of te-bift-id.";
            }
        }
        description "The types of BIER TE bift-id. If this type
                    is not set, mpls is default type.";
    }
    leaf value {
        type rt-types:mpls-label;
    }
}
```

```
        mandatory true;
        description "The bift-id value of the forwarding
                     item. It can be a mpls label or an
                     index of ethernet encapsulation which
                     is used to represent specific
                     combination of [SD, BSL, SI]. The
                     ethernet index value is the same range
                     (20bits) as mpls label.";
    }
}

grouping te-items {
    description "The BIER TE forwarding items collection.";
    uses fwd-type;

    leaf dnr-flag {
        type boolean;
        description
            "When the flag is set to 1, the BP of adjacency
             should not be reset when packet copies are
             created. The flag makes sense only when the
             forwarding type is 'connected'.";
    }

    container out-info {
        description "The information of out forwarding
                     packets. Includes the outbound interface
                     and the bift-id of next hop.";
        leaf fwd-intf {
            type if:interface-ref;
            mandatory true;
            description "The out interface of this
                         forwarding item.";
        }
        container te-out-bift-id {
            description "The bift-id information
                         corresponding to a specific
                         outbound interface.";
            uses te-bift-id;
        }
    }

    container te-frr {
        if-feature bier-te-frr;
        leaf frr-index {
            type uint16;
            description "The index of this frr path.";
        }
    }
}
```

```
    list resetbitmask {
        key "bitmask";
        description "The deleting bitmask of the
                    forwarding item.";
        leaf bitmask {
            type bit-string;
            description "The deleting bitmask of the
                    forwarding item.";
        }
    }
    description "If this link is protected, frr items can
                be used to forward flows when this link
                is down.";
}

}

grouping fwd-items {
    list si {
        key "si";
        description "The forwarding items of one set identifier.";
        leaf si {
            type uint16;
            mandatory true;
            description "The set identifier of this forwarding
                    item.";
        }

        container te-bift-id {
            description "The bift-id which is used to locate the
                    specific forwarding item.";
            uses te-bift-id;
        }
    }

    list fwd-items {
        key "te-bp";
        description "The forwarding information of one BIER TE
                    item.";
        leaf te-bp {
            type uint16;
            mandatory true;
            description "The bit index of a BIER TE forwarding
                    item.";
        }
    }

    uses bp-type;
    uses te-items;

    list te-ecmp {
        if-feature bier-te-ecmp;
```



```
        key "out-if";
        leaf out-if {
            type if:interface-ref;
            description "The outgoing interface.";
        }
        container te-out-bift-id {
            description "The bift-id info for a specific
                        outbound interface.";
            uses te-bift-id;
        }
        description "The list of the ecmp paths. When the
                    type of BP is set to ecmp, this
                    interface ecmp list should be used to
                    balance the load on each interface.";
    }
}
description "The forwarding items in one combination of
            SD, BSL and SI.";
}

grouping te-info {
    description "The BIER TE forwarding information.";
    list subdomain {
        key "subdomain-id";
        description "The forwarding items of one sub-domain.";
        leaf subdomain-id {
            type uint16;
            description "The sub-domain-id of this sub-domain.";
        }
    }

    container te-adj-id {
        list si {
            key "si";
            description "The forwarding items of a set
                        identifier.";
            leaf si {
                type uint16;
                mandatory true;
                description "The set identifier of this
                            forwarding item.";
            }
        }

        list adj {
            key "adj-id";
            description "The ID of an adjacency.";
            leaf adj-id {
                type uint16;
            }
        }
    }
}
```

```
        mandatory true;
        description "The adjacency id.";
    }
    leaf adj-if {
        type if:interface-ref;
        mandatory true;
        description "The corresponding interface
                    of this adjacency.";
    }
    uses bp-type;
}
description "This adjacency ID information for BIER TE
            in a SI.";
}

list bsl {
    key "fwd-bsl";
    description "The forwarding items in one BSL.";
    leaf fwd-bsl {
        type uint16;
        description "The value of bitstringlength.";
    }
    uses fwd-items;
}

container te-frr-items {
    if-feature bier-te-frr;
    uses te-frr;
    description "The TE protective fast re-route items.";
}
}

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

/*
 * Notifications
 */
```

```
notification bier-te-notification {
  description
    "The notification is sent when a condition changes.";
  list bp-is-zero {
    key "if-index";
    description "The adjacency id is zero. It is invalid.";
    leaf if-index {
      type if:interface-ref;
      description "The adjacency id is zero.";
    }
    uses bp-type;
  }
}
}
<CODE ENDS>
```

7. IANA Considerations

The IANA is requested to assign two new URIs from the IETF XML registry ([RFC3688]). Authors are suggesting the following URI:

URI: urn:ietf:params:xml:ns:yang:ietf-bier-te

Registrant Contact: BIER WG

XML: N/A, the requested URI is an XML namespace

This document also requests one new YANG module name in the YANG Module Names registry ([RFC6020]) with the following suggestion:

name: ietf-bier-te

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

prefix: bier-te

reference: RFC XXXX

8. Acknowledgement

The authors would like to thank Min Gu (gumin20181129@163.com) for her testing, verification and valuable suggestion.

9. Normative 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.
- [I-D.ietf-bier-bier-yang]
Chen, R., hu, f., Zhang, Z., dai.xianxian@zte.com.cn, d.,
and M. Sivakumar, "YANG Data Model for BIER Protocol",
draft-ietf-bier-bier-yang-04 (work in progress), September
2018.
- [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-01 (work in progress), October
2018.
- [RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, RFC 3688,
DOI 10.17487/RFC3688, January 2004,
<<https://www.rfc-editor.org/info/rfc3688>>.
- [RFC6020] Bjorklund, M., Ed., "YANG - A Data Modeling Language for
the Network Configuration Protocol (NETCONF)", RFC 6020,
DOI 10.17487/RFC6020, October 2010,
<<https://www.rfc-editor.org/info/rfc6020>>.
- [RFC6087] Bierman, A., "Guidelines for Authors and Reviewers of YANG
Data Model Documents", RFC 6087, DOI 10.17487/RFC6087,
January 2011, <<https://www.rfc-editor.org/info/rfc6087>>.
- [RFC7223] Bjorklund, M., "A YANG Data Model for Interface
Management", RFC 7223, DOI 10.17487/RFC7223, May 2014,
<<https://www.rfc-editor.org/info/rfc7223>>.
- [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>>.
- [RFC8349] Lhotka, L., Lindem, A., and Y. Qu, "A YANG Data Model for
Routing Management (NMDA Version)", RFC 8349,
DOI 10.17487/RFC8349, March 2018,
<<https://www.rfc-editor.org/info/rfc8349>>.

Authors' Addresses

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

Email: zhang.zheng@zte.com.cn

Cui(Linda) Wang
ZTE Corporation

Email: lindawangjoy@gmail.com

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

Email: chen.ran@zte.com.cn

Fangwei Hu
ZTE Corporation
No.889 Bibo Rd
Shanghai
China

Email: hu.fangwei@zte.com.cn

Mahesh Sivakumar
Cisco Systems, Inc.
510 McCarthy Blvd
Milpitas, California 95035
United States

Email: masivaku@cisco.com

Huanan Chen
China Telecom
109 West Zhongshan Ave
Guangzhou, Guangdong 510630
China

Phone: +86 20 38639346
Email: chenhuanan@gsta.com