

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: 28 April 2022

H. Chen
M. McBride
Futurewei
A. Wang
China Telecom
G. Mishra
Verizon Inc.
Y. Liu
China Mobile
M. Menth
University of Tuebingen
B. Khasanov
Yandex LLC
X. Geng
Huawei
Y. Fan
Casa Systems
L. Liu
Fujitsu
X. Liu
Volta Networks
25 October 2021

BIER Egress Protection
draft-chen-bier-egress-protect-03

Abstract

This document describes a mechanism for fast protection against the failure of an egress node of a "Bit Index Explicit Replication" (BIER) domain. It is called BIER egress protection. It does not require any per-flow state in the core of the domain. With BIER egress protection the failure of a primary BFER (Bit Forwarding Egress Router) is protected with a backup BFER such that traffic destined to the primary BFER in the BIER domain is fast rerouted by a neighbor BFR to the backup BFER on the BIER layer. The mechanism is applicable if all BIER traffic sent to the primary BFER can reach its destination also via the backup BFER. It is complementary to BIER-FRR which cannot protect against the failure of a BFER.

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] [RFC8174] when, and only when, they appear in all capitals, as shown here.

Status of This Memo

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

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

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

This Internet-Draft will expire on 28 April 2022.

Copyright Notice

Copyright (c) 2021 IETF Trust and the persons identified as the document authors. All rights reserved.

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

Table of Contents

1. Introduction	3
1.1. Terminology	3
2. Overview of BIER Egress Protection	4
3. Protocol Extensions	7
3.1. Extensions to OSPF	7
3.2. Extensions to IS-IS	8
4. Extensions to BIFT	9
4.1. Integrated one BIFT	9
4.1.1. EP-BIFT on BFR as PLR	9
4.1.2. EP-BIFT on Backup Egress	12
4.1.3. Updated Forwarding Procedure for Integrated BIFT	14
4.2. Multiple Backup BIFTs	15
4.2.1. Multiple Backup BIFTs on BFR as PLR	16
4.2.2. Multiple Backup BIFTs on Backup Egress	17
4.2.3. Updated Forwarding Procedure for Multiple BIFTs	18

4.2.4. Switching between EP and Normal Forwarding	19
5. Example Application of BIER Egress Protection	20
5.1. BIRT and BIFT on a BFR	20
5.2. Backup BIRTs and Backup BIFTs on a BFR	21
5.3. Forwarding using Backup BIFT	24
6. Security Considerations	25
7. IANA Considerations	25
8. Acknowledgements	25
9. References	25
9.1. Normative References	25
9.2. Informative References	26
Authors' Addresses	27

1. Introduction

[RFC8279] specifies "Bit Index Explicit Replication" (BIER). It provides optimal forwarding of multicast data packets through a "multicast/BIER domain". It does not require the use of a protocol for explicitly building multicast distribution trees, and it does not require intermediate nodes to maintain any per-flow state.

This document describes a mechanism for fast protection against the failure of an egress node of a "Bit Index Explicit Replication" (BIER) domain, which is called BIER Egress Protection.

This BIER Egress Protection does not require intermediate nodes to maintain any per-flow state for fast protection against the failure of an egress node of the flow.

1.1. Terminology

BFR: Bit-Forwarding Router.

BFIR: Bit-Forwarding Ingress Router.

BFER: Bit-Forwarding Egress Router.

BFR-id: BFR Identifier. It is a number in the range [1,65535].

BFR-NBR: BFR Neighbor.

F-BM: Forwarding Bit Mask.

BFR-prefix: An IP address (either IPv4 or IPv6) of a BFR.

BIRT: Bit Index Routing Table. It is a table that maps from the BFR-id (in a particular sub-domain) of a BFER to the BFR-prefix of that BFER, and to the BFR-NBR on the path to that BFER.

BIFT: Bit Index Forwarding Table.

FRR: Fast Re-Route.

PLR: Point of Local Repair.

LFA: Loop-Free Alternate.

Basic LFA: It is the LFA defined in [RFC5286].

RLFA: Remote LFA. It is the LFA defined in [RFC7490].

TI-LFA: Topology Independent LFA. It is the LFA defined in [I-D.ietf-rtgwg-segment-routing-ti-lfa].

IGP: Interior Gateway Protocol.

LSDB: Link State DataBase.

SPF: Shortest Path First.

SPT: Shortest Path Tree.

OSPF: Open Shortest Path First.

IS-IS: Intermediate System to Intermediate System.

LSA: Link State Advertisement in OSPF.

LSP: Link State Protocol Data Unit (PDU) in IS-IS.

FIB: Forwarding Information Base or Forwarding Table.

2. Overview of BIER Egress Protection

This section introduces BIER egress protection and describes its operation using the BIER topology in Figure 1 as an example. The figure illustrates a BIER sub-domain with the 8 nodes/BFRs A, B, C, D, E, F, G and H. Each link connecting these nodes/BFRs has a cost. The cost of a link (for routing purposes) is indicated in the figure unless it is 1 by default. Nodes/BFRs D, F, E, H and A are BFERs and have BFR-ids 1, 2, 3, 4, and 5 respectively. For simplicity, these BFR-ids are represented by (SI:BitString), where SI = 0 and BitString is 5 bits long. BFR-ids 1, 2, 3, 4, and 5 are represented by (0:00001), (0:00010), (0:00100), (0:01000) and (0:10000), respectively.

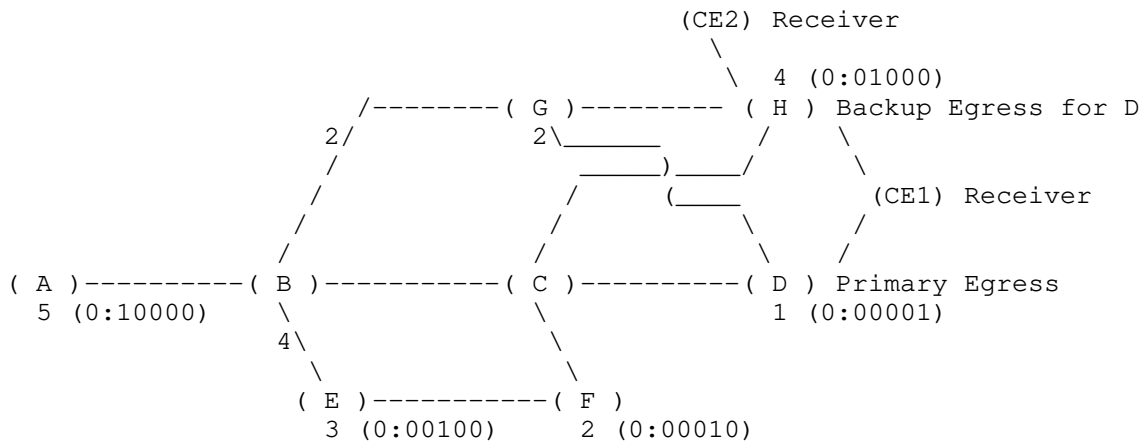


Figure 1: Example BIER topology

CE1 and CE2 in neighboring networks are multicast traffic receivers. CE1 is connected to both BFER D and BFER H. CE2 is connected to H but it is not connected to D.

We explain BIER egress protection for primary BFER D using backup BFER H. At first, BFER H is configured to protect BFER D. In addition, whether primary egress D and backup egress H send their BIER packets' payloads to the same receiver CE1 (i.e., after decapsulating their BIER packets, whether they send the same decapsulated packets to the same receiver CE1) is configured. And then, this information is distributed to BFR D's neighbors (BFR C and BFR G) and the domain by IGP. BFR C, BFR G, and BFER H know that H is the backup egress to protect the primary egress D. Two different backup strategies or methods, Bit Protection Switching and Proxy Backup, are specified for two different configurations regarding to whether D and H send their BIER packets' payloads to the same receiver.

1. Bit Protection Switching: If a neighbor of D detects D's outage, it performs the following operations on all the packets that are destined to D. It clears the bit for destination D and sets the bit for H. Afterwards, these packets are forwarded towards H and eventually reach H which decapsulates them and delivers their payloads to the same receiver CE as D does.
2. Proxy Backup: If a neighbor as PLR of D detects D's outage, it

reroutes a copy of the packet with D as a destination towards H. When H as backup BFER detects its primary BFER D's outage, H, acting as a proxy of D, decapsulates all the BIER packets with destination D and forwards their payloads according to D's forwarding behavior for the payloads.

Bit Protection Switching is well applicable to the case where primary egress D and backup egress H send their BIER packets' payloads to the same receiver CE1. In this case, after D decapsulates D's BIER packet (i.e., the BIER packet with BFER D as a destination), D sends the decapsulated packet (i.e., the payload of the BIER packet) to receiver CE1 through its multicast layer. After H decapsulates H's BIER packet (i.e., the BIER packet with BFER H as a destination), H sends the same decapsulated packet (i.e., the same payload as the one in D's BIER packet) to the same receiver CE1 through its multicast layer as D.

During normal operations, there is no multicast traffic to CE1 from backup egress H, and CE1 receives the multicast traffic only from primary egress D. There is no duplicated traffic to receiver CE1.

When primary egress D fails, the BIER packet with destination D is updated through bit switch (i.e., the bit for D is cleared and bit for H is set in the packet) by a PLR such as BFR C when the PLR detects the failure of D. The updated packet with destination H is sent to backup egress H. H decapsulates the packet and delivers the packet's payload to its multicast layer, which sends the payload to CE1.

Proxy Backup is applicable to the case where D and H send their BIER packets' payloads to different receivers. In this case, after D decapsulates D's BIER packet, D sends the decapsulated packet (i.e., the payload of the BIER packet) to receiver CE1 through its multicast layer. After H decapsulates H's BIER packet, H drops the same decapsulated packet (i.e., the same payload as the one in D's BIER packet) or sends it to different receiver CE2 through its multicast layer.

During normal operations, primary egress D sends the payload of the BIER packet with destination D to receiver CE1 and backup egress H sends the payload of the BIER packet with destination H to receiver CE2. H sends the BIER packet with destination D towards node D along the shortest path to D.

When D fails, the BIER packet with destination D is sent to backup egress H by a PLR such as BFR C when the PLR detects the failure of D. H acting as a proxy of D MUST have a fast way to detect the failure of D and obtain the forwarding behavior of D for the payload

of the BIER packet with destination D in advance. When H as the proxy of D detects the failure of D, it sends the payload of the BIER packet with destination D to receiver CE1 according to the forwarding behavior of D for the payload.

Backup egress H may obtain the forwarding behavior of its primary egress D for the payload of the BIER packet with the primary egress as a destination from configurations or through some protocols such as BGP or PCEP. How for a backup egress to obtain the forwarding behavior of its primary egress is out scope of this document.

The fast egress protection mechanism in this document is different from MoFRR in [RFC7431], where the same traffic is sent through two separated paths/trees to both primary egress node D and backup egress node H, to which the receiver CE1 is dual homed. It will use less network resources such as link bandwidth than MoFRR in [RFC7431].

3. Protocol Extensions

This section defines extensions to OSPF and IS-IS for advertising the backup information (including the backup egress node for protecting a primary egress node).

3.1. Extensions to OSPF

When a node P (as a primary egress node) has a backup egress node configured to protect against its failure, node P advertises the information about the backup egress node to its neighbors in its router information opaque LSA of LS type 9 or 10. Using the LSA of LS type 9, node P will advertise the information only to its neighbors (which will not advertise the information further). Using the LSA of LS type 10, node P will advertise the information to the whole BIER network domain (i.e., P's neighbors will advertise the information further until the information reaches every node in the domain). The information is included in a backup egress node TLV. The format of the TLV is shown in Figure 2.

After each of the neighbors receives the backup egress node TLV, it knows that node P as a primary egress node will be protected by the backup egress node in the TLV. Once detecting the failure of node P, it sends the BIER packet with the bit for destination P towards node P's backup egress node.

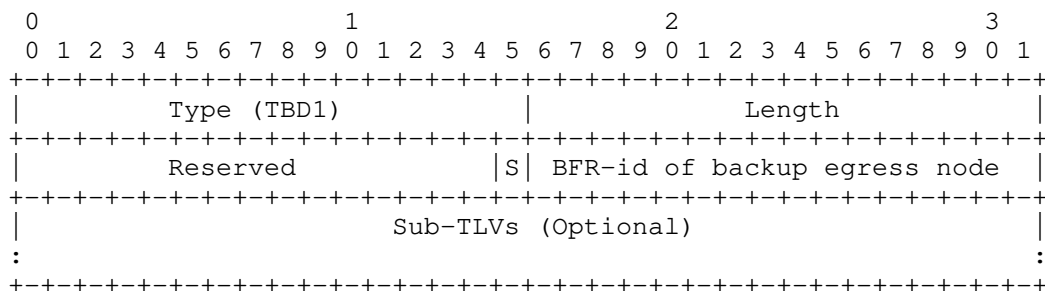


Figure 2: OSPF Backup Egress TLV

Type: 2 octets, its value (TBD1) is to be assigned by IANA.

Length: 2 octets, its value is 4 plus the length of the Sub-TLVs included. If no Sub-TLV is included, its value is 4.

Reserved: 15 bits, they MUST be set to zero when sending and be ignored while receiving.

S flag: 1 bit. It is set to one to indicate that the primary egress and backup egress send their BIER packets' payloads to the same CE receiver ; it is set to zero to indicate that the primary egress and backup egress send their BIER packets' payloads to different CE receivers .

BFR-id of backup egress node: 2 octets, its value is the BFR-id of the backup egress node configured to protect against the failure of the primary egress node.

Sub-TLVs (Optional): No Sub-TLV is defined now.

3.2. Extensions to IS-IS

For supporting fast protection against the failure of a primary egress node in a BIER domain, a new IS-IS TLV, called IS-IS backup egress node TLV, is defined. It contains the BFR-id of a backup egress node.

When a node P (as a primary egress node) has a backup egress node configured to protect against its failure, node P advertises the information about the backup egress node using a IS-IS backup egress node TLV.

This TLV may be advertised in IS-IS Hello (IIH) PDUs, LSPs, or in Circuit Scoped Link State PDUs (CS-LSP) [RFC7356]. Using CS-LSP or IIH PDUs, node P will advertise the information only to its

neighbors. Using LSPs, node P will advertise the information to the whole BIER network domain. The format of the TLV is shown in Figure 3.

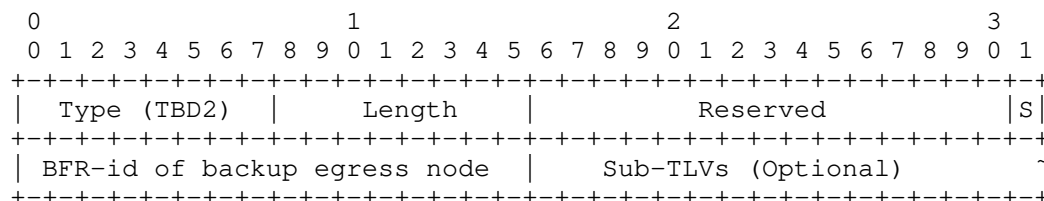


Figure 3: IS-IS Backup Egress TLV

Type: 1 octet, its value (TBD2) is to be assigned by IANA.

Length: 1 octet, its value is 4 plus the length of the Sub-TLVs included. If no Sub-TLV is included, its value is 4.

The other fields are the same as those in Figure 2.

4. Extensions to BIFT

This section specifies the BIFT extended for egress protection (EP-BIFT) on a BFR as a PLR and the BIFT extended on a backup egress node. In one option, the EP-BIFT is implemented in an Integrated one BIFT. In another, it is implemented in Multiple Backup BIFTs.

4.1. Integrated one BIFT

A BFR has an integrated BIFT for both normal operations and protections against the failure of each of its neighbor BFRs. That is that the normal BIFT on the BFR is extended to have a backup entry (or say sub-entry) for each of its neighbor BFRs.

4.1.1. EP-BIFT on BFR as PLR

To protect a primary egress node (e.g., BFER D in Figure 1), a BFR as the primary egress node's neighbor (e.g., BFR C in Figure 1) and a PLR has a backup entry in its BIFT extended for egress protection (EP-BIFT). The backup entry contains: Backup Entry Active (BEA), Same CE receiver (SC), Backup Egress BFER (BE-BFER), Backup F-BM (BF-BM) and Backup BFR-NBR (BBFR-NBR).

* BEA = 1 indicates that the Backup Entry for egress protection is active.

- * SC = 1 indicates that both primary egress node and backup egress node send their BIER packets' payloads to the same receiver CE.
- * BE-BFER is the BFR-id of the backup egress node for the primary egress node.
- * BBFR-NBR is the backup BFR-NBR to the backup egress node (e.g., H in Figure 1). When SC = 1 (i.e., both primary egress node and backup egress node send their BIER packets' payloads to the same receiver CE), the BFR finds a basic, remote or topology independent (TI) LFA to the backup egress node and sets BBFR-NBR to the LFA. When SC = 0 (i.e., the primary egress node and its backup egress node send their BIER packets' payloads to different receiver CEs), the BFR obtains the value of BBFR-NBR in following steps. At first, the BFR finds a basic, remote or TI LFA to the backup egress node. And then the BFR checks if the LFA is the backup egress node or the backup egress node is on the shortest path from the LFA to the primary egress node without going through the primary egress node. If so, the LFA is used as the BBFR-NBR; otherwise (i.e., the LFA is not the backup egress node and the backup egress node is not on the shortest path from the LFA to the primary egress node without going through the primary egress node), the BBFR-NBR is set to the backup egress node through a tunnel to the backup egress node without going through the primary egress node. This is to make sure that the BIER packet with the primary egress node as a destination reaches the backup egress node.

When primary egress node (e.g., BFER D in Figure 1) fails, the BFR as a PLR sets BEA in the entry for primary egress node to one after the BFR detects the failure. The BFR uses the backup entry with BEA = 1 to forward the BIER packet with primary egress node as a destination. The BFR forwards the packet to BBFR-NBR. Before forwarding the packet, the BFR checks whether SC equals to one in the entry. If SC = 1, the BFR as a PLR replaces the primary egress node as a destination with its backup egress node as a destination through clearing the bit for primary egress node (e.g., D) as a destination in the BIER packet and setting the bit for backup egress node (e.g., H) as a destination in the packet.

For example, the integrated BIFT (or say EP-BIFT) on BFR C in Figure 1 is shown in Figure 4.

BFR-id (SI:BitString)	F-BM	BFR-NBR	BEA	SC	BE-BFER	BF-BM	BBFR-NBR
1 (0:00001)	00001	D	0	1	H(01000)	01001	H
2 (0:00010)	00110	F	0	0	E(00100)	00010	E(TI-LFA)
3 (0:00100)	00110	F	0	0	F(00010)	00110	F
4 (0:01000)	01000	H	0	1	D(00001)	01001	D
5 (0:10000)	10000	B	0		0	NULL	NULL

Figure 4: Integrated BIFT on BFR C

BFR C in Figure 1 has three neighbor BFERs D, F and H with BFR-ids 1, 2 and 4 respectively. The backup entry for BFER D with BFR-id = 1 is the last five columns in the first row of Figure 4.

- * BEA = 0 means that D is working well.
- * SC = 1 means that the primary egress node D and backup egress node H send their BIER packets' payloads to the same CE receiver.
- * BE-BFER = H means that H is the backup egress node for primary egress node D.
- * BF-BM = 01001 is computed by ORing the bit of BFR-id with BFR-NBR = H and the bit of BFR-id with BBFR-NBR = H. BFR-id = 1 is with BBFR-NBR = H and BFR-id = 4 is with BFR-NBR = H.
- * BBFR-NBR = H means that BFER H is the next hop on the shortest path to H without going D.

The backup entry for BFER F with BFR-id = 2 is the last five columns in the second row of Figure 4.

- * BEA = 0 means that F is working well.
- * SC = 0 means that the primary egress node F and backup egress node E send their BIER packets' payloads to different CE receivers.
- * BE-BFER = E means that E is the backup egress node for primary egress node F.

- * BF-BM = 00010 is computed by ORing the bit of BFR-id with BFR-NBR = E and the bit of BFR-id with BBFR-NBR = E. Since there is no BFR-id with BFR-NBR = E, BF-BM = 00010.
- * BBFR-NBR = E (TI-LFA) means that B and E in Figure 1 are not on the shortest path to E without going F and TI-LFA tunnel is used to send primary egress node F's BIER packet to backup egress node E when F fails and BEA is set to one.

The backup entry for BFER H is similar to the one for BFER D. The backup entry for BFER E is similar to the one for BFER F.

4.1.2. EP-BIFT on Backup Egress

If a primary egress node (e.g., D in Figure 1) and its backup egress node (e.g., H in Figure 1) send their BIER packets' payloads to the same receiver CE (e.g., CE1 in Figure 1), then the forwarding entry for the primary egress node in the BIFT on the backup egress node keeps the same as normal.

For example, the integrated BIFT on backup egress node H in Figure 1 with SC = 1 is the same as H's normal BIFT, which is illustrated in Figure 5.

BFR-id (SI:BitString)	F-BM	BFR-NBR
1 (0:00001)	10111	C
2 (0:00010)	10111	C
3 (0:00100)	10111	C
4 (0:01000)	01000	H
5 (0:10000)	10111	C

Figure 5: Integrated BIFT on Backup Egress H with SC = 1

If the primary egress node and the backup egress node send their BIER packets' payloads to different receiver CEs, for example, D as a primary egress node sends its BIER packet's payload to CE1, H as the backup egress node for D sends its BIER packet's payload to CE2, then the forwarding entry for the primary egress node on the backup egress node is extended to contain a backup entry for primary egress node. The backup entry includes:

- * Backup Entry Active (BEA), SC, BE-BFER, Backup F-BM (BF-BM). These have the same meanings as those in Section 4.1.1.
- * Backup BFR-NBR or Pointer to FIB for Primary Egress (BBFR-NBR/P-FIB) is a pointer to the FIB for the primary egress node. Using this FIB, the backup egress node will forward the payload of the BIER packet with the primary egress node as a destination to the same CE receiver as the primary egress node.

BEA is set to one when the backup egress node detects the failure of the primary egress node. After detecting the failure and receiving the BIER packet with the bit for the primary egress node as a destination set to one, the backup egress node forwards the packet's payload to the primary egress node's CE receiver using the backup forwarding entry with BEA = 1.

For example, the integrated BIFT on backup egress node H in Figure 1 with SC = 0 is illustrated in Figure 6.

BFR-id (SI:BitString)	F-BM	BFR-NBR	BEA	SC	BE-BFER	BF-BM	BBFR-NBR /P-FIB
1 (0:00001)	10111	C	0	0	H(01000)	00001	P-FIB-4D
2 (0:00010)	10111	C	0	0			NULL
3 (0:00100)	10111	C	0	0			NULL
4 (0:01000)	01000	H	0	0			NULL
5 (0:10000)	10111	C	0				NULL

Figure 6: Integrated BIFT on Backup Egress H with SC = 0

In Figure 6, the backup entry for primary egress node D with BFR-id = 1 is the last five columns in the first row.

- * BEA = 0 means that D is working well.
- * SC = 0 means that the primary egress node D and backup egress node H send their BIER packets' payloads to different CE receivers.
- * BE-BFER = H means that H is the backup egress node for primary egress node D.

- * BF-BM = 00001 is computed by ORing the bit of BFR-id with BFR-NBR = P-FIB-4D and the bit of BFR-id with BBFR-NBR = P-FIB-4D. Since there is no BFR-id with BFR-NBR = P-FIB-4D, BF-BM = 00001.
- * BBFR-NBR/P-FIB = P-FIB-4D is the pointer to the FIB for the primary egress node D. When D fails and BEA is set to one, backup egress node H for D acts as a proxy of D and sends D's BIER packet's payload to CE receiver CE1 using the FIB for D. Backup egress node H for D decapsulates the BIER packet with D as a destination and forwards the payload using the FIB for D after it detects the failure of D.

4.1.3. Updated Forwarding Procedure for Integrated BIFT

The forwarding procedure defined in [RFC8279] is updated/enhanced for integrated BIFT to consider the egress protection.

For a multicast packet with the BitString indicating a BFER as one of its destinations, the updated forwarding procedure on a BFR as a PLR sends the packet towards the backup egress node of the BFER if the BFER is protected. On the backup egress, the procedure sends the packet's payload to the BFER's CE receiver.

It checks whether BEA = 1 in the forwarding entry for the BFER. If BEA = 1, it determines whether the current node is backup egress node. On backup egress node, the procedure sends the packet's payload to the CE receiver. On the BFR as a PLR, the procedure sends the packet copy to BBFR-NBR. Before sending the packet copy, the procedure updates the packet copy by clearing the bit for primary egress node and setting the bit for backup egress node when primary egress node and backup egress node send their BIER packets' payload to the same CE receiver. The bits for the other destinations which are not through BBFR-NBR are cleared in the packet copy's BitString by ANDing the BitString with BF-BM. The original packet's BitString is updated to remove the bits for the destinations towards which the packet copy is sent through BBFR-NBR by ANDing the BitString with the INVERSE of BF-BM.

The updated forwarding procedure for integrated BIFT is described in Figure 7.

```

Packet = the packet received by BFR;
FOR each BFER k (from the rightmost in Packet's BitString) {
  IF BFER k is the BFR itself {
    copies Packet, sends the copy to the multicast
    flow overlay and clears bit k in Packet's BitString
  } ELSE {
    finds the row in the EP-BIFT for the sub-domain using
    Packet's SI and BitString as the key/index
    IF BEA == 1 { // Primary Egress fails
      IF (BBFR-NBR/P-FIB is Pointer to FIB) { // on Backup Egress
        Sends payload to CE using the FIB for primary egress;
      } ELSE {
        IF (SC == 1) { // on PLR and SC == 1
          clears bit k in Packet's BitString; // BFER k is PE-BFER
          sets bit j in Packet's BitString; // BFER j is BE-BFER
        } // SC == 0, no updates to packet
        Copies Packet, updates the copy's BitString by ANDing it
        with BF-BM in the entry, sends updated copy to BBFR-NBR;
      }
      updates Packet's BitString by ANDing it with
      the INVERSE of BF-BM;
    } ELSE {
      Copies Packet, updates the copy's BitString by ANDing
      it with F-BM in the entry, sends updated copy to BFR-NBR;
      updates Packet's BitString by ANDing it with the INVERSE
      of the F-BM in the entry
    }
  }
}

```

Figure 7: Updated Forwarding Procedure for Integrated BIFT

4.2. Multiple Backup BIFTs

A BFR has a normal BIFT and multiple backup BIFTs for egress protection. For each of the BFR's neighbor BFERs, the BFR has a backup BIFT for the BFER, which considers the failure of the BFER. In normal operations, the BFR uses its normal BIFT to forward all the BIER packets. When the BFR detects the failure of the BFER, the BFR uses the backup BIFT for the BFER to forward all the BIER packets.

4.2.1. Multiple Backup BIFTs on BFR as PLR

A BFR as a PLR has a backup BIFT for a BFER that has the same structure as the normal BIFT except for a backup BFER (BE-BFER) for the BFER and same CE receiver (SC) flag indicating whether the BE-BFER and BFER send their BIER packets' payloads to the same CE receiver. In the entry for the BFER in the backup BIFT, the value of BFR-NBR is the backup BFR-NBR (BBFR-NBR), which is computed in the same way as the BBFR-NBR is computed in Section 4.1.1.

For example, the backup BIFT for BFER D on BFR C in Figure 1 is shown in Figure 8. The backup BIFT for D considers BFER D's failure.

BFR-id (SI:BitString)	F-BM	BFR-NBR	SC	BE-BFER
1 (0:00001)	01001	H	1	H(01000)
2 (0:00010)	00110	F		
3 (0:00100)	00110	F		
4 (0:01000)	01001	H		
5 (0:10000)	10000	B		

Figure 8: BFR C's Backup BIFT for BFER D

In Figure 8, the entry for BFER D with BFR-id = 1 has its BFR-NBR with value of the BBFR-NBR (which is H) and contains SC = 1 and BE-BFER = H. BE-BFER = H means that BFER H is the backup egress node for primary egress node D. SC = 1 means that primary egress node D and backup egress node H send their BIER packets' payloads to the same CE receiver.

For the entry with BFR-NBR = X, its F-BM has the bit of the BFR-id in each entry with BFR-NBR = X. For example, the first entry with BFR-NBR = H, its F-BM in the first entry has the bit of BFR-id = 1 and BFR-id = 4 in the first entry and the fourth entry, which are with BFR-NBR = H.

When BFR C detects the failure of BFER D, it uses the backup BIFT for D to forwards all the BIER packets. For the packet with destination D (i.e., BitString = 00001), BFR C sends the packet to BFR-NBR H after clearing the bit for primary egress node D and setting the bit for backup egress node H since SC = 1. The packet received by H

contains BitString = 01000 for destination H. After receiving the packet, BFER H sends the packet's payload to the same CE receiver CE1.

If SC = 0, BFR C sends the packet to BFR-NBR H without clearing the bit for D or setting the bit for H. After receiving the packet with destination D (i.e., BitString 00001) and detecting the failure of D, BFER H as a proxy of D sends the packet's payload to primary egress node D's CE receiver CE1.

4.2.2. Multiple Backup BIFTs on Backup Egress

When a primary egress node and its backup egress node send their BIER packets' payloads to the same CE receiver, the backup BIFT for the primary egress node on the backup egress node is the same as the normal BIFT on the backup egress node. For example, the backup BIFT for primary egress node on backup egress node H in Figure 1 with SC = 1 is the same as H's normal BIFT, which is illustrated in Figure 5.

When a primary egress node and its backup egress node send their BIER packets' payloads to different CE receivers, the backup BIFT for the primary egress node on the backup egress node considers the failure of the primary egress node. The BFR-NBR/P-FIB in the entry for the primary egress node is the pointer to the FIB for the primary egress node which is used to forward the payload of the BIER packet with the primary egress node as a destination. For example, the backup BIFT for primary egress node D on backup egress node H in Figure 1 with SC = 0 is illustrated in Figure 9.

BFR-id (SI:BitString)	F-BM	BFR-NBR /P-FIB	SC	BE-BFER
1 (0:00001)	00001	P-FIB-4D	0	H(01000)
2 (0:00010)	00110	C		
3 (0:00100)	00110	C		
4 (0:01000)	01001	H		
5 (0:10000)	10000	C		

Figure 9: Backup Egress H's Backup BIFT for Egress D

In Figure 9, the entry for BFER D with BFR-id = 1 has its BFR-NBR/P-FIB = P-FIB-4D (the pointer to the FIB for primary egress node D) and contains BE-BFER = H and SC = 0. BE-BFER = H means that BFER H is the backup egress node for primary egress node D. SC = 0 means that primary egress node D and backup egress node H send their BIER packets' payloads to different CE receivers. Note that the last two columns can be removed since they are not used for forwarding.

When backup egress node H detects the failure of primary egress node D, node H uses the backup BIFT for egress D to forward all the BIER packets. For the packet with destination D (i.e., BitString = 00001), node H as a proxy of D sends the packet's payload to the CE1 (D's CE receiver) using the FIB for BFER D, which contains the forwarding behavior of primary egress node D for the payload of D's BIER packet.

4.2.3. Updated Forwarding Procedure for Multiple BIFTs

The updated forwarding procedure for multiple BIFTs is illustrated in Figure 10. This forwarding procedure is used with the normal BIFT on a BFR in normal operations. It is used with a backup BIFT for a primary egress node on a BFR as a PLR and on a backup egress node when the primary egress node fails.

On the backup egress node (i.e., BFR-NBR/P-FIB is a pointer to the FIB for the primary egress node), the procedure sends the payload of the packet with primary egress node/BFER as a destination to the BFER's CE receiver.

The forwarding procedure on a BFR as a PLR for each of multiple backup BIFTs is the same as the one defined in [RFC8279] except for sending the packet with primary egress node as a destination to the backup egress node of primary egress node. Before sending the packet to the backup egress node, the procedure updates the BitString in the packet by clearing the bit for the primary egress node and setting the bit for the backup egress node when SC = 1 (i.e., the primary egress node and backup egress node send their BIER packets' payloads to the same CE receiver).

```

Packet = the packet received by BFR;
FOR each BFER k (from the rightmost in Packet's BitString) {
  IF BFER k is the BFR itself {
    copies Packet, sends the copy to the multicast
    flow overlay and clears bit k in Packet's BitString
  } ELSE {
    finds the row in the EP-BIFT for the sub-domain using
    Packet's SI and BitString as the key/index
    IF (BFR-NBR/P-FIB is Pointer to FIB) { // on Backup Egress
      Sends payload using the FIB for the primary egress;
    } ELSE {
      IF (SC == 1) { // on PLR and SC == 1
        clears bit k in Packet's BitString; // BFER k is PE-BFER
        sets bit j in Packet's BitString; // BFER j is BE-BFER
      } // SC == 0, no updates to packet
      Copies Packet, updates the copy's BitString by ANDing
      it with F-BM in the entry, sends updated copy to BFR-NBR;
    }
    updates Packet's BitString by ANDing it with the INVERSE
    of the F-BM in the entry
  }
}

```

Figure 10: Updated Forwarding Procedure for Multiple BIFTs

4.2.4. Switching between EP and Normal Forwarding

When multiple backup BIFTs are used, the multiple backup BIFTs are pre-computed and installed ready for activation when an egress node failure is detected. In normal operations, a BFR uses its normal BIFT to forward BIER packets. Once the BFR detects the failure of its BFR-NBR X as an egress, it activates (i.e., uses) the backup BIFT for X to forward BIER packets and de-activates (i.e., does not use) its normal BIFT. After activation of the backup BIFT, it remains in effect until it is no longer required.

In general, when the routing protocol has re-converged on the new topology taking into account the failure of X, the BIRT is re-computed using the updated LSDB and the BIFT is re-derived from the BIRT. Once the BIFT is installed ready for activation, it is activated to forward packets with BIER headers and the backup BIFT for X is de-activated.

From the new topology, the BFR computes/re-computes the backup BIRT for each BFR-NBR Y as an egress and the backup BIFT for Y is derived/re-derived from the backup BIRT for Y. The backup BIFT is installed/re-installed ready for activation when Y fails.

5. Example Application of BIER Egress Protection

This section illustrates an example application of BIER Egress Protection using multiple backup BIFTs on a BFR in a BIER topology in Figure 1.

5.1. BIRT and BIFT on a BFR

Every BFR in a BIER sub-domain/topology builds and maintains a Bit Index Routing Table (BIRT). For the BIER topology in Figure 1, each of 8 nodes/BFRs A, B, C, D, E, F, G and H builds and maintains a BIRT using the LSDB for the topology.

The BIRT built on BFR C (i.e., node C) is shown in Figure 11.

BFR-id (SI:BitString)	BFR-Prefix of Dest BFER	BFR-NBR (Next Hop)
1 (0:00001)	D	D
2 (0:00010)	F	F
3 (0:00100)	E	F
4 (0:01000)	H	H
5 (0:10000)	A	B

Figure 11: Bit Index Routing Table on BFR C

The 1st row in the BIRT indicates that the next hop BFR-NBR on the shortest path to BFER D with BFR-id 1 is BFR D.

The 2nd row in the BIRT indicates that the next hop BFR-NBR on the shortest path to BFER F with BFR-id 2 is BFR F.

The 3rd row in the BIRT indicates that the next hop BFR-NBR on the shortest path to BFER E with BFR-id 3 is BFR F.

The 4-th row in the BIRT indicates that the next hop BFR-NBR on the shortest path to BFER H with BFR-id 4 is BFR H.

The 5-th row in the BIRT indicates that the next hop BFR-NBR on the shortest path to BFER A with BFR-id 5 is BFR B.

From this BIRT on BFR C, a Bit Index Forwarding Table (BIFT) is derived. This BIFT is shown in Figure 12.

The 2nd and 3-th rows in the BIRT have the same SI = 0 and next hop BFR-NBR = F. The F-BM for each of these two rows in the BIFT is the logical OR of the BitStrings of these rows, which is 00110 (00010 OR 00100 = 00110).

The F-BM for 1st row in the BIFT is 00001.

The F-BM for 4-th row in the BIFT is 01000.

The F-BM for 5-th row in the BIFT is 10000.

BFR-id (SI:BitString)	F-BM	BFR-NBR (Next Hop)
1 (0:00001)	00001	D
2 (0:00010)	00110	F
3 (0:00100)	00110	F
4 (0:01000)	01000	H
5 (0:10000)	10000	B

Figure 12: Bit Index Forwarding Table on BFR C

5.2. Backup BIRTs and Backup BIFTs on a BFR

Each of the BFRs that are neighbors of egress nodes (i.e., BFERs) in a BIER sub-domain/topology builds and maintains a number of Egress Protection Bit Index Routing Tables (EP-BIRTs) or say backup BIRTs.

For the BIER topology in Figure 1,

BFR B is the neighbor of BFERs A and E;
 BFR C is the neighbor of BFERs D, F and H;
 BFR E is the neighbor of BFER F;
 BFR F is the neighbor of BFER E;
 BFR G is the neighbor of BFERs D and H.

Each of 5 nodes/BFRs B, C, E, F and G builds and maintains a number of backup BIRTs using the LSDB for the topology for its every BFR-NBR as an egress node.

For example, BFR C (i.e., node C) in the BIER topology builds and maintains three backup BIRTs for its three BFR-NBRs (BFRs D, F and H) that are egress nodes respectively.

The backup BIRT for BFER D built by BFR C based on the BIRT on BFR C (refer to Figure 11) is shown in Figure 13.

The BIRT is copied to the backup BIRT for BFER D (i.e., the first three columns of the backup BIRT). The new backup information (i.e., the 4-th column) for every row in the backup BIRT is initialized with BE-BFER = 0/NULL.

BFR-id (SI:BitString)	BFR-Prefix of Dest BFER	BFR-NBR (Next Hop)	BE-BFER
1 (0:00001)	D	H	H
2 (0:00010)	F	F	0
3 (0:00100)	E	F	0
4 (0:01000)	H	H	0
5 (0:10000)	A	B	0

Figure 13: C's Backup BIRT for BFER D

In the backup BIRT for BFER D, the row that has Destination BFER == D is the 1st row. This row has the new backup information BE-BFER = H, which indicates that BFER D (i.e., primary egress node D) is protected by BFER H (i.e., backup egress node H). Each of the other rows has the new backup information BE-BFER = 0/NULL.

The 1st row in the EP-BIRT indicates that the packet with destination D will be sent to D's backup egress node H when D fails.

The 2nd row in the backup BIRT indicates that the next hop BFR-NBR on the path to BFER F with BFR-id 2 is BFR F.

The 3rd row in the backup BIRT indicates that the next hop BFR-NBR on the path to BFER E with BFR-id 3 is BFR F.

The 4-th row in the backup BIRT indicates that the next hop BFR-NBR on the path to BFER H with BFR-id 4 is BFR H.

The 5-th row in the backup BIRT indicates that the next hop BFR-NBR on the path to BFER A with BFR-id 5 is BFR B.

From this backup BIRT for BFER D on BFR C, an Egress Protection Bit Index Forwarding Table (EP-BIFT) or say backup BIFT for BFER D is derived. This backup BIFT for BFER D is shown in Figure 14.

The first and 4-th rows in the backup BIRT have the same next hop BFR-NBR = H. The F-BM for each of these two rows in the backup BIFT is the logical OR of the BitStrings of these rows, which is 01001 (00001 OR 01000 = 01001).

The 2nd and 3rd rows in the backup BIRT have the same next hop BFR-NBR = E. The F-BM for each of these two rows in the backup BIFT is the logical OR of the BitStrings of these rows, which is 00110 (00010 OR 00100 = 00110).

BFR-id (SI:BitString)	F-BM	BFR-NBR (Next Hop)	SC	BE-BFER
1 (0:00001)	01001	H	1	H
2 (0:00010)	00110	F	0	0
3 (0:00100)	00110	F	0	0
4 (0:01000)	01001	H	0	0
5 (0:10000)	10000	B	0	0

Figure 14: C's Backup BIFT for BFER D

The F-BM for 5-th row in the backup BIFT is 10000.

5.3. Forwarding using Backup BIFT

Suppose that there is a multicast traffic from BFR A as ingress/BFIR to egresses/BFERs D, F and E. For every packet of the traffic, after receiving it, BFR A adds a BIER header into the packet and sends the packet with the BIER header to BFR B, which sends the packet BFR C. The BIER header contains (SI:BitString) = (0:00111) for egresses/BFERs D, F and E.

In normal operations, after receiving the packet from BFR B, BFR C copies, updates and sends the packet to BFR D and BFR F using the normal BIFT on BFR C according to the forwarding procedure defined in [RFC8279].

Once BFR C detects the failure of its BFR-NBR D, which is a BFER, after receiving the packet from BFR B, BFR C copies, updates and sends the packet using the backup BIFT for BFER D on BFR C according to the updated forwarding procedure.

For the packet targeting to BFER D (i.e., primary egress node), BFR C sends it towards BFER H (i.e., backup egress node), which is configured to protect BFER D.

For example, once BFR C detects the failure of its BFR-NBR D, after receiving the packet from BFR B, BFR C copies, updates and sends the packet to BFR H and BFR F using the backup BIFT for BFER D on BFR C.

The packet received by BFR C from BFR B contains (SI:BitString) = (0:00111). The rightmost one bit in BitString is bit 1. For BFER 1 (0:00001) (i.e., BFR D as BFER), BFR C gets the 1st row (i.e., forwarding entry) in the backup BIFT for BFER D. BE-BFER = H in the row indicates that BFER D is protected against the failure of D by backup BFER H. BFR C clears bit 1 in Packet's BitString and sets bit 4 (i.e., the bit for BE-BFER = H) in Packet's BitString to one since SC = 1. The BitString in Packet is 01110 now. BFR C copies, updates the BitString by ANDing it with F-BM (which is 01001) and sends the packet copy with BitString = 01000 to BFR-NBR H in the entry.

After sending the packet to H, BFR C updates the original packet by ANDing its BitString with the INVERSE of the F-BM in the first row. The updated BitString = 00110, which is 01110 & ~F-BM in the row = 01110 & 10110 = 00110.

For the packet containing BitString = 00110, the rightmost one bit in BitString is bit 2. For BFER 2 (0:00010) (i.e., BFR F as BFER), BFR C gets the 2nd row (i.e., forwarding entry) in the backup BIFT for BFER D. The next hop BFR-NBR is F in the row. BFR C copies, updates and sends the packet to F.

The packet sent to F contains the updated BitString = 00110, which is 00110 & F-BM in the 2nd row = 00110 & 00110 = 00110.

After sending the packet to F, BFR C updates the original packet by ANDing its BitString with the INVERSE of the F-BM in the 2nd row. The updated BitString = 00000, which is 00110 & ~F-BM in the row = 00110 & 11001 = 00000.

The updated packet has BitString without any one bit. BFR C finishes forwarding the packet to F and H (backup for D). BFR F will send the packet to E.

6. Security Considerations

TBD.

7. IANA Considerations

No requirements for IANA.

8. Acknowledgements

The authors would like to thank Jeffrey Zhang, Jingrong Xie for their comments to this work.

9. References

9.1. 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>>.
- [RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", RFC 5226, DOI 10.17487/RFC5226, May 2008, <<https://www.rfc-editor.org/info/rfc5226>>.
- [RFC5250] Berger, L., Bryskin, I., Zinin, A., and R. Coltun, "The OSPF Opaque LSA Option", RFC 5250, DOI 10.17487/RFC5250, July 2008, <<https://www.rfc-editor.org/info/rfc5250>>.
- [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>>.

- [RFC5714] Shand, M. and S. Bryant, "IP Fast Reroute Framework", RFC 5714, DOI 10.17487/RFC5714, January 2010, <<https://www.rfc-editor.org/info/rfc5714>>.
- [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>>.
- [RFC7356] Ginsberg, L., Previdi, S., and Y. Yang, "IS-IS Flooding Scope Link State PDUs (LSPs)", RFC 7356, DOI 10.17487/RFC7356, September 2014, <<https://www.rfc-editor.org/info/rfc7356>>.
- [RFC7490] Bryant, S., Filss, C., Previdi, S., Shand, M., and N. So, "Remote Loop-Free Alternate (LFA) Fast Reroute (FRR)", RFC 7490, DOI 10.17487/RFC7490, April 2015, <<https://www.rfc-editor.org/info/rfc7490>>.
- [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>>.
- [RFC7770] Lindem, A., Ed., Shen, N., Vasseur, JP., Aggarwal, R., and S. Shaffer, "Extensions to OSPF for Advertising Optional Router Capabilities", RFC 7770, DOI 10.17487/RFC7770, February 2016, <<https://www.rfc-editor.org/info/rfc7770>>.
- [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>>.
- [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>>.

9.2. Informative References

- [I-D.ietf-rtgwg-segment-routing-ti-lfa]
Litkowski, S., Bashandy, A., Filss, C., Francois, P., Decraene, B., and D. Voyer, "Topology Independent Fast

Reroute using Segment Routing", Work in Progress, Internet-Draft, draft-ietf-rtgwg-segment-routing-ti-lfa-07, 29 June 2021, <<https://www.ietf.org/archive/id/draft-ietf-rtgwg-segment-routing-ti-lfa-07.txt>>.

[I-D.ietf-spring-segment-protection-sr-te-paths]

Hegde, S., Bowers, C., Litkowski, S., Xu, X., and F. Xu, "Segment Protection for SR-TE Paths", Work in Progress, Internet-Draft, draft-ietf-spring-segment-protection-sr-te-paths-01, 11 July 2021, <<https://www.ietf.org/archive/id/draft-ietf-spring-segment-protection-sr-te-paths-01.txt>>.

[RFC7431] Karan, A., Filss, C., Wijnands, IJ., Ed., and B. Decraene, "Multicast-Only Fast Reroute", RFC 7431, DOI 10.17487/RFC7431, August 2015, <<https://www.rfc-editor.org/info/rfc7431>>.

[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>>.

Authors' Addresses

Huaimo Chen
Futurewei
Boston, MA,
United States of America

Email: Huaimo.chen@futurewei.com

Mike McBride
Futurewei

Email: michael.mcbride@futurewei.com

Aijun Wang
China Telecom
Beiqijia Town, Changping District
Beijing
102209
China

Email: wangaj3@chinatelecom.cn

Gyan S. Mishra
Verizon Inc.
13101 Columbia Pike
Silver Spring, MD 20904
United States of America

Phone: 301 502-1347
Email: gyan.s.mishra@verizon.com

Yisong Liu
China Mobile

Email: liuyisong@chinamobile.com

Michael Menth
University of Tuebingen

Email: menth@uni-tuebingen.de

Boris Khasanov
Yandex LLC
Moscow

Email: bhassanov@yahoo.com

Xuesong Geng
Huawei

Email: gengxuesong@huawei.com

Yanhe Fan
Casa Systems
United States of America

Email: yfan@casa-systems.com

Lei Liu
Fujitsu
United States of America

Email: liulei.kddi@gmail.com

Xufeng Liu
Volta Networks
McLean, VA
United States of America

Email: xufeng.liu.ietf@gmail.com

Network Working Group
Internet-Draft
Intended status: Experimental
Expires: August 25, 2021

H. Chen
M. McBride
Futurewei
A. Wang
China Telecom
G. Mishra
Verizon Inc.
Y. Liu
China Mobile
Y. Fan
Casa Systems
L. Liu
Fujitsu
X. Liu
Volta Networks
February 21, 2021

BIER Fast ReRoute
draft-chen-bier-frr-02

Abstract

This document describes a mechanism for fast re-route (FRR) protection against the failure of a node or link in the core of a "Bit Index Explicit Replication" (BIER) domain. It does not have any per-flow state in the core. For a multicast packet to traverse a node in the domain, when the node fails, its upstream hop as a PLR reroutes the packet around the failed node once it detects the failure.

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] [RFC8174] when, and only when, they appear in all capitals, as shown here.

Status of This Memo

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

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

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

This Internet-Draft will expire on August 25, 2021.

Copyright Notice

Copyright (c) 2021 IETF Trust and the persons identified as the document authors. All rights reserved.

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

Table of Contents

1. Introduction	3
1.1. Terminology	4
2. BIER FRR Solution	5
2.1. Overview of BIER forwarding	5
2.2. FRR Bit Index Routing Tables	6
2.3. FRR Bit Index Forwarding Tables	7
2.4. Updated Forwarding Procedure	7
2.5. Switching between FRR and Normal Forwarding	8
3. Example Application of BIER FRR	8
3.1. Example BIER Topology	9
3.2. BIRT and BIFT on a BFR	9
3.3. FRR-BIRTs and FRR-BIFTs on a BFR	11
3.4. Forwarding using FRR-BIFT	13
4. Security Considerations	14
5. IANA Considerations	14
6. Acknowledgements	15
7. References	15
7.1. Normative References	15
7.2. Informative References	16
Authors' Addresses	17

1. Introduction

[RFC8279] specifies "Bit Index Explicit Replication" (BIER). It provides optimal forwarding of multicast data packets through a "multicast/BIER domain". It does not require the use of a protocol for explicitly building multicast distribution trees, and it does not require intermediate nodes to maintain any per-flow state.

[I-D.merling-bier-frr] proposes a tunnel-based fast re-route (FRR) method for protecting a node or link in the core of a BIER domain, which is called tunnel-based BIER-FRR. It tunnels BIER packets around the failure to BIER nodes downstream in multicast distribution trees. For a (next hop) node failure, it tunnels BIER packets to the next next hop nodes (NNHs). The BIFT in every BFR is enhanced to have two forwarding entries for every BFER. One is the primary forwarding entry with primary NH such as BFR neighbor and primary bit mask, and the other is the backup forwarding entry with backup NH such as NNH and backup bit mask. Using one BIFT in a BFR for both normal and backup forwarding will save memory.

In normal operations, the primary forwarding entries are used to forward BIER packets. When a failure such as a node failure happens, the backup forwarding entry corresponding to the failure and the other primary forwarding entries are used to forward BIER packets. In the BIFT, the primary bit mask in every primary forwarding entry is computed before the failure. After the failure, the primary bit mask needs to be recomputed from the changed topology. Before the primary bit mask is recomputed and updated, some of BIER packets may be forwarded incorrectly.

This document describes a mechanism for fast re-route (FRR) protection against the failure of a node or link in the core of a BIER domain, which resolves the above issue. It is based on LFA, which is called LFA-based BIER-FRR. On a BFR, there is a FRR BIFT for each of its neighbors, which has considered the neighbor failure. There is one forwarding entry for every BFER in any BIFT, including normal BIFT and FRR BIFT. This may use more memory.

In normal operations, the normal BIFT is used to forward BIER packets. When a neighbor fails, the BFR as PLR uses the FRR BIFT for the neighbor to forward BIER packets. For a BIER packet to traverse the BFR and the failed neighbor, the BFR reroutes the packet around the failed neighbor using the FRR BIFT for the neighbor. For a BIER packet to traverse the BFR and any other neighbors, the BFR forwards the packet to its expected next hop neighbors using the forwarding entries with these BFR neighbors in the FRR BIFT.

1.1. Terminology

BFR: Bit-Forwarding Router.

BFIR: Bit-Forwarding Ingress Router.

BFER: Bit-Forwarding Egress Router.

BFR-id: BFR Identifier. It is a number in the range [1,65535].

BFR-NBR: BFR Neighbor.

F-BM: Forwarding Bit Mask.

BFR-prefix: An IP address (either IPv4 or IPv6) of a BFR.

BIRT: Bit Index Routing Table. It is a table that maps from the BFR-id (in a particular sub-domain) of a BFER to the BFR-prefix of that BFER, and to the BFR-NBR on the path to that BFER.

BIFT: Bit Index Forwarding Table.

FRR: Fast Re-Route.

PLR: Point of Local Repair.

LFA: Loop-Free Alternate.

RLFA: Remote LFA.

DLFA: Remote LFA with Directed forwarding.

IGP: Interior Gateway Protocol.

LSDB: Link State DataBase.

SPF: Shortest Path First.

SPT: Shortest Path Tree.

SPT-old(R): The SPT rooted at node R using LSDB before X fails (i.e., old LSDB).

SPT-new(R, X): The SPT rooted at node R using LSDB without X after X fails (i.e., new LSDB).

P-Space $P(R,X)$: The set of nodes that are reachable from R without going through X. In other words, it is the set of nodes that are not downstream of X in SPT-old(R).

Extended P-Space $P'(R,X)$: The set of nodes that are reachable from R or a neighbor of R, without going through X.

Q-Space $Q(D,X)$: The set of nodes that do not use X to reach destination D using the old LSDB.

PQ node(R,X): A member of both the P-Space $P(R, X)$ (or the extended P-Space $P'(R, X)$) and the Q-Space (D, X) .

2. BIER FRR Solution

A Bit-Forwarding Router (BFR) in a BIER sub-domain builds and maintains a "FRR Bit Index Routing Table" (FRR-BIRT) for each of its BFR Neighbors (BFR-NBRs) to provide BIER-FRR. The BFR builds each FRR-BIRT based on a BIRT defined in [RFC8279]. A "FRR Bit Index Forwarding Table" (FRR-BIFT) is derived from a FRR-BIRT in the same way as a BIFT is derived from a BIRT, which is defined in [RFC8279].

The forwarding procedure defined in [RFC8279] is enhanced/updated for FRR-BIFTs. Once the BFR as a PLR detects the failure of its BFR-NBR X, it uses the FRR-BIFT for X to forward packets with BIER headers to get around failed X according to the updated/enhanced forwarding procedure.

2.1. Overview of BIER forwarding

This section briefs the BIRT, BIFT and forwarding procedure defined in [RFC8279].

There is a "Bit Index Routing Table" (BIRT) for a BIER sub-domain on a BFR. The BIRT maps the BFR Identifier (BFR-id) (in the sub-domain) of a Bit-Forwarding Egress Router (BFER) to the BFR-prefix of that BFER, and to the BFR-NBR on the shortest path to that BFER. In other words, the BIRT has a route or say a next hop (i.e., BFR-NBR on the path) to every BFER.

From the BIRT on the BFR, a "Bit Index Forwarding Table" (BIFT) is derived. In addition to having a route to a BFER in each row of the BIFT which is the same as the BIRT, it has a "Forwarding Bit Mask" (F-BM) in its each row. For the rows in the BIRT that have the same SI and the same BFR-NBR, the F-BM for each of these rows in the BIFT is the logical OR of the BitStrings of these rows.

This BIFT is programmed into the data plane and used to forward a packet with a BIER header. The header contains SI, BitString, BitStringLength, and sub-domain.

When a BFR receives a packet, for each BFER *k* (from the rightmost to the leftmost) represented in the SI and BitString of the packet, if BFER *k* is the BFR itself, the BFR copies the packet, sends the copy to the multicast flow overlay and clears bit *k* in the original packet; otherwise the BFR finds the row (i.e., forwarding entry) in the BIFT for the sub-domain using the SI and BitString as the key or say index, and then copies, updates and forwards the packet to the BFR-NBR (i.e., the next hop) indicated by the row (i.e., forwarding entry).

After copying the packet and before forwarding it to the BFR-NBR, the packet's BitString is updated by ANDing it with the F-BM in the forwarding entry (i.e., `PacketCopy->BitString &= F-BM`).

After forwarding the updated packet to a BFR-NBR and before forwarding the original packet to another BFR-NBR, the original packet's BitString is changed by ANDing it with the INVERSE of the F-BM (i.e., `Packet->BitString &= ~F-BM`).

2.2. FRR Bit Index Routing Tables

Each BFR in a BIER sub-domain builds and maintains a number of "FRR Bit Index Routing Tables" (FRR-BIRTs). There is a FRR-BIRT for each BFR-NBR of the BFR. The BFR builds each FRR-BIRT based on its BIRT. It has the same format as the BIRT.

The FRR-BIRT for BFR-NBR *X* of the BFR considers the failure of *X* and maps the BFR-id (in the sub-domain) of a BFER to the BFR-prefix of that BFER, and to BFR-NBR *N* on the path to that BFER. In other words, the FRR-BIRT has a route or say a next hop (i.e., BFR-NBR *N* on the path, where *N* is not *X*) to every BFER when BFR-NBR *X* fails.

The BFR may build the FRR-BIRT for BFR-NBR *X* by copying its BIRT to the FRR-BIRT first, and then change the next hop with value BFR-NBR *X* in the FRR-BIRT to a backup next hop (BNH) to protect against the failure of *X*. In other words, for the BFR-id of a BFER in the FRR-BIRT for BFR-NBR *X*, if the next hop BFR-NBR on the path to the BFER is *X*, it is changed to a BNH when there is a BNH on a backup path to the BFER without going through *X* and the link from the BFR to *X*.

If there is not any BNH to a BFER to protect against the failure of *X*, the next hop BFR-NBR *X* to the BFER in the FRR-BIRT for BFR-NBR *X* is changed to NULL. For a multicast packet having the BFER as one of its destinations, if the next hop BFR-NBR to the BFER is NULL, the

BFR does not send the packet to the next hop BFR-NBR NULL but drops it when X fails.

Note: In another option, the next hop BFR-NBR X to the BFER in the FRR-BIRT for BFR-NBR X keeps unchanged when there is not any BNH to the BFER to protect against the failure of X. In this case, for a multicast packet having the BFER as one of its destinations, the BFR sends the packet to X when X fails.

In one implementation, the BNH is the Loop-Free Node-Protecting Alternate defined in [RFC5286] to protect against the failure of X and link from the BFR to X. In another implementation, the BNH is the virtual Loop-Free Alternate (LFA), i.e., PQ node, defined in [RFC7490]. In a special case, a PQ node is a Loop-Free Node-Protecting Alternate defined in [RFC5286].

2.3. FRR Bit Index Forwarding Tables

From each FRR-BIRT on the BFR, a "FRR Bit Index Forwarding Table" (FRR-BIFT) is derived. In addition to having a route to a BFER in each row of the FRR-BIFT which is the same as the FRR-BIRT, it has a "Forwarding Bit Mask" (F-BM) in its each row. For the rows in the FRR-BIRT that have the same SI and the same BFR-NBR, the F-BM for each of these rows in the FRR-BIFT is the logical OR of the BitStrings of these rows.

This FRR-BIFT is programmed into the data plane and is not used to forward any packet in normal operations. It is activated to forward a packet with a BIER header once the BFR detects the failure of BFR-NBR. The header contains SI, BitString, BitStringLength, and sub-domain.

2.4. Updated Forwarding Procedure

The forwarding procedure defined in [RFC8279] is updated/enhanced for a FRR-BIFT to consider the case where the next hop BFR-NBR to a BFER is NULL. For a multicast packet with the BitString indicating a BFER as one of its destinations, the updated forwarding procedure checks whether the next hop BFR-NBR to the BFER in the FRR-BIFT is NULL. If it is NULL, the procedure will not send the packet to this next hop BFR-NBR NULL but drop the packet.

The updated procedure is described in Figure 1. It is used with a FRR-BIFT for BFR-NBR X on a BFR to forward multicast packets when X fails. It can also be used with a BIFT on the BFR to forward multicast packets in normal operations.

```
Packet = the packet received by BFR;
FOR each BFER k (from the rightmost in Packet's BitString) {
  IF BFER k is the BFR itself {
    copies Packet, sends the copy to the multicast
    flow overlay and clears bit k in Packet's BitString
  } else {
    finds the row in the FRR-BIFT for the sub-domain using
    Packet's SI and BitString as the key/index
    IF BFR-NBR in the row is not NULL {
      Copies Packet, updates the copy's BitString by ANDing
      it with F-BM in the row, sends updated copy to BFR-NBR
    } // BFR-NBR == NULL, not sent Packet to BFR-NBR
    updates Packet's BitString by ANDing it with the INVERSE
    of the F-BM in the row
  }
}
```

Figure 1: Updated Forwarding Procedure

2.5. Switching between FRR and Normal Forwarding

The FRR-BIFTs will be pre-computed and installed ready for activation when a failure is detected. Once the BFR detects the failure of its BFR-NBR X, it activates the FRR-BIFT for X to forward packets with BIER headers and de-activates its BIFT. After activation of the FRR-BIFT, it remains in effect until it is no longer required.

In general, when the routing protocol has re-converged on the new topology taking into account the failure of X, the BIRT is re-computed using the updated LSDB and the BIFT is re-derived from the BIRT. Once the BIFT is installed ready for activation, it is activated to forward packets with BIER headers and the FRR-BIFT for X is de-activated.

From the new topology, the BFR computes/re-computes the FRR-BIRT for each BFR-NBR Y of the BFR and the FRR-BIFT for Y is derived/re-derived from the FRR-BIRT for Y. The FRR-BIFT is installed/re-installed ready for activation when Y fails.

3. Example Application of BIER FRR

This section illustrates an example application of BIER FRR on a BFR in a BIER topology in Figure 2.

3.1. Example BIER Topology

An example BIER topology for a BIER sub-domain is shown in Figure 2. It has 8 nodes/BFRs A, B, C, D, E, F, G and H. Each of the links connecting these nodes/BFRs has a cost. The link cost of 1 is default and is not indicated in the figure. The link cost of other value such as 2 is indicated in the figure.

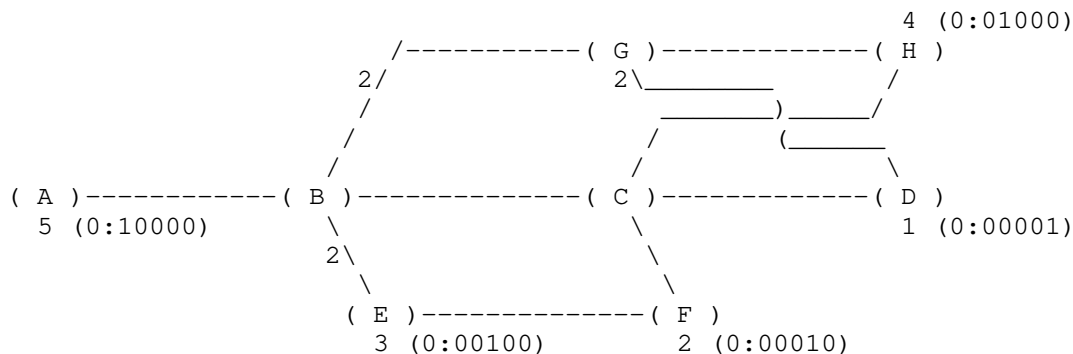


Figure 2: Example BIER Topology

Nodes/BFRs D, F, E, H and A are BFRs and have BFR-ids 1, 2, 3, 4, and 5 respectively. For simplicity, these BFR-ids are represented by (SI:BitString), where SI = 0 and BitString is of 5 bits. BFR-ids 1, 2, 3, 4, and 5 are represented by (0:00001), (0:00010), (0:00100), (0:01000) and (0:10000) respectively.

3.2. BIRT and BIFT on a BFR

Every BFR in a BIER sub-domain/topology builds and maintains a Bit Index Routing Table (BIRT). For the BIER topology in Figure 2, each of 8 nodes/BFRs A, B, C, D, E, F, G and H builds and maintains a BIRT using the LSDB for the topology.

The BIRT built on BFR B (i.e. node B) is shown in Figure 3.

BFR-id (SI:BitString)	BFR-Prefix of Dest BFER	BFR-NBR (Next Hop)
1 (0:00001)	D	C
2 (0:00010)	F	C
3 (0:00100)	E	E
4 (0:01000)	H	C
5 (0:10000)	A	A

Figure 3: BIRT on BFR B

The 1st row in the BIRT indicates that the next hop BFR-NBR on the shortest path to BFER D with BFR-id 1 is BFR C.

The 2nd row in the BIRT indicates that the next hop BFR-NBR on the shortest path to BFER F with BFR-id 2 is BFR C.

The 3rd row in the BIRT indicates that the next hop BFR-NBR on the shortest path to BFER E with BFR-id 3 is BFR E.

The 4-th row in the BIRT indicates that the next hop BFR-NBR on the shortest path to BFER H with BFR-id 4 is BFR C.

The 5-th row in the BIRT indicates that the next hop BFR-NBR on the shortest path to BFER A with BFR-id 5 is BFR A.

From this BIRT on BFR B, a Bit Index Forwarding Table (BIFT) is derived. This BIFT is shown in Figure 4.

The 1st, 2nd and 4-th rows in the BIRT have the same SI = 0 and next hop BFR-NBR = C. The F-BM for each of these three rows in the BIFT is the logical OR of the BitStrings of these rows, which is 01011 (00001 OR 00010 OR 01000 = 01011).

The F-BM for 3rd row in the BIFT is 00100. The F-BM for 5-th row in the BIFT is 10000.

BFR-id (SI:BitString)	F-BM	BFR-NBR (Next Hop)
1 (0:00001)	01011	C
2 (0:00010)	01011	C
3 (0:00100)	00100	E
4 (0:01000)	01011	C
5 (0:10000)	10000	A

Figure 4: BIFT on BFR B

3.3. FRR-BIRTs and FRR-BIFTs on a BFR

Every BFR in a BIER sub-domain/topology builds and maintains a number of FRR Bit Index Routing Tables (FRR-BIRTs). For the BIER topology in Figure 2, each of 8 nodes/BFRs A, B, C, D, E, F, G and H builds and maintains a number of FRR-BIRTs using the LSDB for the topology for its every BFR-NBR.

For example, BFR B (i.e., node B) in the BIER topology builds and maintains four FRR-BIRTs for its four BFR-NBRs (BFR C, BFR E, BFR A and BFR G) respectively. The FRR-BIRT for BFR C built by BFR B is shown in Figure 5.

BFR-id (SI:BitString)	BFR-Prefix of Dest BFER	BFR-NBR (Next Hop)
1 (0:00001)	D	G
2 (0:00010)	F	E
3 (0:00100)	E	E
4 (0:01000)	H	G
5 (0:10000)	A	A

Figure 5: FRR BIRT for BFR C on BFR B

The 1st row in the FRR-BIRT indicates that the next hop BFR-NBR on the path to BFER D with BFR-id 1 is BFR G. G is the Loop-Free Node-Protecting Alternate defined in [RFC5286] to protect against the failure of C and link from B to C.

The 2nd row in the FRR-BIRT indicates that the next hop BFR-NBR on the path to BFER F with BFR-id 2 is BFR E. E is the Loop-Free Node-Protecting Alternate defined in [RFC5286] to protect against the failure of C and link from B to C.

The 3rd row in the FRR-BIRT indicates that the next hop BFR-NBR on the path to BFER E with BFR-id 3 is BFR E.

The 4-th row in the FRR-BIRT indicates that the next hop BFR-NBR on the path to BFER H with BFR-id 4 is BFR G. G is the Loop-Free Node-Protecting Alternate defined in [RFC5286] to protect against the failure of C and link from B to C.

The 5-th row in the FRR-BIRT indicates that the next hop BFR-NBR on the path to BFER A with BFR-id 5 is BFR A.

From this FRR-BIRT for BFR C on BFR B, a FRR Bit Index Forwarding Table (FRR-BIFT) is derived. This FRR-BIFT for BFR C is shown in Figure 6.

The 1st and 4-th rows in the FRR-BIRT have the same SI = 0 and next hop BFR-NBR = G. The F-BM for each of these two rows in the FRR-BIFT is the logical OR of the BitStrings of these rows, which is 01001 (00001 OR 01000 = 01001).

BFR-id (SI:BitString)	F-BM	BFR-NBR (Next Hop)
1 (0:00001)	01001	G
2 (0:00010)	00110	E
3 (0:00100)	00110	E
4 (0:01000)	01001	G
5 (0:10000)	10000	A

Figure 6: FRR BIFT for BFR C on BFR B

The 2nd and 3rd rows in the FRR-BIRT have the same SI = 0 and next hop BFR-NBR = E. The F-BM for each of these two rows in the FRR-BIFT is the logical OR of the BitStrings of these rows, which is 00110 (00010 OR 00100 = 00110).

The F-BM for 5-th row in the FRR-BIFT is 10000.

The number of entries in a FRR BIFT is the number of BFERs. Each FRR BIFT on a BFR can be compressed through combining all the entries with the same BFR-BNR and F-BM into one entry. The number of entries in a compressed FRR BIFT is the number of neighbors of the BFR minus one.

For example, the compressed FRR-BIFT for BFR C on BFR B is shown in Figure 7. The number of entries in it is three, which equals the number (four) of neighbors of BFR B minus one.

BFR-id (SI:BitString)	F-BM	BFR-NBR (Next Hop)
1, 4 (0:01001)	01001	G
2, 3 (0:00110)	00110	E
5 (0:10000)	10000	A

Figure 7: Compressed FRR BIFT for BFR C on BFR B

For a BIER packet with a BFR-ID as a destination, the entry containing the BFR-ID is used to forward the packet.

3.4. Forwarding using FRR-BIFT

Suppose that there is a multicast traffic from BFR A as ingress/BFIR to egresses/BFERs D, F, E and H. For every packet of the traffic, after receiving it, BFR A adds a BIER header into the packet and sends the packet with the BIER header to BFR B. The BIER header contains (SI:BitString) = (0:01111) for egresses/BFERs D, F, E and H.

In normal operations, after receiving the packet from BFR A, BFR B copies, updates and sends the packet to BFR C and BFR E using the BIFT on BFR B according to the forwarding procedure defined in [RFC8279].

Once BFR B detects the failure of its BFR-NBR X, after receiving the packet from BFR A, BFR B copies, updates and sends the packet using

the FRR-BIFT for X on BFR B to avoid X and link from B to X according to the forwarding procedure defined in [RFC8279].

For example, once BFR B detects the failure of its BFR-NBR C, after receiving the packet from BFR A, BFR B copies, updates and sends the packet to BFR G and BFR E using the FRR-BIFT for BFR C on BFR B to avoid C and link from B to C.

The packet received by BFR B from BFR A contains (SI:BitString) = (0:01111). The rightmost one bit in BitString is bit 1. For BFER 1 (0:00001) (i.e., BFR D as BFER), BFR B gets the 1st row (i.e., forwarding entry) in the FRR-BIFT for BFR C. The next hop BFR-NBR is G in the row. BFR B copies, updates and forwards the packet to G.

The packet sent to G contains the updated BitString = 01001, which is 01111 & F-BM in the row = 01111 & 01001.

After sending the packet to G, BFR B updates the original packet by ANDing its BitString with the INVERSE of the F-BM in the row. The updated BitString = 00110, which is 01111 & ~F-BM in the row = 01111 & 00110.

For the packet containing BitString = 00110, the rightmost one bit in BitString is bit 2. For BFER 2 (0:00010) (i.e., BFR F as BFER), BFR B gets the 2nd row (i.e., forwarding entry) in the FRR-BIFT for BFR C. The next hop BFR-NBR is E in the row. BFR B copies, updates and forwards the packet to E.

The packet sent to E contains the updated BitString = 00110, which is 00110 & F-BM in the 2nd row = 00110 & 00110.

After sending the packet to E, BFR B updates the original packet by ANDing its BitString with the INVERSE of the F-BM in the 2nd row. The updated BitString = 00000, which is 00110 & ~F-BM in the row = 00110 & 11001.

The updated packet has BitString without any one bit. BFR B finishes forwarding the packet from A to D, F, E and H.

4. Security Considerations

TBD.

5. IANA Considerations

No requirements for IANA.

6. Acknowledgements

The authors would like to thank Jeffrey Zhang, Daniel Merling and Geng Xuesong for their comments to this work.

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>>.
- [RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", RFC 5226, DOI 10.17487/RFC5226, May 2008, <<https://www.rfc-editor.org/info/rfc5226>>.
- [RFC5250] Berger, L., Bryskin, I., Zinin, A., and R. Coltun, "The OSPF Opaque LSA Option", RFC 5250, DOI 10.17487/RFC5250, July 2008, <<https://www.rfc-editor.org/info/rfc5250>>.
- [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>>.
- [RFC5714] Shand, M. and S. Bryant, "IP Fast Reroute Framework", RFC 5714, DOI 10.17487/RFC5714, January 2010, <<https://www.rfc-editor.org/info/rfc5714>>.
- [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>>.
- [RFC7356] Ginsberg, L., Previdi, S., and Y. Yang, "IS-IS Flooding Scope Link State PDUs (LSPs)", RFC 7356, DOI 10.17487/RFC7356, September 2014, <<https://www.rfc-editor.org/info/rfc7356>>.
- [RFC7490] Bryant, S., Filsfils, C., Previdi, S., Shand, M., and N. So, "Remote Loop-Free Alternate (LFA) Fast Reroute (FRR)", RFC 7490, DOI 10.17487/RFC7490, April 2015, <<https://www.rfc-editor.org/info/rfc7490>>.

- [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>>.
- [RFC7770] Lindem, A., Ed., Shen, N., Vasseur, JP., Aggarwal, R., and S. Shaffer, "Extensions to OSPF for Advertising Optional Router Capabilities", RFC 7770, DOI 10.17487/RFC7770, February 2016, <<https://www.rfc-editor.org/info/rfc7770>>.
- [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>>.
- [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>>.

7.2. Informative References

- [I-D.ietf-rtgwg-segment-routing-ti-lfa]
Litkowski, S., Bashandy, A., Filsfils, C., Decraene, B., and D. Voyer, "Topology Independent Fast Reroute using Segment Routing", draft-ietf-rtgwg-segment-routing-ti-lfa-05 (work in progress), November 2020.
- [I-D.ietf-spring-segment-protection-sr-te-paths]
Hegde, S., Bowers, C., Litkowski, S., Xu, X., and F. Xu, "Segment Protection for SR-TE Paths", draft-ietf-spring-segment-protection-sr-te-paths-00 (work in progress), September 2020.
- [I-D.merling-bier-frr]
Merling, D. and M. Menth, "BIER Fast Reroute", draft-merling-bier-frr-00 (work in progress), March 2019.
- [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>>.

Authors' Addresses

Huaimo Chen
Futurewei
Boston, MA
USA

Email: Huaimo.chen@futurewei.com

Mike McBride
Futurewei

Email: michael.mcbride@futurewei.com

Aijun Wang
China Telecom
Beiqijia Town, Changping District
Beijing, 102209
China

Email: wangaj3@chinatelecom.cn

Gyan S. Mishra
Verizon Inc.
13101 Columbia Pike
Silver Spring MD 20904
USA

Phone: 301 502-1347
Email: gyan.s.mishra@verizon.com

Yisong Liu
China Mobile

Email: liuyisong@chinamobile.com

Yanhe Fan
Casa Systems
USA

Email: yfan@casa-systems.com

Lei Liu
Fujitsu

USA

Email: liulei.kddi@gmail.com

Xufeng Liu
Volta Networks

McLean, VA
USA

Email: xufeng.liu.ietf@gmail.com

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: 28 August 2022

H. Chen
M. McBride
Futurewei
A. Wang
China Telecom
G. Mishra
Verizon Inc.
L. Liu
Fujitsu
X. Liu
Volta Networks
24 February 2022

BIER-TE for Broadcast Link
draft-chen-bier-te-lan-03

Abstract

This document describes extensions to "Bit Index Explicit Replication Traffic Engineering" (BIER-TE) for supporting LANs (i.e., broadcast links). For a multicast packet with an explicit point-to-multipoint (P2MP) path traversing LANs, the packet is replicated and forwarded statelessly along the path.

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] [RFC8174] when, and only when, they appear in all capitals, as shown here.

Status of This Memo

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

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

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

This Internet-Draft will expire on 28 August 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
1.1. Terminology	3
2. Example Application of Current BIER-TE with LAN	4
2.1. Example BIER-TE Topology with LAN	4
2.2. BIER-TE BIFT on BFR	5
2.3. Example P2MP Path with LAN	10
3. Improved BIER-TE with LAN	12
3.1. New BP Assignments for LAN	12
3.2. Improved BIER-TE BIFT on BFR	13
3.3. Updated Forwarding Procedure	17
4. Example Application of Improved BIER-TE	18
5. Security Considerations	19
6. IANA Considerations	20
7. Acknowledgements	20
8. References	20
8.1. Normative References	20
8.2. Informative References	21
Authors' Addresses	22

1. Introduction

[I-D.ietf-bier-te-arch] introduces Bit Index Explicit Replication (BIER) Traffic/Tree Engineering (BIER-TE). It is an architecture for per-packet stateless explicit point to multipoint (P2MP) multicast path/tree. A Bit-Forwarding Router (BFR) in a BIER-TE domain has a BIER-TE Bit Index Forwarding Table (BIFT). A BIER-TE BIFT on a BFR comprises a forwarding entry for a BitPosition (BP) assigned to each of the adjacencies of the BFR. If the BP represents a forward connected adjacency, the forwarding entry for the BP forwards the multicast packet with the BP to the directly connected BFR neighbor of the adjacency. If the BP represents a BFER (i.e., egress node) or say a local decap adjacency, the forwarding entry for the BP

decapsulates the multicast packet with the BP and passes a copy of the payload of the packet to the packet's NextProto within the BFR.

In [I-D.ietf-bier-te-arch], for a LAN, the adjacency to each neighboring BFR on the LAN is given a unique BitPosition. The adjacency of this BitPosition is a forward connected adjacency towards the BFR and this BitPosition is populated into the BIFT of all the other BFRs on that LAN. This solution for a LAN does not work in some cases.

For a packet with an explicit point-to-multipoint (P2MP) path, if the path traverses some BFRs/nodes on a LAN, each of these BFRs/nodes on the LAN may receive duplicated packets. Thus some of the egress nodes will receive duplicated packets.

This document proposes a solution for LANs to resolve this issue. For a packet with an explicit P2MP path traversing LANs (i.e., broadcast links), the packet is replicated and forwarded statelessly along the path. Each of the egress nodes of the path will not receive any duplicated packet.

1.1. Terminology

BIER: Bit Index Explicit Replication.

BIER-TE: BIER Traffic Engineering.

BFR: Bit-Forwarding Router.

BFIR: Bit-Forwarding Ingress Router.

BFER: Bit-Forwarding Egress Router.

BFR-id: BFR Identifier. It is a number in the range [1,65535].

BFR-NBR: BFR Neighbor.

BFR-prefix: An IP address (either IPv4 or IPv6) of a BFR.

BIRT: Bit Index Routing Table. It is a table that maps from the BFR-id (in a particular sub-domain) of a BFER to the BFR-prefix of that BFER, and to the BFR-NBR on the path to that BFER.

BIFT: Bit Index Forwarding Table.

IGP: Interior Gateway Protocol.

LSDB: Link State DataBase.

OSPF: Open Shortest Path First.

IS-IS: Intermediate System to Intermediate System.

2. Example Application of Current BIER-TE with LAN

This section illustrates an example application of the current BIER-TE defined in [I-D.ietf-bier-te-arch] to the BIER-TE topology with LAN in Figure 1.

2.1. Example BIER-TE Topology with LAN

An example BIER-TE topology with a LAN for a BIER-TE domain is shown in Figure 1. It has 9 nodes/BFRs A, B, C, D, E, F, G, H and K. Nodes/BFRs D, F, E, H, A and K are BFERs and have local decap adjacency BitPositions (BPs for short) 1, 2, 3, 4, 5 and 6 respectively. For simplicity, these BPs are represented by (SI:BitString), where SI = 0 and BitString is of 8 bits. BPs 1, 2, 3, 4, 5 and 6 are represented by 1 (0:00000001), 2 (0:00000010), 3 (0:00000100), 4 (0:00001000), 5 (0:00010000) and 6 (0:00100000) respectively.

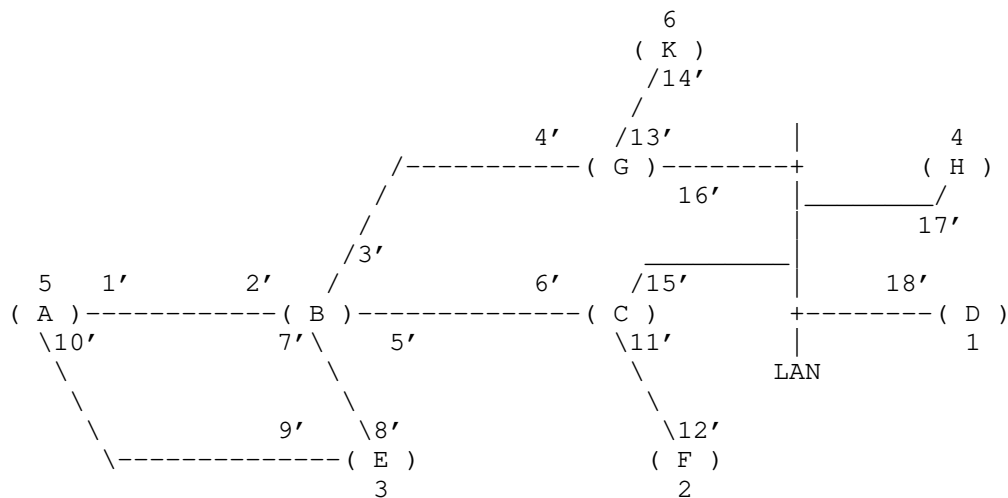


Figure 1: Example BIER-TE Topology with BP to BFR on LAN

The BitPositions for the forward connected adjacencies are represented by i' , where i is from 1 to 18. In one option, they are encoded as $(n+i)$, where n is a power of 2 such as 32768. For simplicity, these BitPositions are represented by (SI:BitString), where $SI = (6 + (i-1)/8)$ and BitString is of 8 bits. BitPositions i'

(i from 1 to 18) are represented by 1' (6:00000001), 2' (6:00000010), 3' (6:00000100), 4' (6:00001000), 5' (6:00010000), 6' (6:00100000), 7' (6:01000000), 8' (6:10000000), 9' (7:00000001), 10' (7:00000010), . . . , 16' (7:10000000), 17' (8:00000001), 18' (8:00000010).

For a link between two nodes X and Y, there are two BitPositions for two forward connected adjacencies. These two forward connected adjacency BitPositions are assigned on nodes X and Y respectively. The BitPosition assigned on X is the forward connected adjacency of Y. The BitPosition assigned on Y is the forward connected adjacency of X.

For example, for the link between nodes B and C in the figure, two forward connected adjacency BitPositions 5' and 6' are assigned to two ends of the link. BitPosition 5' is assigned on node B to B's end of the link. It is the forward connected adjacency of node C. BitPosition 6' is assigned on node C to C's end of the link. It is the forward connected adjacency of node B.

For a LAN (i.e., broadcast link) connecting nodes X1, X2, ..., Xm, there are m BitPositions for m forward connected adjacencies. These m forward connected adjacency BitPositions are assigned on nodes X1, X2, ..., Xm respectively.

For the LAN connecting 4 nodes C, G, H and D in the figure, 4 forward connected adjacency BitPositions 15', 16', 17' and 18' are assigned to C, G, H and D respectively.

2.2. BIER-TE BIFT on BFR

Every BFR in a BIER-TE domain/topology has a BIER-TE BIFT. This section shows the BIER-TE BIFT on every BFR/node of the BIER-TE topology with LAN in Figure 1.

For the BIER-TE topology in Figure 1, each of 9 nodes/BFRs A, B, C, D, E, F, G, H and K has its BIER-TE BIFT for the topology. The BIFT on a BFR comprises a forwarding entry for each of the adjacencies of the BFR.

The BIER-TE BIFT on BFR A (i.e., node A) is shown in Figure 2. There are three adjacencies of A. The 1st one is the forward connected adjacency from A to B (represented by BP 2'); the 2nd is the forward connected adjacency from A to E (represented by BP 9'); the 3rd is the local decap adjacency (represented by BP 5) for BFER (i.e., egress) A. The BIFT on A has three forwarding entries.

Adjacency BP (SI:BitString)	Action	BFR-NBR (Next Hop)
2' (6:00000010)	fw-connected	B
9' (7:00000001)	fw-connected	E
5 (0:00010000)	local-decap	

Figure 2: BIER-TE BIFT on BFR A

The 1st forwarding entry in the BIFT is for BitPosition 2', which is the forward connected adjacency from A to B. For a multicast packet with BitPosition 2', which indicates that the P2MP path in the packet traverses the adjacency from A to B, the forwarding entry forwards the packet to B along the link from A to B.

The 2nd forwarding entry in the BIFT is for BitPosition 9', which is the forward connected adjacency from A to E. For a multicast packet with BitPosition 9', which indicates that the P2MP path in the packet traverses the adjacency from A to E, the forwarding entry forwards the packet to E along the link from A to E.

The 3rd forwarding entry in the BIFT locally decapsulates a multicast packet with BitPosition 5 and passes a copy of the payload of the packet to the packet's NextProto. It is for BitPosition 5, which is the local decap adjacency for BFER (i.e., egress) A. For a multicast packet with BitPosition 5, which indicates that the P2MP path in the packet has node A as one of its destinations (i.e., egress nodes), the forwarding entry decapsulates the packet and passes a copy of the payload of the packet to the packet's NextProto within node A.

The BIER-TE BIFT on BFR B (i.e., node B) is shown in Figure 3. There are four forward connected adjacencies of B. They are the forward connected adjacencies from B to A (represented by BP 1'), B to G (represented by BP 4'), B to C (represented by BP 6') and B to E (represented by BP 8') respectively. The BIFT on B has four forwarding entries for these adjacencies.

Adjacency BP (SI:BitString)	Action	BFR-NBR (Next Hop)
1' (6:00000001)	fw-connected	A
4' (6:00001000)	fw-connected	G
6' (6:00100000)	fw-connected	C
8' (6:10000000)	fw-connected	E

Figure 3: BIER-TE BIFT on BFR B

The 1st forwarding entry in the BIFT is for BitPosition 1', which is the forward connected adjacency from B to A. For a multicast packet with BitPosition 1', which indicates that the P2MP path in the packet traverses the adjacency from B to A, the forwarding entry forwards the packet to A along the link from B to A.

The 2nd forwarding entry in the BIFT is for BitPosition 4', which is the forward connected adjacency from B to G. For a multicast packet with BitPosition 4', which indicates that the P2MP path in the packet traverses the adjacency from B to G, the forwarding entry forwards the packet to G along the link from B to G.

The 3rd forwarding entry in the BIFT is for BitPosition 6', which is the forward connected adjacency from B to C. For a multicast packet with BitPosition 6', which indicates that the P2MP path in the packet traverses the adjacency from B to C, the forwarding entry forwards the packet to C along the link from B to C.

The 4-th forwarding entry in the BIFT is for BitPosition 8', which is the forward connected adjacency from B to E. For a multicast packet with BitPosition 8', which indicates that the P2MP path in the packet traverses the adjacency from B to E, the forwarding entry forwards the packet to E along the link from B to E.

The BIER-TE BIFT on BFR C (i.e., node C) is shown in Figure 4. There are five forward connected adjacencies of C. They are the forward connected adjacencies from C to B (represented by BP 5'), C to F (represented by BP 12'), C to G (represented by BP 14'), C to H (represented by BP 15') and C to D (represented by BP 16') respectively. The BIFT on C has five forwarding entries for these adjacencies.

Adjacency BP (SI:BitString)	Action	BFR-NBR (Next Hop)
5' (6:00010000)	fw-connected	B
12' (7:00001000)	fw-connected	F
16' (7:10000000)	fw-connected	G
17' (8:00000001)	fw-connected	H
18' (8:00000010)	fw-connected	D

Figure 4: BIER-TE BIFT on BFR C

The BIER-TE BIFT on BFR D (i.e., node D) is shown in Figure 5. There are four adjacencies of D. Three of them are the forward connected adjacencies from D to C (represented by BP 13'), D to G (represented by BP 14') and D to H (represented by BP 15') respectively; the other is the local decap adjacency (represented by BP 1) for BFER (i.e., egress) D. The BIFT on D has four forwarding entries for these adjacencies.

Adjacency BP (SI:BitString)	Action	BFR-NBR (Next Hop)
15' (7:01000000)	fw-connected	C
16' (7:10000000)	fw-connected	G
17' (8:00000001)	fw-connected	H
1 (0:00000001)	local-decap	

Figure 5: BIER-TE BIFT on BFR D

The BIER-TE BIFT on BFR E (i.e., node E) is shown in Figure 6. There are three adjacencies of E. Two of them are the forward connected adjacencies from E to B (represented by BP 7') and E to A (represented by BP 10') respectively; the other is the local decap adjacency (represented by BP 3) for BFER (i.e., egress) E. The BIFT on E has three forwarding entries for these adjacencies.

Adjacency BP (SI:BitString)	Action	BFR-NBR (Next Hop)
7' (6:01000000)	fw-connected	B
10' (7:00000010)	fw-connected	A
3 (0:00000100)	local-decap	

Figure 6: BIER-TE BIFT on BFR E

The BIER-TE BIFT on BFR F (i.e., node F) is shown in Figure 7. There are two adjacencies of F. The 1st one is the forward connected adjacencies from F to C (represented by BP 11'); the 2nd is the local decap adjacency (represented by BP 2) for BFER (i.e., egress) F. The BIFT on F has two forwarding entries for these adjacencies.

Adjacency BP (SI:BitString)	Action	BFR-NBR (Next Hop)
11' (7:00000100)	fw-connected	C
2 (0:00000010)	local-decap	

Figure 7: BIER-TE BIFT on BFR F

The BIER-TE BIFT on BFR G (i.e., node G) is shown in Figure 8. There are four forward connected adjacencies of G. They are the adjacencies from G to B (represented by BP 3'), G to C (represented by BP 13'), G to H (represented by BP 15') and G to D (represented by BP 16') respectively. The BIFT on G has four forwarding entries for these adjacencies.

Adjacency BP (SI:BitString)	Action	BFR-NBR (Next Hop)
3' (6:00000100)	fw-connected	B
14' (7:00100000)	fw-connected	K
15' (7:01000000)	fw-connected	C
17' (8:00000001)	fw-connected	H
18' (8:00000010)	fw-connected	D

Figure 8: BIER-TE BIFT on BFR G

The BIER-TE BIFT on BFR H (i.e., node H) is shown in Figure 9. There are four adjacencies of H. Three of them are the forward connected adjacencies from H to C (represented by BP 13'), H to G (represented by BP 14') and H to D (represented by BP 16') respectively; the other is the local decap adjacency (represented by BP 4) for BFER (i.e., egress) H. The BIFT on H has four forwarding entries for these adjacencies.

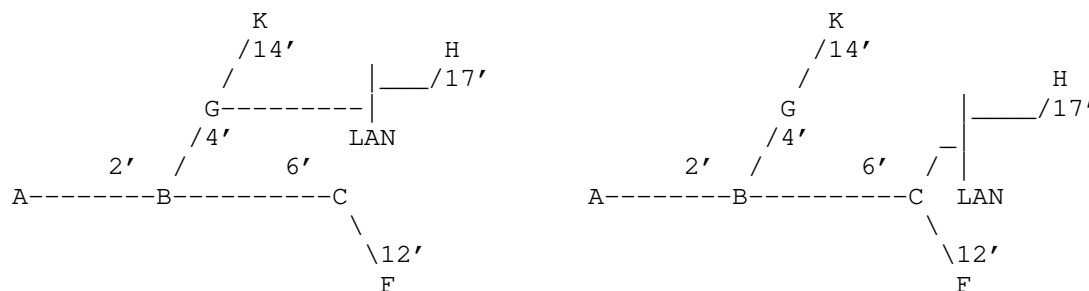
Adjacency BP (SI:BitString)	Action	BFR-NBR (Next Hop)
15' (7:01000000)	fw-connected	C
16' (7:10000000)	fw-connected	G
18' (8:00000010)	fw-connected	D
4 (0:00001000)	local-decap	

Figure 9: BIER-TE BIFT on BFR H

2.3. Example P2MP Path with LAN

This section presents the issue about receiving duplicated packets at BFER H for any explicit P2MP path/tree from BFIR A to BFERs K, H and F in Figure 1 with minimum height of the tree, which is 3 (hops). Any path will go through the LAN in order to reach BFER H.

There are only following explicit P2MP paths from A to K, H and D.



Path 1 from A to K, H and F

Path 2 from A to K, H and F

Figure 10: All explicit paths with height 3 from A to K,H and F

Path 1 and path 2 are represented by {2', 4', 6', 12', 14', 17', 2, 4, 6}. Path 1 traverses the link/adjacency from A to B (indicated by BP 2'), the link/adjacency from B to G (indicated by BP 4'), the link/adjacency from B to C (indicated by BP 6'), the link/adjacency from G to K (indicated by BP 14'), the link/adjacency from G to H (indicated by BP 17') [the link/adjacency from C to H (also indicated by BP 17') for Path 2], and the link/adjacency from C to F (indicated by BP 12'). Path 1 is represented by {2', 4', 6', 12', 14', 17', 2, 4, 6}. Path 2 has the same representation. The packet at A has this representation.

For the packet with the P2MP path, A forwards the packet to B according to the forwarding entry for BP 2' in its BIFT.

After receiving the packet from A, B forwards the packet to G and C according to the forwarding entries for BPs 4' and 6' in B's BIFT respectively. The packet received by G has path {12', 14', 17', 2, 4, 6}. The packet received by C has path {12', 14', 17', 2, 4, 6}.

After receiving the packet from B, G sends a copy of the packet to K according to the forwarding entry for BP 14' in G's BIFT, and another copy of the packet to H according to the forwarding entry for BP 17' in G's BIFT.

After receiving the packet from B, C copies and sends the packet to H and F according to the forwarding entries for BPs 17' and 12' in C's BIFT respectively.

Egress node H of the P2MP path receives the duplicated packets. One packet is from G, and the same copy is from C.

The solution proposed for LANs in this document resolve this issue. For a packet with an explicit P2MP path traversing LANs (i.e., broadcast links), the packet is replicated and forwarded statelessly along the path. Each of the egress nodes of the path will not receive any duplicated packet.

3. Improved BIER-TE with LAN

3.1. New BP Assignments for LAN

For all the nodes/BFRs attached to a LAN (i.e., broadcast link), it is assumed that they are connected a pseudo node. In one implementation, the pseudo node is the Designated Router (DR) of the LAN in OSPF or the Designated Intermediate System (DIS) of the LAN in IS-IS.

For the connection between the pseudo node and each of the nodes/BFRs attached to a LAN, two BPs are assigned to it. One is for the adjacency from the BFR to the pseudo node, the other is for the adjacency from the pseudo node to the BFR.

The adjacency from a BFR to the pseudo node is called a LAN adjacency. The adjacency from the pseudo node to a BFR is a forward connected adjacency.

For example, suppose that the pseudo node for the LAN in Figure 1 is Px. The BP assignments for the LAN (i.e., connections between Px and BFRs C, G, H and D) are illustrated in Figure 11.

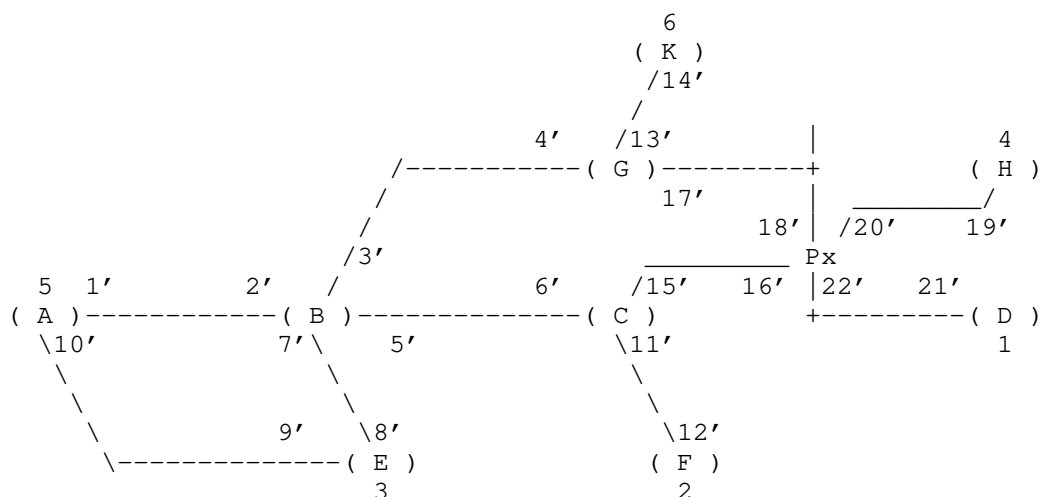


Figure 11: Example BIER-TE Topology with BPs for LAN

The connection/adjacency from Px to C is assigned BP 15', and the connection/adjacency from C to Px is assigned BP 16'.

The connection/adjacency from Px to G is assigned BP 17', and the connection/adjacency from G to Px is assigned BP 18'.

The connection/adjacency from Px to H is assigned BP 19', and the connection/adjacency from H to Px is assigned BP 20'.

The connection/adjacency from Px to D is assigned BP 21', and the connection/adjacency from D to Px is assigned BP 22'.

In an alternative, all the nodes/BFRs attached to a LAN are assumed fully connected each other (i.e., they are fully meshed). For a connection between any two BFRs on the LAN, two forward connected adjacencies are assigned to the two ends of the connection.

For example, there are four BFRs C, G, H and D attached to the LAN in Figure 1. There are six connections among these four BFRs. They are connections between C and G, C and H, C and D, G and H, G and D, H and D. Twelve BPs are needed for these six connections.

In general, for n BFRs attached to a LAN, there are $n*(n-1)/2$ connections among these n BFRs and $n*(n-1)$ BPs are needed for these connections. This may not be scalable. But for this alternative, the BIER-TE BIFT on a BFR needs not to be changed except for considering the full mesh connections among the BFRs attached to a LAN.

3.2. Improved BIER-TE BIFT on BFR

Each BFR in a BIER-TE domain has a BIER-TE BIFT. For a BFR not attached to any LAN, the BIER-TE BIFT on the BFR is the same as before. For a BFR attached to a LAN, its BIER-TE BIFT is changed for considering the LAN.

For example, BFRs C, G, H and D are attached to a LAN in Figure 1. The BIER-TE BIFT on each of these four BFRs is changed for the new BP assignments for the LAN in Figure 11.

For a BFR attached to a LAN, suppose that the pseudo node for the LAN is Px. The improved BIER-TE BIFT on the BFR comprises a forwarding entry for the LAN adjacency from the BFR to Px and a secondary BIFT for Px. The secondary BIFT for Px on the BFR contains a forwarding entry for each of the forward connected adjacencies from Px to the BFRs attached to the LAN except for the adjacency from Px to the BFR.

For example, the improved BIER-TE BIFT on BFR C is illustrated in Figure 12. It comprises the forwarding entry for the LAN adjacency from C to Px (indicated by BP 16') and the secondary BIFT for Px on BFR C. The secondary BIFT contains three forwarding entries for three forward connected adjacencies from Px to G (indicated by 17'), Px to H (indicated by 19') and Px to D (indicated by 21') respectively.

Adjacency BP (SI:BitString)	Action	BFR-NBR (Next Hop)
17' (8:00000001)	fw-connected	G
19' (8:00000100)	fw-connected	H
21' (8:00010000)	fw-connected	D

Secondary BIFT for Px on BFR C

Adjacency BP (SI:BitString)	Action	BFR-NBR (Next Hop)
5' (6:00010000)	fw-connected	B
12' (7:00001000)	fw-connected	F
16' (7:10000000)	lan-connected	Px

Figure 12: Improved BIER-TE BIFT on BFR C

The improved BIER-TE BIFT on BFR G is illustrated in Figure 13. It comprises the forwarding entry for the LAN adjacency from G to Px (indicated by BP 18') and the secondary BIFT for Px on BFR G. The secondary BIFT contains three forwarding entries for three forward connected adjacencies from Px to C (indicated by 15'), Px to H (indicated by 19') and Px to D (indicated by 21') respectively.

Adjacency BP (SI:BitString)	Action	BFR-NBR (Next Hop)
15' (7:01000000)	fw-connected	C
19' (8:00000100)	fw-connected	H
21' (8:00010000)	fw-connected	D
Secondary BIFT for Px on BFR G		
Adjacency BP (SI:BitString)	Action	BFR-NBR (Next Hop)
3' (6:00000100)	fw-connected	B
14' (7:00100000)	fw-connected	K
18' (8:00000010)	lan-connected	Px

Figure 13: Improved BIER-TE BIFT on BFR G

The improved BIER-TE BIFT on BFR H is illustrated in Figure 14. It comprises the forwarding entry for the LAN adjacency from H to Px (indicated by BP 20') and the secondary BIFT for Px on BFR H. The secondary BIFT contains three forwarding entries for three forward connected adjacencies from Px to C (indicated by 15'), Px to G (indicated by 17') and Px to D (indicated by 21') respectively.

Adjacency BP (SI:BitString)	Action	BFR-NBR (Next Hop)
15' (7:01000000)	fw-connected	C
17' (8:00000001)	fw-connected	G
21' (8:00010000)	fw-connected	D
Secondary BIFT for Px on BFR H		
Adjacency BP (SI:BitString)	Action	BFR-NBR (Next Hop)
4 (0:00001000)	local-decap	
20' (8:00001000)	lan-connected	Px

Figure 14: Improved BIER-TE BIFT on BFR H

The improved BIER-TE BIFT on BFR D is illustrated in Figure 15. It comprises the forwarding entry for the LAN adjacency from D to Px (indicated by BP 22') and the secondary BIFT for Px on BFR D. The secondary BIFT contains three forwarding entries for three forward connected adjacencies from Px to C (indicated by 15'), Px to G (indicated by 17') and Px to H (indicated by 19') respectively.

Adjacency BP (SI:BitString)	Action	BFR-NBR (Next Hop)
15' (7:01000000)	fw-connected	C
17' (8:00000001)	fw-connected	G
19' (8:00000100)	fw-connected	H
Secondary BIFT for Px on BFR D		
Adjacency BP (SI:BitString)	Action	BFR-NBR (Next Hop)
1 (0:00000001)	local-decap	
22' (8:00000100)	lan-connected	Px

Figure 15: Improved BIER-TE BIFT on BFR D

3.3. Updated Forwarding Procedure

The forwarding procedure defined in [I-D.ietf-bier-te-arch] is updated/enhanced for using an improved BIER-TE BIFT to support BIER-TE with LAN.

The updated procedure is described in Figure 16. For a multicast packet containing the BitString encoding an explicit P2MP path, if the BP in the BitString is for a LAN adjacency to pseudo node Px for the LAN, the updated forwarding procedure on a BFR sends the packet towards Px's next hop nodes on the P2MP path encoded in the packet.

The procedure on a BFR "sends" (i.e., works as sending) the packet with the BP for the LAN adjacency to Px according to the forwarding entry for the BP in the improved BIER-TE BIFT on the BFR. And then it acts on Px to "send" (i.e., works as sending) the packet to each of the Px's next hop nodes that are on the P2MP path using the secondary BIFT for Px.

It obtains the secondary BIFT for Px on the BFR, clears all the BPs for the adjacencies of the BFR including the adjacency from the BFR to Px, copies and sends the packet to each of the Px's next hop nodes on the P2MP path using the secondary BIFT for Px.

For each Px's next hop node on the P2MP path, which is represented by BP j in the packet's BitString, it gets the forwarding entry for BP j from the secondary BIFT for Px, copies the packet, updates the copy's BitString by clearing all the BPs for Px's adjacencies, and sends the updated copy to the next hop node according to the forwarding entry.

```

Packet = the packet received by BFR;
FOR each BP k (from the rightmost in Packet's BitString) {
  IF BP k is local decap adjacency (or say BP of BFER) {
    Copies Packet, sends the copy to the multicast
    flow overlay and clears bit k in Packet's BitString
  } ELSE IF BP k is forward connected adjacency of the BFR {
    Finds the forwarding entry in the BIER-TE BIFT using BP k,
    Copies Packet, updates the copy's BitString by
    clearing all the BPs for the adjacencies of the BFR,
    and sends the updated copy to BFR-NBR
  } ELSE IF BP k is LAN adjacency to Px {
    Obtains the secondary BIFT for Px,
    Clears all the BPs for the adjacencies of the BFR,
    FOR each BP j (from the rightmost in Packet's BitString) {
      IF BP j is Px's forward connected adjacency {
        Gets the forwarding entry for BP j in the
        secondary BIFT for Px,
        Copies Packet, updates the copy's BitString by
        clearing all the BPs for Px's adjacencies,
        and sends the updated copy to BFR-NBR
      }
    }
  }
}

```

Figure 16: Updated Forwarding Procedure

4. Example Application of Improved BIER-TE

This section illustrates an example application of improved BIER-TE to Figure 1. It shows the forwarding behaviors along an explicit P2MP path in Figure 11 going through the LAN in the figure.

The new BP assignments for the LAN in Figure 1 is shown in Figure 11. The improved BIER-TE BIFT on each of the BFRs attached to the LAN is given in Section 3.2.

The explicit P2MP path traverses the link/adjacency from A to B (indicated by BP 2'), the link/adjacency from B to G (indicated by BP 4') and the link/adjacency from B to C (indicated by BP 6'), the link/adjacency from G to K (indicated by BP 14'), the link/adjacency

from G to Px (indicated by BP 18'), the link/adjacency from C to F (indicated by BP 12'), and the link/adjacency from Px to H (indicated by BP 19'). This path is represented by {2', 4', 6', 12', 14', 18', 19', 2, 4, 6}. The packet at A has this path.

For the packet with the P2MP path, A forwards the packet to B according to the forwarding entry for BP 2' in its BIFT.

After receiving the packet from A, B forwards the packet to G and C according to the forwarding entries for BPs 4' and 6' in B's BIFT respectively. The packet received by G has path {12', 14', 18', 19', 2, 4, 6}. The packet received by C has path {12', 14', 18', 19', 2, 4, 6}.

After receiving the packet from B, G sends a copy of the packet to K according to the forwarding entry for BP 14' in G's improved BIER-TE BIFT and "sends" another copy of the packet to Px according to the forwarding entry for BP 18' in G's improved BIER-TE BIFT. After receiving the packet from G, which has path {12', 19', 2, 4, 6}, Px "sends" the packet to H according to the forwarding entry for BP 19' in the secondary BIFT for Px (a part of G's improved BIER-TE BIFT).

After receiving the packet from G, which has path {12', 19', 2, 4, 6}, K decapsulates the packet and passes a copy of the payload of the packet to the packet's NextProto within node K according to the forwarding entry for BP 6 in K's BIFT.

After receiving the packet from G, which has path {12', 2, 4, 6}, H decapsulates the packet and passes a copy of the payload of the packet to the packet's NextProto within node H according to the forwarding entry for BP 4 in H's improved BIER-TE BIFT.

After receiving the packet from B, which has path {12', 14', 18', 19', 2, 4, 6}, C sends the packet to F according to the forwarding entry for BP 12' in C's improved BIER-TE BIFT.

After receiving the packet from C, which has path {14', 18', 19', 2, 4, 6}, F decapsulates the packet and passes a copy of the payload of the packet to the packet's NextProto within node F according to the forwarding entry for BP 2 in F's BIER-TE BIFT.

Egress node H of the P2MP path does not receive any duplicated packet.

5. Security Considerations

TBD.

6. IANA Considerations

No requirements for IANA.

7. Acknowledgements

The authors would like to thank people for their comments to this work.

8. References

8.1. Normative References

- [I-D.ietf-bier-te-arch]
Eckert, T., Menth, M., and G. Cauchie, "Tree Engineering for Bit Index Explicit Replication (BIER-TE)", Work in Progress, Internet-Draft, draft-ietf-bier-te-arch-12, 28 January 2022, <<https://www.ietf.org/archive/id/draft-ietf-bier-te-arch-12.txt>>.
- [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>>.
- [RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", RFC 5226, DOI 10.17487/RFC5226, May 2008, <<https://www.rfc-editor.org/info/rfc5226>>.
- [RFC5250] Berger, L., Bryskin, I., Zinin, A., and R. Coltun, "The OSPF Opaque LSA Option", RFC 5250, DOI 10.17487/RFC5250, July 2008, <<https://www.rfc-editor.org/info/rfc5250>>.
- [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>>.
- [RFC5714] Shand, M. and S. Bryant, "IP Fast Reroute Framework", RFC 5714, DOI 10.17487/RFC5714, January 2010, <<https://www.rfc-editor.org/info/rfc5714>>.
- [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>>.

- [RFC7356] Ginsberg, L., Previdi, S., and Y. Yang, "IS-IS Flooding Scope Link State PDUs (LSPs)", RFC 7356, DOI 10.17487/RFC7356, September 2014, <<https://www.rfc-editor.org/info/rfc7356>>.
- [RFC7490] Bryant, S., Filsfils, C., Previdi, S., Shand, M., and N. So, "Remote Loop-Free Alternate (LFA) Fast Reroute (FRR)", RFC 7490, DOI 10.17487/RFC7490, April 2015, <<https://www.rfc-editor.org/info/rfc7490>>.
- [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>>.
- [RFC7770] Lindem, A., Ed., Shen, N., Vasseur, JP., Aggarwal, R., and S. Shaffer, "Extensions to OSPF for Advertising Optional Router Capabilities", RFC 7770, DOI 10.17487/RFC7770, February 2016, <<https://www.rfc-editor.org/info/rfc7770>>.
- [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>>.
- [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>>.

8.2. Informative References

- [I-D.eckert-bier-te-frr]
Eckert, T., Cauchie, G., Braun, W., and M. Menth,
"Protection Methods for BIER-TE", Work in Progress,
Internet-Draft, draft-eckert-bier-te-frr-03, 5 March 2018,
<<https://www.ietf.org/archive/id/draft-eckert-bier-te-frr-03.txt>>.
- [I-D.ietf-rtgwg-segment-routing-ti-lfa]
Litkowski, S., Bashandy, A., Filsfils, C., Francois, P.,
Decraene, B., and D. Voyer, "Topology Independent Fast
Reroute using Segment Routing", Work in Progress,

Internet-Draft, draft-ietf-rtgwg-segment-routing-ti-lfa-08, 21 January 2022, <<https://www.ietf.org/archive/id/draft-ietf-rtgwg-segment-routing-ti-lfa-08.txt>>.

- [I-D.ietf-spring-segment-protection-sr-te-paths]
Hegde, S., Bowers, C., Litkowski, S., Xu, X., and F. Xu,
"Segment Protection for SR-TE Paths", Work in Progress,
Internet-Draft, draft-ietf-spring-segment-protection-sr-
te-paths-02, 21 January 2022,
<<https://www.ietf.org/archive/id/draft-ietf-spring-segment-protection-sr-te-paths-02.txt>>.
- [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>>.

Authors' Addresses

Huaimo Chen
Futurewei
Boston, MA,
United States of America
Email: Huaimo.chen@futurewei.com

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

Aijun Wang
China Telecom
Beiqijia Town, Changping District
Beijing
102209
China
Email: wangaj3@chinatelecom.cn

Gyan S. Mishra
Verizon Inc.
13101 Columbia Pike
Silver Spring, MD 20904
United States of America
Phone: 301 502-1347
Email: gyan.s.mishra@verizon.com

Lei Liu
Fujitsu
United States of America
Email: liulei.kddi@gmail.com

Xufeng Liu
Volta Networks
McLean, VA
United States of America
Email: xufeng.liu.ietf@gmail.com

BIER
Internet-Draft
Intended status: Experimental
Expires: 28 April 2022

T. Eckert
Futurewei Technologies USA
25 October 2021

Carrier Grade Minimalist Multicast (CGM2) using Bit Index Explicit
Replication (BIER) with Recursive BitString Structure (RBS) Addresses
draft-eckert-bier-cgm2-rbs-00

Abstract

This memo introduces the architecture of a multicast architecture derived from BIER-TE, which this memo calls Carrier Grade Minimalist Multicast (CGM2). It reduces limitations and complexities of BIER-TE by replacing the representation of the in-packet-header delivery tree of packets through a "flat" BitString of adjacencies with a hierarchical structure of BFR-local BitStrings called the Recursive BitString Structure (RBS) Address.

Benefits of CGM2 with RBS addresses include smaller/fewer BIFT in BFR, less complexity for the network architect and in the CGM2 controller (compared to a BIER-TE controller) and fewer packet copies to reach a larger set of BFER.

The additional cost of forwarding with RBS addresses is a slightly more complex processing of the RBS address in BFR compared to a flat BitString and the novel per-hop rewrite of the RBS address as opposed to bit-reset rewrite in BIER/BIER-TE.

CGM2 can support the traditional deployment model of BIER/BIER-TE with the BIER/BIER-TE domain terminating at service provider PE routers as BFIR/BFER, but it is also the intention of this document to expand CGM2 domains all the way into hosts, and therefore eliminating the need for an IP Multicast flow overlay, further reducing the complexity of Multicast services using CGM2. Note that this is not fully detailed in this version of the document.

This document does not specify an encapsulation for CGM2/RBS addresses. It could use existing encapsulations such as [RFC8296], but also other encapsulations such as IPv6 extension headers.

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 28 April 2022.

Copyright Notice

Copyright (c) 2021 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. Overview	3
1.1. Introduction	3
1.2. Encapsulation Considerations	4
2. CGM2/RBS Architecture	5
3. CGM2/RBS forwarding plane	6
3.1. RBS BIFT	7
3.2. Reference encoding of RBS addresses	8
3.3. RBS Address	8
3.3.1. RecursiveUnit	8
3.3.2. AddressingField	9
4. BIER-RBS Example	9
4.1. BFR B	10
4.2. BFR R	12
4.3. BFR S	13
4.4. BFR C	14
4.5. BFR D	14
4.6. BFR E	15
5. RBS forwarding Pseudocode	16
6. Operational and design considerations (informational)	18
6.1. Comparison with BIER-TE / BIER	18

6.1.1.	Eliminating the need for large BIFT	18
6.1.2.	Reducing number of duplicate packet copies across BFR	19
6.1.3.	BIER-TE forwarding plane complexities	20
6.1.4.	BIER-TE controller complexities	20
6.1.5.	BIER-TE specification complexities	20
6.1.6.	Forwarding plane complexity	21
6.2.	CGM2 / RBS controller considerations	21
6.3.	Analysis of performance gain with CGM2	21
6.4.	Example use case scenarios	21
7.	Acknowledgements	21
8.	Security considerations	21
9.	Changelog	21
10.	References	22
10.1.	Normative References	22
10.2.	Informative References	22
	Author's Address	22

1. Overview

1.1. Introduction

Carrier Grade Minimalist Multicast (CGM2) is an architecture derived from the BIER-TE architecture [I-D.ietf-bier-te-arch] with the following changes/improvements.

CGM2 forwarding is based on the principles of BIER-TE forwarding: It is based on an explicit, in-packet, "source routed" tree indicated through bits for each adjacency that the packet has to traverse. Like in BIER-TE, adjacencies can be L2 to a subnet local neighbor in support of "native" deployment of CGM2 and/or L3, so-called "routed" adjacencies to support incremental or partial deployment of CGM2 as needed.

The address used to replicate packets in the network is not a flat network wide BitString as in BIER-TE, but a hierarchical structure of BitStrings called a Recursive BitString Structure (RBS) Address. The significance of the BitPositions (BP) in each BitString is only local to the BIFT of the router/BFR that is processing this specific BitString.

RBS addressing allows for a more compact representation of a large set of adjacencies especially in the common case of sparse set of receivers in large Service Provider Networks (SP).

CGM2 thereby eliminates the challenges in BIER [RFC8279] and BIER-TE having to send multiple copies of the same packet in large SP networks and the complexities especially for BIER-TE (but also BIER)

to engineer multiple set identifier (SI) and/or sub-domains (SD) BIER-TE topologies for limited size BitStrings (e.g.: 265) to cover large network topologies.

Like BIER-TE, CGM2 is intended to leverage a Controller to minimize the control plane complexity in the network to only a simple unicast routing underlay required only for routed adjacencies.

The controller centric architecture provides most easily any type of required traffic optimization for its multicast traffic due to their need to perform often NP-complete calculations across the whole topology: reservation of bandwidth to support CIR/PIR traffic buffer/latency to support Deterministic Network (DetNet) traffic, cost optimized Steiner trees, failure point disjoint trees for higher resilience including DetNet deterministic services.

CGM2 can be deployed as BIER/BIER-TE are specified today, by encapsulating IP Multicast traffic at Provider Edge (PE) routers, but it is also considered to be highly desirable to extend CGM2 all the way into Multicast Sender/Receivers to eliminate the overhead of an Overlay Control plane for that (legacy) IP Multicast layer and the need to deal with yet another IP multicast group addressing space. In this deployment option Controller signaling extends directly (or indirectly via BFIR) into senders.

1.2. Encapsulation Considerations

This document does not define a specific BIER-RBS encapsulation nor does it preclude that multiple different encapsulations may be beneficial to better support different use-cases or operator/user technology preferences. Instead, it discusses considerations for specific choices.

BIER-RBS can easily re-use [RFC8296] encapsulation. The RBS address is inserted into the [RFC8296] BitString field. The BFR forwarding plane needs to be configured (from Controller or control plane) that the BIFT-id(s) used with RBS addresses are mapped to BIFT and forwarding rules with RBS semantic.

SI/SD fields of [RFC8296] may be used as in BIER-TE, but given that CGM2 is designed (as described in the Overview section) to simplify multicast services, a likely and desirable configuration would be to only use a single BIFT in each BFR for RBS addresses, and mapping these to a single SD and SI 0.

IP Multicast [RFC1112] was defined as an extension of IP [RFC791], reusing the same network header, and IPv6 multicast inherits the same approach. In comparison, [RFC8296] defines BIER encapsulation as a

completely separate (from IP) layer 3 protocol, and duplicates both IP and MPLS header elements into the [RFC8296] header. This not only results in always unused, duplicate header parameters (such as TC vs. DSCP), but it also foregoes the option to use any non-considered IPv6 extension headers with BIER and would require the introduction of a whole new BIER specific socket API into host operating systems if it was to be supported natively in hosts.

Therefore an encapsulation of RBS addresses using an IP and/or IPv6 extension header may be more desirable in otherwise IP and/or IPv6 only deployments, for example when CGM2 is extended into hosts, because it would allow to support CGM2 via existing IP/IPv6 socket APIs as long as they support extension headers, which the most important host stacks do today.

2. CGM2/RBS Architecture

This section describes the basic CGM2 architecture via Figure 1 through its key differences over the BIER-TE architecture.

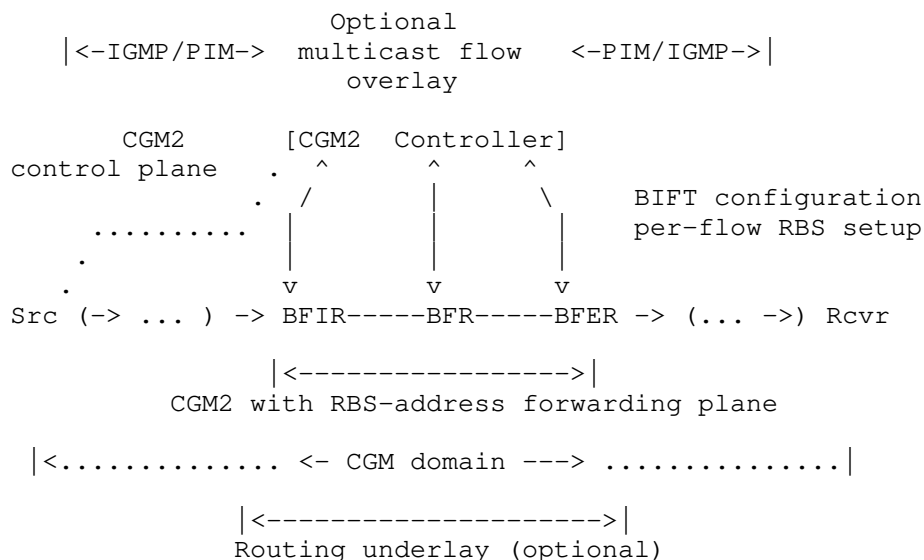


Figure 1: CGM2/RBS Architecture

In the "traditional" option, when deployed with a domain spanning from BFIR to BFER, the CGM2 architecture is very much like the BIER-TE architecture, in which the BIER-TE forwarding rules for (BitString,SI,SD) addresses are replaced by the RBS address forwarding rules.

The CGM2 Controller replaces the BIER-TE controller, populating during network configuration the BIFT, which are very much like BIER-TE BIFT, except that they do not cover a network-wide BP address space, but instead each BFR BIFT only needs as many BP in its BIFT as it has link-local adjacencies, and in partial deployments also additional L3 adjacencies to tunnel across non-CGM capable routers.

Per-flow operations in this "traditional" option is very much as in BIER/BIER-TE, with the CGM2 controller determining the RBS address (instead of the BIER-TE (BitString,SI,SD)) to be imposed as part of the RBS address header (compared to the BIER encapsulation [RFC8296]) on the BFIR.

To eliminate the need for an IP Multicast flow overlays, a CGM2 domain may extend all the way into Sender/Receiver hosts. This is called "end-to-end" deployment model. In that case, the sender host and CGM2 controller collaborate to determine the desired receivers for a packet as well as desired path policy/requirements, the controller indicates to the sender of the packet the necessary RBS address and address of the BFIR, and the Sender imposes an appropriate RBS address header together with a unicast encapsulation towards the BFIR.

CGM2 is also intended so especially simplify controller operations that also instantiate QoS policies for multicast traffic flows, such as bandwidth and latency reservations (e.g.: DetNet). As in BIER-TE, this is orthogonal to the operations of the CGM2/RBS address forwarding operations and will be covered in separate documents.

3. CGM2/RBS forwarding plane

Instead of a (flat) BitString as in BIER-TE that use a network wide shared BP address space for adjacencies across multiple BFR, CGM2 uses a structured address built from so-called RecursiveUnits (RU) that contain BitStrings, each of which is to be parsed by exactly one BFR along the delivery tree of the packet.

The equivalent to a BIER/BIER-TE BitString is therefore called the RecursiveUnit BitString Structure (RBS) Address. Forwarding for CGMP2 is therefore also called RBS forwarding.

3.1. RBS BIFT

RBS BIFT as shown in Figure 2 are, like BIER-TE BIFT, tables that are indexed by BP, containing for each BP an adjacency. The core difference over BIER-TE BIFT is that the BP of the BIFT are all local to the BFR, whereas in BIER-TE, the BP are shared across a BIER-TE domain, each BFR can only use a subset the BP for its own adjacencies, and only in some cases can BP be shared for adjacencies across two (or more) BFR. Because of this difference, most of the complexities of BIER-TE BIFT are not required with BIER-RBS BIFT, see Section 6.1.3.

BP	Recursive	Adjacency
1	1	adjacent BFR
2	0	punt/host
.....	...	
N

Figure 2: RBS BIFT

An RBS BIFT has a configured number of N addressable BP entries. When a BFR receives a packet with an RBS address, it expects that the BitString inside the RBS address that needs to be parsed by the BFR (see Section 3.3 has a length that matches N according to the encapsulation used for the RBS address. Therefore, N MUST support configuration in increments of the supported size of the BitString in the encapsulation of the RBS Address. In the reference encoding (see Section 3.3), the increment for N is 1 (bit). If an encapsulation would call for a byte accurate encoding of the BitString, N would have to be configurable in increments of 8.

BFR MUST support a value of N larger than the maximum number of adjacencies through which RBS forwarding/replication of a single packet is required, such as the number of physical interfaces on BFR that are intended to be deployed as a Provider Core (P) routers.

RBS BIFT introduce a new "Recursive" flag for each BP. These are used for adjacencies to other BFR to indicate that the BFR processing the packet RBS address BitString also has to expect for every BP with the recursive flag set another RU inside the RBS address.

3.2. Reference encoding of RBS addresses

Structure elements of the RBS Address and its components are parameterized according to a specific encapsulation for RBS addresses, such as the total size of the TotalLen field and the unit in which it is counted (see Section 3.3). These parameters are outside the scope of this document. Instead, this document defines example parameters that together form the so called "Reference encoding of RBS addresses". This encoding may or may not be adopted for any particular encapsulation of RBS addresses.

3.3. RBS Address

An RBS address is structured as shown in Figure 3.

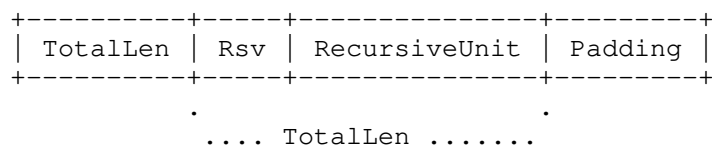


Figure 3: RBS Address

TotalLen counts in some unit, such as bits, nibbles or bytes the length of the RBS Address excluding itself and Padding. For the reference encoding, TotalLen is an 8-bit field that counts the size of the RBS address in bits, permitting for up to 256 bit long RBS addresses.

In case additional, non-recursive flags/fields are determined to be required in the RBS Address, they should be encoded in a field between TotalLen and RecursiveUnit, which is called Rsv. In the reference encoding, this field has a length of 0.

Padding is used to align the RBS address as required by the encapsulation. In the reference encoding, this alignment is to 8 bits (byte boundaries). Therefore, $\text{Padding (bits)} = (8 - \text{TotalLen} \% 8)$.

3.3.1. RecursiveUnit

The RecursiveUnit field is structured as shown in Figure 4.

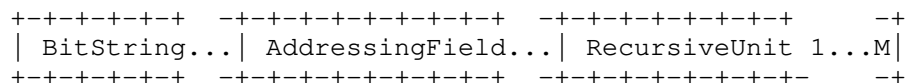


Figure 4: RBS RecursiveUnit

The BitString field indicates the bit positions (BPs) to which the packet is to be replicated using the BIFT of the BFR that is processing the Recursive unit.

For each of M BP set in the BitString of the RecursiveUnit for which the Recursive flag is set in the BIFT of the BFR, the RecursiveUnit contains a RecursiveUnit i , $i=1\dots M$, in order of increasing BP index.

If adjacencies between BFR are not configured as recursive in the BIFT, this recursive extraction does not happen for an adjacency, no RecursiveUnit i has to be encoded for the BP, and BFRs across such adjacencies would have to share the BP of a common BIFT as in BIER-TE. This option is not further discussed in this version of the document.

3.3.2. AddressingField

The AddressingField of an RBS address is structured as shown in Figure 5.

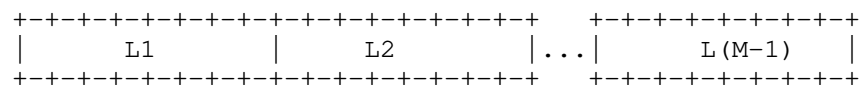


Figure 5: RBS AddressingField

The AddressingField consists of one or more fields L_i , $i=1\dots(M-1)$. L_i is the length of RecursiveUnit i for the i 'th recursive bit set in the BitString preceding it.

In the reference encoding, the lengths are 8-bit fields indicating the length of RecursiveUnits in bits.

The length of the M 'th RecursiveUnit is not explicitly encoded but has to be calculated from TotalLen.

4. BIER-RBS Example

Figure 6 shows an example for RBS forwarding.

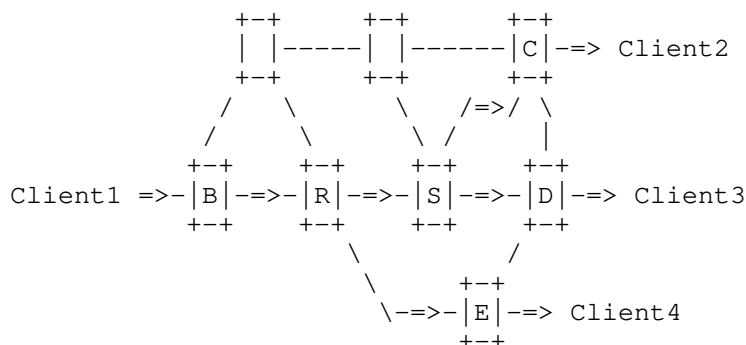


Figure 6: Example Network Topology

A packet from Client1 connected to BFIR B is intended to be replicated to Client2,3,4. The example initially assumes the traditional option of the architecture, in which the imposition of the header for the RBS address happens on BFIR B, for example based on functions of an IP multicast flow overlay.

A controller determines that the packet should be forwarded hop-by-hop across the network as shown in Figure 7.

```

Client 1 ->B(impose BIER-RBS)
=>R(
=> E (dispose BIER-RBS)
=> Client4
=> S(
=>C (dispose BIER-RBS)
=> Client2
=>D (dispose BIER-RBS)
=> Client3
)
)
  
```

Figure 7: Desired example forwarding tree

4.1. BFR B

The 34 bit long (without padding) RBS address shown in Figure 8 is constructed to represent the desired tree from Figure 7 and is imposed at B onto the packet through an appropriate header supporting the reference encoding of RBS addresses.

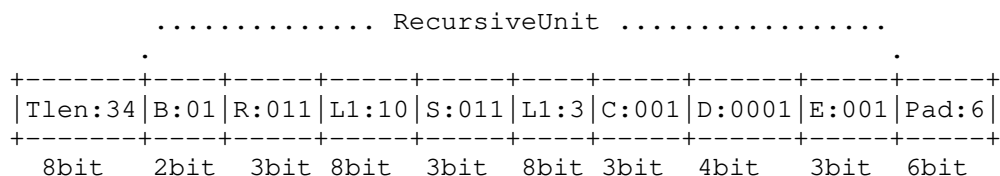


Figure 8: RBS Address imposed at BFIR-B

In Figure 8 and further the illustrations of RBS addresses, BitStrings are preceded by the name of the BFR for whom they are destined and their values are shown as binary with the lowest BP 1 starting on the left. TotalLength (Tlen:), AddressingField (L1:) and Padding (Pad:) fields are shown with decimal values.

RBS forwarding on B examines this address based on its RBS BIFT with N=2 BP entries, which is shown in Figure 9.

BP	Recursive	Adjacency
1	0	client1
2	1	R

Figure 9: BIER-RBS BIFT on B

This results in the parsing of the RBS address as shown in Figure 10, which shows that B does not need (nor can) parse all structural elements, but only those relevant to its own RBS forwarding procedure.

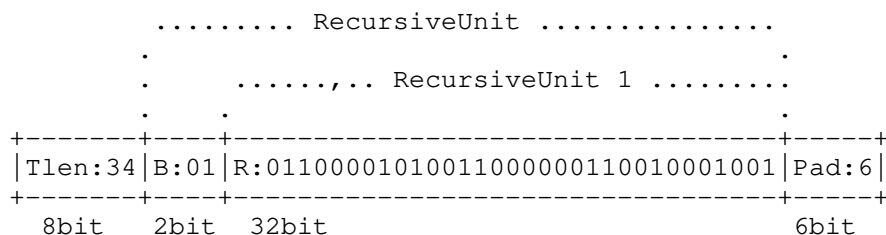


Figure 10: RBS Address as processed by BFIR-B

There is only one BP towards BFR R set in the BitString B:01, so the RecursiveUnit 1 follows directly after the end of the BitString B:01 and it covers the whole Tlen - length of BitString (34 - 2 = 32 bit).

B rewrites the RBS address by replacing the RecursiveUnit with RecursiveUnit 1 and adjusts the Padding to zero bits. The resulting RBS address is shown in Figure 11. It then sends the packet copy with that rewritten RBS address to BFR R.

4.2. BFR R

BFR R receives from BFR B the packet with that RBS address shown in Figure 11.

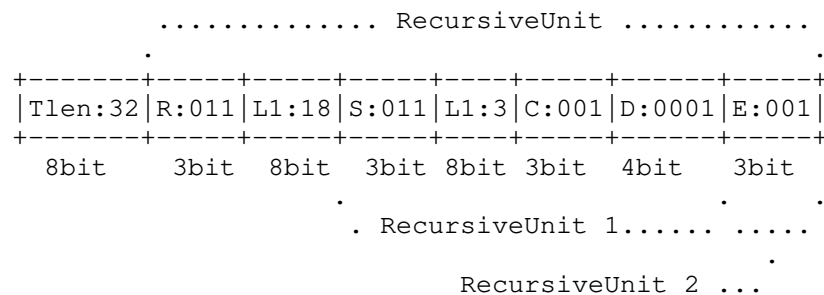


Figure 11: RBS Address processed by BFR-R

BFR R parses the RBS Address as shown in Figure 12 using its RBS BIFT of N=3 BP entries shown in Figure 13.

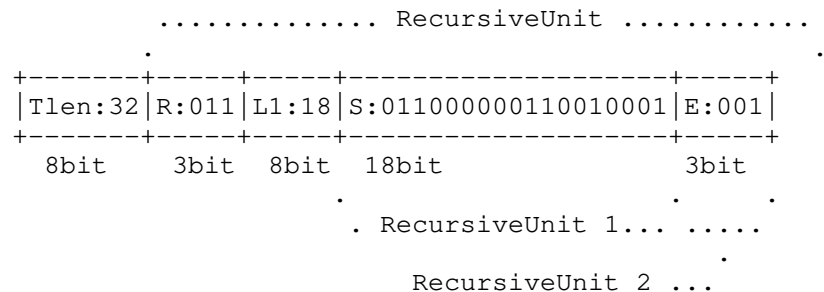


Figure 12: RBS Address processed by BFR-R

Because there are two recursive BP set in the BitString for R, one for BFR S and one for BFR E, one Length field L1 is required in the AddressingField, indicating the length of the RecursiveUnit 1 for BFR S, followed by the remainder of the RBS address being the RecursiveUnit 2 for BFR E.

BP	Recursive	Adjacency
1	1	B
2	1	S
3	1	E

Figure 13: RBS BIFT on BFR R

BFR R accordingly creates one copy for BFR S using RecursiveUnit 1, and only copy for BFR E using RecursiveUnit 2, updating Padding accordingly for each copy.

4.3. BFR S

BFR S receives from BFR B the packet and parses the RBS address as shown in Figure 14 using its RBS BIFT of N=3 BP shown in Figure 15.

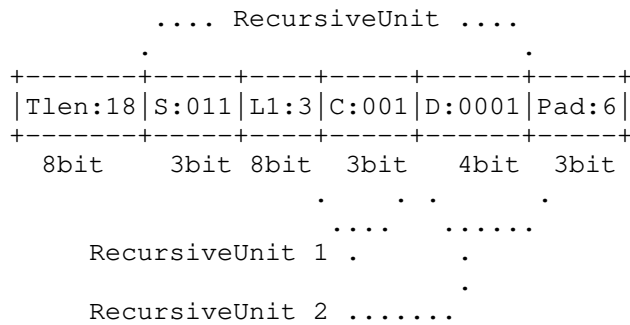


Figure 14: RBS Address processed by BFR-S

BP	Recursive	Adjacency
1	1	R
2	1	C
3	1	D

Figure 15: RBS BIFT on BFR-S

BFR S accordingly sends one packet copy with RecursiveUnit 1 in the RBS address to BFR C and a second packet copy with RecursiveUnit 2 to BFR D.

4.4. BFR C

BFR C receives from BFR S the packet and parses the RBS address according to its N=3 BP entries BIFT (shown in Figure 17) as shown in Figure 16.

```

+-----+-----+-----+
|Tlen:3 |C:001|Pad:5|
+-----+-----+-----+
   8bit    3bit 5bit

```

Figure 16: RBS Address processed by BFR-C

BP	Recursive	Adjacency
1	1	S
2	1	D
3	0	local_decap

Figure 17: RBS BIFT on BFR-C

BFR S accordingly creates one packet copy for BP 3 where the RBS address encapsulation is disposed of, and the packet is ultimately forwarded to Client 2, for example because of an IP multicast payload for which the multicast flow overlay identifies Client 2 as an interested receiver, as in BIER/BIER-TE.

To avoid having to use an IP flow overlay, the BIFT could instead have one BP allocated for every non-RBS destination, in this example BP 3 would then explicitly be allocated for Client 2, and instead of disposing of the RBS address encapsulation, BFR C would impose or rewrite a unicast encapsulation to make the packet become a unicast packet directed to Client 2. This option is not further detailed in this version of the document.

4.5. BFR D

The procedures for processing of the packet on BFR D are very much the same as on BFR C. Figure 18 shows the RBS address at BFR D, Figure 19 shows the N=4 bit RBS BIFT of BFR D.

Tlen:4	D:0001	Pad:4
8bit	4bit	4bit

Figure 18: RBS Address processed by BFR-D

BP	Recursive	Adjacency
1	1	S
2	1	C
3	1	E
4	0	local_decap

Figure 19: RBS BIFT on BFR-D

4.6. BFR E

The procedures for processing of the packet on BFR E are very much the same as on BFR C and D. Figure 20 shows the RBS address at BFR D, Figure 21 shows the N=E bit RBS BIFT of BFR E.

Tlen:3	E:001	Pad:5
8bit	3bit	5bit

Figure 20: RBS Address processed by BFR-E

BP	Recursive	Adjacency
1	1	R
2	1	D
3	0	local_decap

Figure 21: RBS BIFT on BFR-E

5. RBS forwarding Pseudocode

The following example RBS forwarding Pseudocode assumes the reference encoding of bit-accurate length of BitStrings and RecursiveUnits as well as 8-bit long TotalLen and AddressingField Lengths. All packet field addressing and address/offset calculations is therefore bit-accurate instead of byte accurate (which is what most CPU memory access today is).

```

void ForwardRBSPacket (Packet)
{
    RBS = GetPacketMulticastAddr(Packet);
    Total_len = RBS;
    Rsv = Total_len + length(Total_Len);
    BitStringA = Rsv + length(Rsv);
    AddressingField = BitStringA + BIFT.entries;

    // [1] calculate number of recursive bits set in BitString
    CopyBitString(*BitStringA, *RecursiveBits, BIFT.entries);
    And(*RecursiveBits, *BIFTRecursiveBits, BIFT.entries);
    N = CountBits(*RecursiveBits, BIFT.entries);

    // Start of first RecursiveUnit in RBS address
    // After AddressingField array with 8-bit length fields
    RecursiveUnit = AddressingField + (N - 1) * 8;

    RemainLength = *Total_len - length(Rsv)
                  - BIFT.entries;

    Index = GetFirstBitPosition(*BitStringA);
    while (Index) {
        PacketCopy = Copy(Packet);

        if (BIFT.BP[Index].recursive) {
            if(N == 1) {
                RecursiveUnitLength = RemainLength;
            } else {
                RecursiveUnitLength = *AddressingField;
                N--;
                AddressingField += 8;
                RemainLength -= RecursiveUnitLength;
                RemainLength -= 8; // 8 bit of AddressingField
            }
            RewriteRBS(PacketCopy, RecursiveUnit, RecursiveUnitLength);
            SendTo(PacketCopy, BIFT.BP[Index].adjacency);

            RecursiveUnit += RecursiveUnitLength;
        } else {
            DisposeRBSHeader(PacketCopy);
            SendTo(PacketCopy, BIFT.BP[Index].adjacency);
        }
        Index = GetNextBitPosition(*BitStringA, Index);
    }
}

```

Figure 22: RBS address forwarding Pseudocode

Explanations for Figure 22.

RBS is the (bit accurate) address of the RBS address in packet header memory. BitStringA is the address of the RBS address BitString in memory. length(Total_Len) and length(Rsv) are the bit length of the two RBS address fields, e.g.: 8 bit and 0 bit for the reference encoding.

The BFR local BIFT has a total number of BIFT.entries addressable BP 1...BIFTentries. The BitString therefore has BIFT.entries bits.

BIFT.RecursiveBits is a BitString pre-filled by the control plane with all the BP with the recursive flag set. This is constructed from the Recursive flag setting of the BP of the BIFT. The code starting at [1] therefore counts the number of recursive BP in the packets BitString.

Because the AddressingField does not have an entry for the last (or only) RecursiveUnit, its length has to be calculated by taking TotalLen into account.

RewriteRBS needs to replace RBS address with the RecursiveUnit address, keeping only Rsv, recalculating TotalLen and adding appropriate Padding.

For non-recursive BP, the Pseudocode assumes disposition of the RBSheader. This is not strictly necessary but non-disposing cases are outside of scope of this version of the document.

6. Operational and design considerations (informational)

6.1. Comparison with BIER-TE / BIER

This section discusses informationally, how and where CGM2 can avoid different complexities of BIER/BIER-TE, and where it introduces new complexities.

6.1.1. Eliminating the need for large BIFT

In a BIER domain with M BFER, every BFR requires M BIFT entries. If the supported BSL is N and $M > 2^N$, then $S = (M / 2^N)$ set indices (SI) are required, and S copies of the packet have to be sent by the BFIR to reach all targeted BFER.

In CGM2, the number of BIFT entries does not need to scale with the number of BFER or paths through the network, but can be limited to only the number of L2 adjacencies of the BFR. Therefore CGM2 requires minimum state maintenance on each BFR, and multiple SI are not required.

6.1.2. Reducing number of duplicate packet copies across BFR

If the total size of an RBS encoded delivery tree is larger than a supported maximum RBS header size, then the CGM2 controller simply needs to divide the tree into multiple subtrees, each only addressing a part of the BFER (leaves) of the target tree and pruning any unnecessary branches.

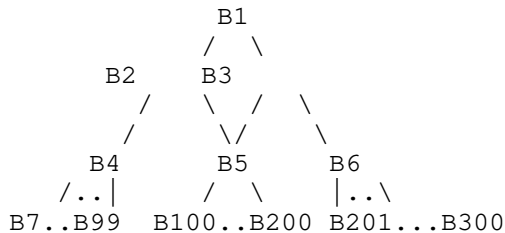


Figure 23: Simple Topology Example

Consider the simple topology in Figure 23 and a multicast packet that needs to reach all BFER B7...B300. Assume that the desired maximum RBM header size is such that a RBS address size of ≤ 256 bits is desired. The CGM2 controller could create an RBS address $B1 \Rightarrow B2 \Rightarrow B4 \Rightarrow (B7..B99)$, for a first packet, an RBS address $B1 \Rightarrow B3 \Rightarrow B5 \Rightarrow (B100..B200)$ for a second packet and a third RBS address $B1 \Rightarrow B3 \Rightarrow B6 \Rightarrow B201...B300$.

The elimination of larger BIFT state in BFR through multiple SI in BIER/BIER-TE does come at the expense of replicating initial hops of a tree in RBS addresses, such as in the example the encoding of $B1 \Rightarrow B3$ in the example.

Consider that the assignment of BFIR-ids with BIER in the above example is not carefully engineered. It is then easily possible that the BFR-ids for B7..B99 are not sequentially, but split over a larger BFIR-id space. If the same is true for all BFER, then it is possible that each of the three BFR B4, B5 and B6 has attached BFER from three different SI and one may need to send for example three multiple packets to B7 to address all BFER B7..B99 or to B5 to address all B100..B200 or B6 to address all B201...B300. These unnecessary duplicate packets across B4, B5 or B6 are because of the addressing principle in BIER and are not necessary in CGM2, as long as the total length of an RBS address does not require it.

For more analysis, see Section 6.3.

6.1.3. BIER-TE forwarding plane complexities

BIER-TE introduces forwarding plane complexities to allow reducing the BSL required. While all of these could be supported / implemented with CGM2, this document contends that they are not necessary, therefore providing significant overall simplifications.

- * BIER-TE supports multiple adjacencies in a single BIFT Index to allow compressing multiple adjacencies into a single Index for traffic that is known to always require replications to all those adjacencies (such as when flooding TV traffic).
- * BIER-TE support ECMP adjacencies which have to calculate which out of 2 or more possible adjacencies a packet should be forwarded to.
- * BIER-TE supports special Do-Not-Clear (DNC) behavior of adjacencies to permit reuse of such a bit for adjacencies on multiple consecutive BFR. This behavior specifically also raises the risk of looping packets.

6.1.4. BIER-TE controller complexities

BIER-TE introduces BIER-TE controller plane mechanisms that allow to reuse bits of the flat BIER-TE BitStrings across multiple BFR solely to reduce the number of BP required but without introducing additional complexities for the BIER-TE forwarding plane.

- * Shared BP for all Leaf BFR.
- * Shared BP for both Interfaces of p2p links.
- * Shared bits for multi-access subnets (LANs).

These bit-sharing mechanisms are unnecessary and inapplicable to CGM2 because there is no need to share BP across the BIFT of multiple BFR.

6.1.5. BIER-TE specification complexities

The BIER-TE specification distinguishes between forward (link scope) and routed (underlay routed) adjacencies to highlight, explain and emphasize on the ability of BIER-TE to be deployed in an overlay fashion especially also to reduce the necessary BSL, even when all routers in the domain could or do support BIER-TE.

In CGM2, routed adjacencies are considered to be only required in partial deployments to forward across non-CGM2 enabled routers. This specification does therefore not highlight link scope vs. routed adjacencies as core distinct features.

6.1.6. Forwarding plane complexity

CGM2 introduces some more processing calculation steps to extract the BitString that needs to be examined by a BFR from the RBS address. These additional steps are considered to be non-problematic for todays programmable forwarding planes such as P4.

Whereas BIER-TE clears bit on each hops processing, CGM2 rewrites the address on every hop by extracting the recursive unit for the next hop and make it become the packet copies address. This rewrite shortens the RBS address. This hopefully has only the same complexity as (tunnel) encapsulations/decapsulations in existing forwarding planes.

6.2. CGM2 / RBS controller considerations

TBD. Any aspects not covered in Section 6.1.

6.3. Analysis of performance gain with CGM2

TBD: Comparison of number of packets/header sizes required in large real-world operator topology between BIER/BIER-TE and CGM2.

6.4. Example use case scenarios

TBD.

7. Acknowledgements

This work is based on the design published by Sheng Jiang, Xu Bing, Yan Shen, Meng Rui, Wan Junjie and Wang Chuang {jiangsheng|bing.xu|yanshen|mengrui|wanjunjie2|wangchuang}@huawei.com, see [CGM2Design].

8. Security considerations

TBD.

9. Changelog

[RFC-Editor: please remove this section].

This document is written in <https://github.com/cabo/kramdown-rfc2629> markup language. This documents source is maintained at <https://github.com/toerless/bier-cgm2-rbs>, please provide feedback to the appropriate IETF mailing list and submit an Issue to the GitHub.

00 - Initial version from [CGM2Design].

10. References

10.1. Normative References

- [I-D.ietf-bier-te-arch]
Eckert, T., Cauchie, G., and M. Menth, "Tree Engineering for Bit Index Explicit Replication (BIER-TE)", Work in Progress, Internet-Draft, draft-ietf-bier-te-arch-10, 9 July 2021, <<https://www.ietf.org/archive/id/draft-ietf-bier-te-arch-10.txt>>.
- [RFC1112] Deering, S., "Host extensions for IP multicasting", STD 5, RFC 1112, DOI 10.17487/RFC1112, August 1989, <<https://www.rfc-editor.org/info/rfc1112>>.
- [RFC791] Postel, J., "Internet Protocol", STD 5, RFC 791, DOI 10.17487/RFC0791, September 1981, <<https://www.rfc-editor.org/info/rfc791>>.
- [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

- [CGM2Design]
Jiang, S., Xu, B.(., Shen, Y., Rui, M., Junjie, W., and W. Chuang, "Novel Multicast Protocol Proposal Introduction", 10 October 2021, <<https://github.com/BingXu1112/CGMM/blob/main/Novel%20Multicast%20Protocol%20Proposal%20Introduction.pptx>>.

Author's Address

Toerless Eckert
Futurewei Technologies USA
2220 Central Expressway
Santa Clara, CA 95050
United States of America

Email: tte@cs.fau.de

Network Working Group
Internet-Draft
Intended status: Informational
Expires: May 5, 2021

H. Bidgoli, Ed.
J. Kotalwar
Nokia
I. Wijnands
M. Mishra
Cisco System
Z. Zhang
Juniper Networks
E. Leyton
Verizon
November 01, 2020

M-LDP Signaling Through BIER Core
draft-ietf-bier-mlbp-signaling-over-bier-00

Abstract

Consider an end to end Multipoint LDP (mLDP) network, where it is desirable to deploy BIER in portion of this network. It might be desirable to deploy BIER with minimum disruption to the mLDP network or redesign of the network.

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

Status of This Memo

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

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). 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 5, 2021.

Copyright Notice

Copyright (c) 2020 IETF Trust and the persons identified as the document authors. All rights reserved.

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

Table of Contents

1. Introduction	2
2. Conventions used in this document	2
2.1. Definitions	3
3. mLDP Signaling Through BIER domain	4
3.1. Ingress BBR procedure	4
3.1.1. Automatic tLDP session Creation	5
3.1.2. ECMP Method on IBBR	5
3.2. Egress BBR procedure	5
3.2.1. IBBR procedure for arriving upstream assigned label .	6
4. Datapath Forwarding	6
4.1. Datapath traffic flow	6
5. Recursive FEC	6
6. IANA Consideration	7
7. Security Considerations	7
8. Acknowledgments	7
9. Informative References	7
Authors' Addresses	7

1. Introduction

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

2. Conventions used in this document

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

2.1. Definitions

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

BIER:

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

BFR:

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

BFIR:

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

BFER:

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

BBR:

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

IBBR:

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

EBBR:

Egress BIER Boundary Router. The egress router in BIER domain from signaling point of view. It terminates the targeted ldp signaling through BIER domain. It also keeps track of all IBBRs that are part of this P2MP tree

BIFT:

Bit Index Forwarding Table.

BIER sub-domain:

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

BFR-ID.

An optional, unique identifier for a BFR within a BIER sub-domain. All BFERs and BFIRs need to be assigned a BFR-ID.

3. mLDP Signaling Through BIER domain

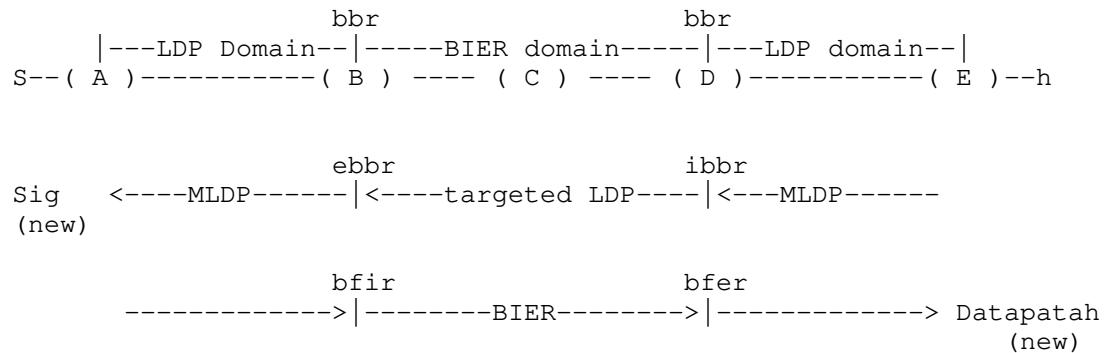


Figure 1: BIER boundry router

As per figure 1, point-to-multipoint (P2MP) and multipoint-to-multipoint (MP2MP) LSPs established via mLDP [RFC6388] can be signaled through a bier domain via Targeted LDP sessions. This procedure is explained in [RFC7060] (Using LDP Multipoint Extension on Targeted LDP Sessions).

This documents provides details and defines some needed procedures.

.

3.1. Ingress BBR procedure

The Ingress BBR (IBBR) is connected to the mLDP domain on downstream and a bier domain on the upstream. To connect the LDP domains via BIER domain, IBBR needs to establish a targeted LDP session with EBBR closest to the root of the P2MP or MP2MP LSP. To do so IBBR will

follow procedures in [RFC7060] in particular the section "6. targeted mLDP with Multicast Tunneling".

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

3.1.1. Automatic tLDP session Creation

tLDP session can be signaled automatically from every IBBR to the appropriate EBBR. When mLDP FEC arrives to IBBR from LDP domain, IBBR can automatically start a tLDP Session to the EBBR closest to the Root node. Both IBBR and EBBR should be in auto-discovery mode and react to the arriving tLDP Signaling packets (i.e. targeted hellos, keep-alives etc...) to establish the session automatically.

The Root node address in the mLDP FEC can be used to find the EBBR. To identify the EBBR same procedures as [RFC7060] section 2.1 can be used or the procedures as explained in the [draft-ietf-bier-pim-signaling] appendix A.

3.1.2. ECMP Method on IBBR

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

3.2. Egress BBR procedure

The Egress BBR (EBBR) is connected to the upstream mLDP domain. The EBBR should accept the tLDP session generated from IBBR. It should assign a unique "upstream assigned label" for each arriving FEC generated by IBBRs.

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

- o The label assigned by EBBR cannot be Implicit Null. This is to ensure that identity of each p2mp and/or mp2mp tunnel in BIER domain is uniquely distinguished.
- o The label can be assigned from a domain-wide Common Block (DCB) [draft-ietf-bess-mvpn-evpn-aggregation-label]
- o The Interface ID TLV, as per [RFC6389] should include a new BIER sub-domain sub-tlv (type TBD)

The EBBR will also generate a new label and FEC toward the ROOT on the LDP domain. The EBBR Should stitch this generate label with the "upstream assigned label" to complete the P2MP or MP2MP LSP.

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

3.2.1. IBBR procedure for arriving upstream assigned label

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

4. Datapath Forwarding

4.1. Datapath traffic flow

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

In the BIER domain normal BIER forwarding procedure will be done, as per [RFC8279]

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

It should be noted that these procedures are also valid if BFIR is the ILER and/or BFER is the ELER as per [RFC7060]

5. Recursive FEC

The above procedures also will work with a recursive FEC [RFC6512]. The root used to determine the EBBR is the outer FECs root. The entire recursive FEC needs to be preserve when it is forwarded via tLDP and the label request.

6. IANA Consideration

1. A new BIER sub-domain sub- tlv for the interface ID TLV to be assigned by IANA

7. Security Considerations

TBD

8. Acknowledgments

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

9. Informative References

- [draft-ietf-bess-mvpn-evpn-aggregation-label]
"Z. Zhang, E. Rosen, W.Lin, Z. Li, I. Wijnands " MVPN/EVPN Tunnel Aggregation with Common Labels"", February 2012.
- [draft-ietf-bier-pim-signaling]
"H.Bidgoli, F. Xu, J. Kotalwar, IJ. Wijnands, M. Mishra, Z. Zhang "PIM Signaling Through BIER Core"", February 2012.
- [RFC6388] "IJ. Wijnands, I. Minei, K. Kompella, B. Thomas "LDP Extensions for P2MP and MP2MP"", November 2011.
- [RFC6389] "R. Aggarwal, JL. Le Roux "MPLS Upstream Label Assignment for LDP"", November 2011.
- [RFC6512] "IJ. Wijnands, E. Rosen, M. Napierala, N. Leymann "Using Multipoint LDP when the backbone has No route to the root"", February 2012.
- [RFC7060] "M. Napierala, E. Rosen, IJ. Wijnands "Using LDP Multipoint Extensions on Targeted LDP Sessions"", November 2013.
- [RFC8279] "IJ. Wijnands, E. Rosen, A. Dolganow, T. Przygienda, S. Aldrin "Multicast using BIER"", April 2018.

Authors' Addresses

Hooman Bidgoli (editor)
Nokia
Ottawa
Canada

Email: hooman.bidgoli@nokia.com

Jayant Kotalwar
Nokia
Mountain View
US

Email: jayant.kotalwar@nokia.com

IJsbrand Wijnands
Cisco System
Diegem
Belgium

Email: ice@cisco.com

Mankamana Mishra
Cisco System
Milpitas
USA

Email: mankamis@cisco.com

Zhaohui Zhang
Juniper Networks
Boston
USA

Email: zzhang@juniper.com

Eddie Leyton
Verizon

Email: Edward.leyton@verizonwireless.com

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: 19 May 2022

T.T.E. Eckert, Ed.
Futurewei
G.C. Cauchie
Bouygues Telecom
M.M. Menth
University of Tuebingen
November 2021

Tree Engineering for Bit Index Explicit Replication (BIER-TE)
draft-ietf-bier-te-arch-11

Abstract

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

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

As it operates on the same per-packet stateless forwarding principles, BIER-TE can also be a good fit to support multicast path steering in Segment Routing (SR) networks.

Status of This Memo

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

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

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

This Internet-Draft will expire on 5 May 2022.

Copyright Notice

Copyright (c) 2021 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. Overview	3
1.1. Requirements Language	5
2. Introduction	5
2.1. Basic Examples	5
2.2. BIER-TE Topology and adjacencies	8
2.3. Relationship to BIER	9
2.4. Accelerated/Hardware forwarding comparison	11
3. Components	11
3.1. The Multicast Flow Overlay	12
3.2. The BIER-TE Control Plane	12
3.2.1. The BIER-TE Controller	13
3.2.1.1. BIER-TE Topology discovery and creation	14
3.2.1.2. Engineered Trees via BitStrings	14
3.2.1.3. Changes in the network topology	15
3.2.1.4. Link/Node Failures and Recovery	15
3.3. The BIER-TE Forwarding Plane	15
3.4. The Routing Underlay	16
3.5. Traffic Engineering Considerations	16
4. BIER-TE Forwarding	17
4.1. The Bit Index Forwarding Table (BIFT)	17
4.2. Adjacency Types	18
4.2.1. Forward Connected	19
4.2.2. Forward Routed	19
4.2.3. ECMP	19
4.2.4. Local Decapsulation	20
4.3. Encapsulation / Co-existence with BIER	20
4.4. BIER-TE Forwarding Pseudocode	21
4.5. Basic BIER-TE Forwarding Example	24
4.6. BFR Requirements for BIER-TE forwarding	26
5. BIER-TE Controller Operational Considerations	27

5.1. Bit position Assignments	27
5.1.1. P2P Links	27
5.1.2. BFER	27
5.1.3. Leaf BFERs	27
5.1.4. LANs	28
5.1.5. Hub and Spoke	29
5.1.6. Rings	29
5.1.7. Equal Cost MultiPath (ECMP)	30
5.1.8. Forward Routed adjacencies	33
5.1.8.1. Reducing bit positions	33
5.1.8.2. Supporting nodes without BIER-TE	34
5.1.9. Reuse of bit positions (without DNC)	34
5.1.10. Summary of BP optimizations	36
5.2. Avoiding duplicates and loops	37
5.2.1. Loops	37
5.2.2. Duplicates	37
5.3. Managing SI, sub-domains and BFR-ids	38
5.3.1. Why SI and sub-domains	38
5.3.2. Assigning bits for the BIER-TE topology	39
5.3.3. Assigning BFR-id with BIER-TE	40
5.3.4. Mapping from BFR to BitStrings with BIER-TE	41
5.3.5. Assigning BFR-ids for BIER-TE	42
5.3.6. Example bit allocations	42
5.3.6.1. With BIER	42
5.3.6.2. With BIER-TE	43
5.3.7. Summary	44
6. BIER-TE and Segment Routing	45
7. Security Considerations	46
8. IANA Considerations	47
9. Acknowledgements	47
10. Change log [RFC Editor: Please remove]	47
11. References	57
11.1. Normative References	57
11.2. Informative References	57
Authors' Addresses	60

1. Overview

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

BIER-TE introduces a new semantic for bit positions (BP) that indicate adjacencies, as opposed to BIER in which BPs indicate Bit-Forwarding Egress Routers (BFER). With BIER-TE, the BIFT of each BFR is only populated with BP that are adjacent to the BFR in the BIER-TE Topology. Other BPs are empty in the BIFT. The BFR replicate and

forwards BIER packets to adjacent BPs that are set in the packet. BPs are normally also cleared upon forwarding to avoid duplicates and loops. This is detailed further below.

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

As it operates on the same per-packet stateless forwarding principles, BIER-TE can also be a good fit to support multicast path steering in Segment Routing (SR) networks.

This document is structured as follows:

- * Section 2 introduces BIER-TE with two reference forwarding examples, followed by an introduction of the new concepts of the BIER-TE (overlay) topology and finally a summary of the relationship between BIER and BIER-TE and a discussion of accelerated hardware forwarding.
- * Section 3 describes the components of the BIER-TE architecture, Flow overlay, BIER-TE layer with the BIER-TE control plane (including the BIER-TE controller) and BIER-TE forwarding plane, and the routing underlay.
- * Section 4 specifies the behavior of the BIER-TE forwarding plane with the different type of adjacencies and possible variations of BIER-TE forwarding pseudocode, and finally the mandatory and optional requirements.
- * Section 5 describes operational considerations for the BIER-TE controller, foremost how the BIER-TE controller can optimize the use of BP by using specific type of BIER-TE adjacencies for different type of topological situations, but also how to assign bits to avoid loops and duplicates (which in BIER-TE does not come for free), and finally how SI, sub-domains and BFR-ids can be managed by a BIER-TE controller, examples and summary.
- * Section 6 concludes the technology specific sections of document by further relating BIER-TE to Segment Routing (SR).

Note that related work, [I-D.ietf-roll-ccast] uses Bloom filters [Bloom70] to represent leaves or edges of the intended delivery tree. Bloom filters in general can support larger trees/topologies with fewer addressing bits than explicit BitStrings, but they introduce

the heuristic risk of false positives and cannot clear bits in the BitString during forwarding to avoid loops. For these reasons, BIER-TE uses explicit BitStrings like BIER. The explicit BitStrings of BIER-TE can also be seen as a special type of Bloom filter, and this is how related work [ICC] describes it.

1.1. Requirements Language

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

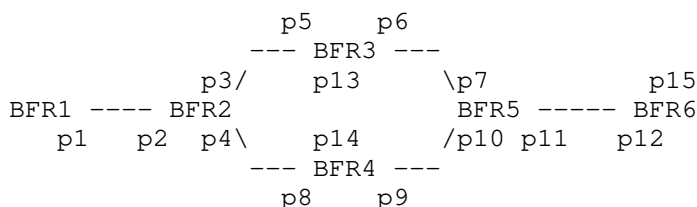
2. Introduction

2.1. Basic Examples

BIER-TE forwarding is best introduced with simple examples.

BIER-TE Topology:

Diagram:



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

```

BFR1:  p1  -> local_decap
       p2  -> forward_connected() to BFR2

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

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

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

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

BFR6:  p11 -> forward_connected() to BFR5
       p15 -> local_decap

```

Figure 1: BIER-TE basic example

Consider the simple network in the above BIER-TE overview example picture with 6 BFRs. p1...p14 are the bit positions (BP) used. All BFRs can act as an ingress BFR (BFIR), BFR1, BFR3, BFR4 and BFR6 can also be egress BFRs (BFERs). Forward_connected() is the name for adjacencies that are representing subnet adjacencies of the network. Local_decap() is the name of the adjacency to decapsulate BIER-TE packets and pass their payload to higher layer processing.

Assume a packet from BFR1 should be sent via BFR4 to BFR6. This requires a BitString (p2,p8,p10,p12,p15). When this packet is examined by BIER-TE on BFR1, the only bit position from the BitString that is also set in the BIFT is p2. This will cause BFR1 to send the only copy of the packet to BFR2. Similarly, BFR2 will forward to BFR4 because of p8, BFR4 to BFR5 because of p10 and BFR5 to BFR6 because of p12. p15 finally makes BFR6 receive and decapsulate the packet.

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

If instead the BitString was (p2,p6,p8,p10,p12,p13,p15), the packet would be copied by BFR5 towards BFR3 because of p6 instead of being copied by BFR2 to BFR3 because of p5 in the prior case. This is showing the ability of the shown BIER-TE Topology to make the traffic pass across any possible path and be replicated where desired.

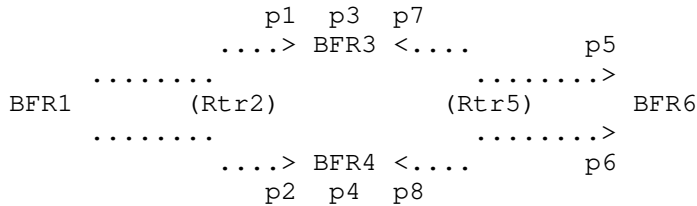
BIER-TE has various options to minimize BP assignments, many of which are based on assumptions about the required multicast traffic paths and bandwidth consumption in the network.

The following picture shows a modified example, in which Rtr2 and Rtr5 are assumed not to support BIER-TE, so traffic has to be unicast encapsulated across them. To emphasize non-L2, but routed/tunneled forwarding of BIER-TE packets, these adjacencies are called "forward_routed". Otherwise there is no difference in their processing over the aforementioned "forward_connected" adjacencies.

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

BIER-TE Topology:

Diagram:



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

```

BFR1:  p1  -> forward_routed() to BFR3
        p2  -> forward_routed() to BFR4

BFR3:  p3  -> local_decap
        p5  -> forward_routed() to BFR6

BFR4:  p4  -> local_decap
        p6  -> forward_routed() to BFR6

BFR6:  p5  -> local_decap
        p6  -> local_decap
        p7  -> forward_routed() to BFR3
        p8  -> forward_routed() to BFR4

```

Figure 2: BIER-TE basic overlay example

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

2.2. BIER-TE Topology and adjacencies

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

The BIER-TE Topology consists of the BIFTs of all the BFR and can also be expressed as a directed graph where the edges are the adjacencies between the BFR labelled with the BP used for the adjacency. Adjacencies are naturally unidirectional. BP can be reused across multiple adjacencies as long as this does not lead to undesired duplicates or loops as explained further down in the text.

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

2.3. Relationship to BIER

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

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

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

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

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

1. In BIER, bits in the BitString of a BIER packet header indicate a BFER and bits in the BIFT indicate the BIER control plane calculated next-hop toward that BFER. In BIER-TE, bits in the BitString of a BIER packet header indicate an adjacency in the BIER-TE topology, and only the BFRs that are the upstream of this adjacency have this bit populated with the adjacency in their BIFT.
2. In BIER, the implied reference option for the core part of the BIER layer control plane are the BIER extensions for distributed routing protocols, such as those standardized in ISIS/OSPF extensions for BIER, [RFC8401] and [RFC8444]. The reference option for the core part of the BIER-TE control plane is the BIER-TE controller. Nevertheless, both BIER and BIER-TE BIFT forwarding plane state could equally be populated by any mechanism.
3. Assuming the reference options for the control plane, BIER-TE replaces in-network autonomous path calculation by explicit paths calculated by the BIER-TE controller.
3. The following elements/functions described in the BIER architecture are not required by the BIER-TE architecture:
 1. BIRTs are not required on BFRs for BIER-TE when using a BIER-TE controller because the controller can directly populate the BIFTs. In BIER, BIRTs are populated by the distributed routing protocol support for BIER, allowing BFRs to populate their BIFTs locally from their BIRTs. Other BIER-TE control plane or management plane options may introduce requirements for BIRTs for BIER-TE BFRs.
 2. The BIER-TE layer forwarding plane does not require BFRs to have a unique BP and therefore also no unique BFR-id. See for example See Section 5.1.3.
 3. Identification of BFRs by the BIER-TE control plane is outside the scope of this specification. Whereas the BIER control plane uses BFR-ids in its BFR to BFR signaling, a BIER-TE controller may choose any form of identification deemed appropriate.
 4. BIER-TE forwarding does not use the BFR-id field of the BIER packet header.
4. Co-existence of BIER and BIER-TE in the same network requires the following:

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

2.4. Accelerated/Hardware forwarding comparison

Forwarding of BIER-TE is designed to easily build/program common forwarding hardware with BIER. The pseudocode in Section 4.4 shows how existing (non-TE) BIER/BIFT forwarding can be modified to support the REQUIRED BIER-TE forwarding functionality, by using BIER BIFT's "Forwarding Bit Mask" (F-BM): Only the clearing of bits to avoid duplicate packets to a BFR's neighbor is skipped in BIER-TE forwarding because it is not necessary and could not be done when using BIER F-BM.

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

3. Components

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

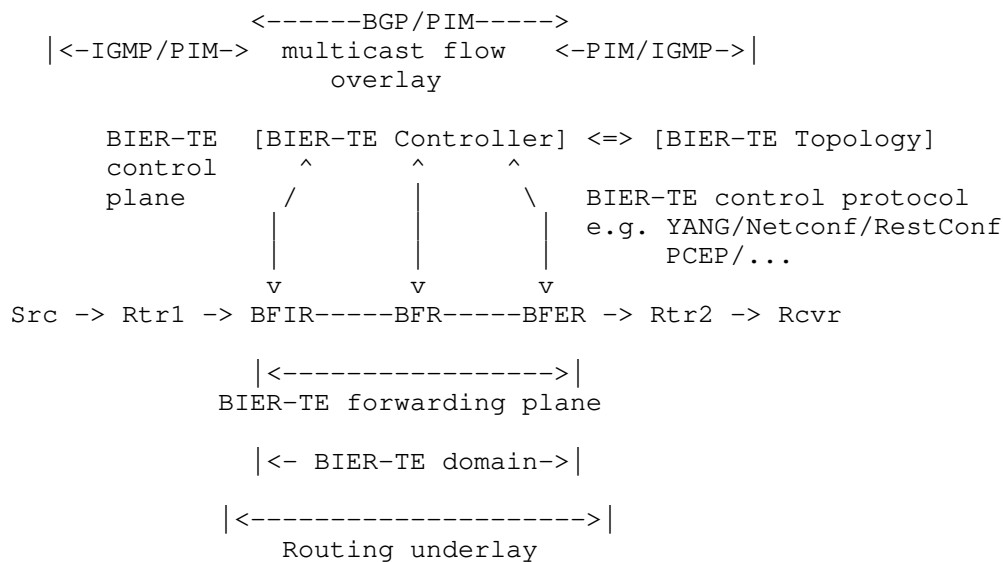


Figure 3: BIER-TE architecture

3.1. The Multicast Flow Overlay

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

3.2. The BIER-TE Control Plane

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

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

1. During initial provisioning of the network and/or during modifications of its topology and/or services: protocols and/or procedures to establish BIER-TE BIFTs:
 1. Determine the desired BIER-TE topology for a BIER-TE sub-domains: the native and/or overlay adjacencies that are assigned to BPs.
 2. Determine the per-BFR BIFT from the BIER-TE topology.

3. Optionally assign BFR-ids to BFIRs for later insertion into BIER-TE headers on BFIRs. Alternatively, bfr-id in BIER packet headers may be managed solely by the flow overlay layer and/or be unused.
 4. Install/update the BIFTs into the BFRs and optionally BFR-ids into BFIRs.
2. During operations of the network: Protocols and/or procedures to support creation/change/removal of overlay flows on BFIRs:
 1. Process the BIER-TE requirements for the multicast overlay flow: BFIR and BFERs of the flow as well as policies for the path selection of the flow.
 2. Determine the BitStrings and optionally Entropy.
 3. Install state on the BFIR to impose the desired BIER packet header(s) for packets of the overlay flow.
 4. Install the necessary state on the BFERs to decapsulate the BIER packet header and properly dispatch its payload.

3.2.1. The BIER-TE Controller

Notwithstanding other options, this architecture describes the BIER control plane as shown in Figure 3 to consists of:

- * A single centralized BIER-TE controller.
- * Data-models and protocols to communicate between controller and BFRs in step 1, such as YANG/Netconf/RestConf.
- * Protocols to communicate between controller and BFIR in step 2, such as BIER-TE extensions for [RFC5440].

The (non-TE) BIER control plane could equally be implemented without any active dynamic components by an operator via CLI on the BFRs. In that case, operator configured local policy on the BFIR would have to determine how to set the appropriate BIER header fields. The BIER-TE control plane could also be decentralized and/or distributed, but this document does not consider any additional protocols and/or procedures which would then be necessary to coordinate its entities to achieve the above described functionality.

3.2.1.1. BIER-TE Topology discovery and creation

Step 1.1 includes network topology discovery and BIER-TE topology creation. The latter describes the process by which a Controller determines which routers are to be configured as BFR and the adjacencies between them.

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

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

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

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

Communications between the BIER-TE Controller and BFRs (beside topology discovery) is ideally via standardized protocols and data-models such as Netconf/RestConf/Yang/PCEP. Vendor-specific CLI on the BFRs is also an option (as in many other SDN solutions lacking definition of standardized data model).

3.2.1.2. Engineered Trees via BitStrings

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

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

3.2.1.3. Changes in the network topology

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

3.2.1.4. Link/Node Failures and Recovery

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

3.3. The BIER-TE Forwarding Plane

The BIER-TE Forwarding Plane constitutes of the following components:

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

When the BIER-TE Forwarding Plane receives a packet, it simply looks up the bit positions that are set in the BitString of the packet in the Bit Index Forwarding Table (BIFT) that was populated by the BIER-TE Controller. For every BP that is set in the BitString, and that has one or more adjacencies in the BIFT, a copy is made according to the type of adjacencies for that BP in the BIFT. Before sending any

copy, the BFR clears all BPs in the BitString of the packet for which the BFR has one or more adjacencies in the BIFT, except when the adjacency indicates "DoNotClear" (DNC, see Section 4.2.1). This is done to inhibit that packets can loop. Because DNC raises the risk of packets looping with inmakes it easier to

3.4. The Routing Underlay

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

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

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

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

3.5. Traffic Engineering Considerations

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

Policy decisions are made within the BIER-TE control plane, i.e., within BIER-TE Controllers. Controllers use policy when composing BitStrings and BFR BIFT state. The mapping of user/IP traffic to

specific BitStrings/BIER-TE flows is made based on policy. The specific details of BIER-TE policies and how a controller uses them are out of scope of this document.

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

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

4. BIER-TE Forwarding

4.1. The Bit Index Forwarding Table (BIFT)

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

Like BIER, BIER-TE can support multiple sub-domains, each with a separate BIFT.

In [RFC8279], Figure 2, indices into the BIFT are both SI:BitString and BFR-id, where BitString is indicating a BP: $BFR-id = SI * 2^{BSL} + BP$. As shown in Figure 4, in BIER-TE, only SI:BP are used as indices into a BIFT because they identify adjacencies and not BFR.

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

Bit Index Forwarding Table

Figure 4: BIFT adjacencies

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

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

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

4.2. Adjacency Types

4.2.1. Forward Connected

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

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

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

4.2.2. Forward Routed

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

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

4.2.3. ECMP

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

A BIER-TE "Equal Cost Multipath" (ECMP) adjacency has a list of two or more non-ECMP adjacencies and a seed parameter. When a BIER-TE packet is copied onto such an ECMP adjacency, an implementation specific so-called hash function will select one out of the list's adjacencies to which the packet is forwarded. This ECMP hash

function MUST select the same adjacency from that list for all packets with the same entropy parameter. The seed parameter allows to design hash functions that are easy to implement at high speed without running into polarization issues across multiple consecutive ECMP hops. See Section 5.1.7 for more explanations.

4.2.4. Local Decap(sulation)

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

4.3. Encapsulation / Co-existence with BIER

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

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

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

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

When a fixed mapping from BSL, SD, SI is used without specifically distinguishing BIER and BIER-TE, such as proposed for non-MPLS forwarding with [RFC8296] in [I-D.ietf-bier-non-mpls-bift-encoding] revision 04, section 5., then it is necessary to allocate disjoint SDs to BIER and BIER-TE BIFT so that both can be addressed by the BIFT-ids. The encoding proposed in section 6. of the same document does not statically encode BSL or SD into the BIFT-id, but allows for a mapping, and hence could provide for the same freedom as when MPLS is being used (same or different SD for BIER/BIER-TE).

"forward_routed" requires an encapsulation that permits to direct unicast encapsulated BIER-TE packets to a specific interface address on a target BFR. With MPLS encapsulation, this can simply be done via a label stack with that addresses label as the top label - followed by the label assigned to the (BSL,SD,SI) BitString. With non-MPLS encapsulation, some form of IP encapsulation would be required (for example IP/GRE).

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

4.4. BIER-TE Forwarding Pseudocode

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

```
void ForwardBitMaskPacket_withTE (Packet)
{
    SI=GetPacketSI(Packet);
    Offset=SI*BitStringLength;
    for (Index = GetFirstBitPosition(Packet->BitString); Index ;
        Index = GetNextBitPosition(Packet->BitString, Index)) {
        F-BM = BIFT[Index+Offset]->F-BM;
        if (!F-BM) continue;                                [3]
        BFR-NBR = BIFT[Index+Offset]->BFR-NBR;
        PacketCopy = Copy(Packet);
        PacketCopy->BitString &= F-BM;                        [2]
        PacketSend(PacketCopy, BFR-NBR);
        // The following must not be done for BIER-TE:
        // Packet->BitString &= ~F-BM;                          [1]
    }
}
```

Figure 5: BIER-TE Forwarding Pseudocode for required functions, based on BIER Pseudocode

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

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

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

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

For BIER-TE, F-BM is handled as follows:

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

This Forwarding Pseudocode can support the REQUIRED BIER-TE forwarding functions (see Section 4.6), forward_connected, forward_routed() and local_decap, but not the RECOMMENDED functions DNC flag and multiple adjacencies per bit nor the OPTIONAL function, ECMP adjacencies. The DNC flag cannot be supported when using only [1] to mask bits.

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

- * This pseudocode eliminates per-bit F-BM, therefore reducing the size of BIFT state by $\text{BitStringLength}^2 * \text{SI}$ and eliminating the need for per-packet-copy masking operation except for adjacencies with the DNC flag set:
 - AdjacentBits[SI] are bits with a non-empty list of adjacencies. This can be computed whenever the BIER-TE Controller updates the adjacencies.
 - Only the AdjacentBits need to be examined in the loop for packet copies.
 - The packet's BitString is masked with those AdjacentBits before the loop to avoid doing this repeatedly for every PacketCopy.
- * The code loops over the adjacencies because there may be more than one adjacency for a bit.
- * When an adjacency has the DNC bit, the bit is set in the packet copy (to save bits in rings for example).
- * The ECMP adjacency is shown. Its parameters are a ListOfAdjacencies from which one is picked.
- * The forward_local, forward_routed, local_decap() adjacencies are shown with their parameters.

```

void ForwardBitMaskPacket_withTE (Packet)
{
    SI=GetPacketSI(Packet);
    Offset=SI*BitStringLength;
    // Set variable for looping across only adjacent bits
    AdjacentBits = Packet->BitString & ~AdjacentBits[SI];
    // Clear adjacent bits in Packet header to avoid loops
    Packet->BitString &= ~AdjacentBits[SI];
    for (Index = GetFirstBitPosition(AdjacentBits); Index ;
        Index = GetNextBitPosition(AdjacentBits, Index)) {
        foreach adjacency BIFT[Index+Offset] {
            if(adjacency == ECMP(ListOfAdjacencies, seed) ) {
                I = ECMP_hash(sizeof(ListOfAdjacencies),
                               Packet->Entropy, seed);
                adjacency = ListOfAdjacencies[I];
            }
            PacketCopy = Copy(Packet);
            switch(adjacency.type) {
                case forward_connected(interface,neighbor,DNC):
                    if(adjacency.DNC)
                        PacketCopy->BitString |= 1<<(Index-1);
                    SendToL2Unicast (PacketCopy,interface,neighbor);

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

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

```

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

4.5. Basic BIER-TE Forwarding Example

[RFC Editor: remove this section.]

THIS SECTION TO BE REMOVED IN RFC BECAUSE IT WAS SUPERCEDED BY SECTION 1.1 EXAMPLE - UNLESS REVIEWERS CHIME IN AND EXPRESS DESIRE TO KEEP THIS ADDITIONAL EXAMPLE SECTION. ALVARO RETANA DID NOT MIND ANOTHER EXAMPLE.

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

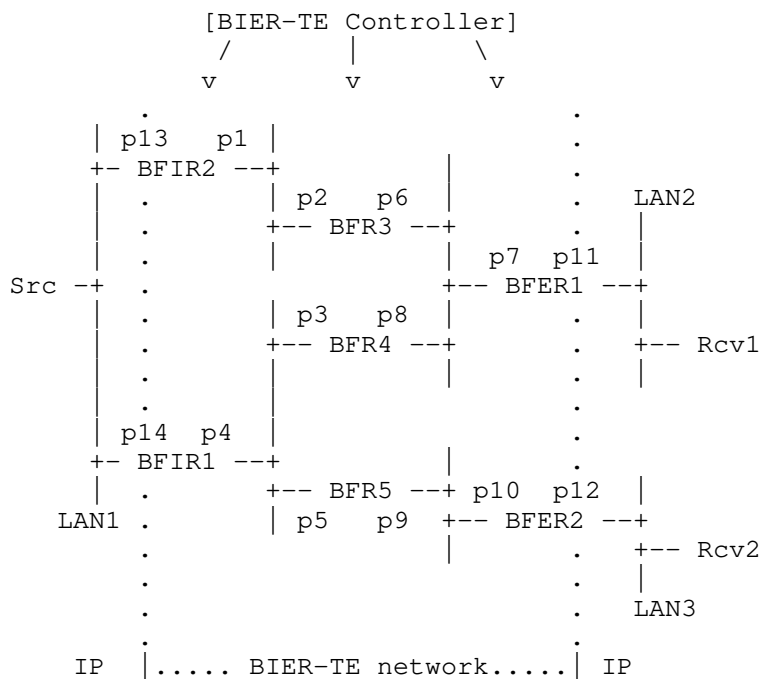


Figure 7: BIER-TE Forwarding Example

pXX indicate the BitPositions number assigned by the BIER-TE Controller to adjacencies in the BIER-TE topology. For example, p9 is the adjacency towards BFR5 on the LAN connecting to BFER2.

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

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

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

Figure 8: BIER-TE Forwarding Example Adjacencies

...and so on.

For example, we assume that some multicast traffic seen on LAN1 needs to be sent via BIER-TE by BFIR2 towards Rcv1 and Rcv2. The BIER-TE Controller determines it wants it to pass this traffic across the following paths:

```

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

```

Figure 9: BIER-TE Forwarding Example Paths

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

This BitString is assigned by BFIR2 to the example multicast traffic received from LAN1.

Then BFIR2 forwards this multicast traffic with BIER-TE based on that BitString. The BIFT of BFIR2 has only p2 and p13 populated. Only p2 is in the BitString and this is an adjacency towards BFR3. BFIR2 therefore clears p2 in the BitString and sends a copy towards BFR2.

BFR3 sees a BitString of p5,p7,p8,p10,p11,p12. For those BPs, it has only adjacencies for p7,p8. It creates a copy of the packet to BFER1 (due to p7) and one to BFR4 (due to p8). It clears both p7 and p8 before sending.

BFER1 sees a BitString of p5,p10,p11,p12. For those BPs, it only has an adjacency for p11. p11 is a "local_decap" adjacency installed by the BIER-TE Controller to receive a copy of the BIER packet - dispose of the BIER header and pass the payload to IP multicast. IP multicast will then forward the packet out to LAN2 because it did receive PIM or IGMP joins on LAN2 for the traffic.

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

4.6. BFR Requirements for BIER-TE forwarding

BFR MUST support to configure the BIFT of sub-domains so that they use BIER-TE forwarding rules instead of (non-TE) BIER forwarding rules. Every BP in the BIFT MUST support to have zero or one adjacency. Forwarding MUST support the adjacency types `forward_connected()` with clear DNC flag, `forward_routed()` and `local_decap`. As explained in Section 4.4, these REQUIRED BIER-TE

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

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

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

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

5. BIER-TE Controller Operational Considerations

5.1. Bit position Assignments

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

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

5.1.1. P2P Links

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

5.1.2. BFER

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

5.1.3. Leaf BFERs

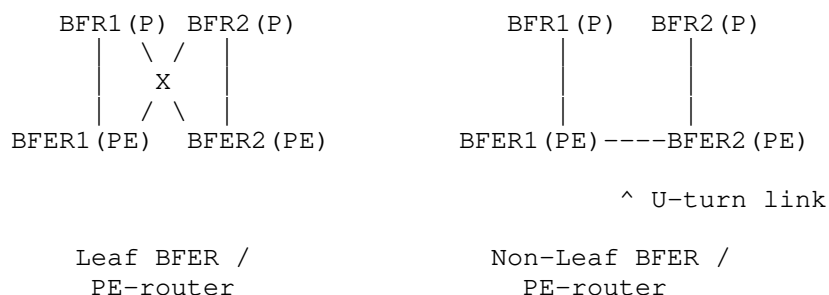


Figure 10: Leaf vs. non-Leaf BFER Example

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

Consider how redundant disjoint traffic can reach BFER1/BFER2 in Figure 10: When BFER1/BFER2 are Non-Leaf BFER as shown on the right hand side, one traffic copy would be forwarded to BFER1 from BFR1, but the other one could only reach BFER1 via BFER2, which makes BFER2 a non-Leaf BFER. Likewise BFER1 is a non-Leaf BFER when forwarding traffic to BFER2. Note that the BFERs in the left hand picture are only guaranteed to be leaf-BFER by fitting routing configuration that prohibits transit traffic to pass through the BFERs, which is commonly applied in these topologies.

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

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

5.1.4. LANs

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

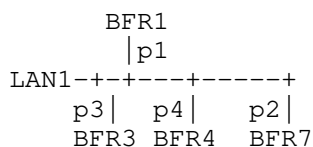


Figure 11: LAN Example

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

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

5.1.5. Hub and Spoke

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

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

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

5.1.6. Rings

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

For the rings shown in Figure 12, a single bit position will suffice to forward traffic entering the ring at BFRa or BFRb all the way up to BFR1:

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

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

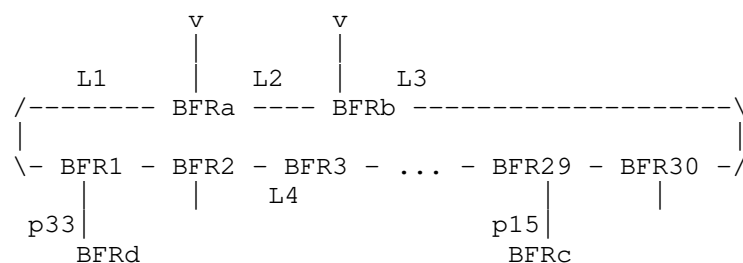


Figure 12: Ring Example

Note that this example only permits for packets intended to make it all the way around the ring to enter it at BFRa and BFRb, and that packets will always travel clockwise. If packets should be allowed to enter the ring at any ring BFR, then one would have to use two ring bit positions. One for each direction: clockwise and counterclockwise.

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

5.1.7. Equal Cost MultiPath (ECMP)

The ECMP adjacency allows to use just one BP per link bundle between two BFRs instead of one BP for each p2p member link of that link bundle. In Figure 13, one BP is used across L1,L2,L3.

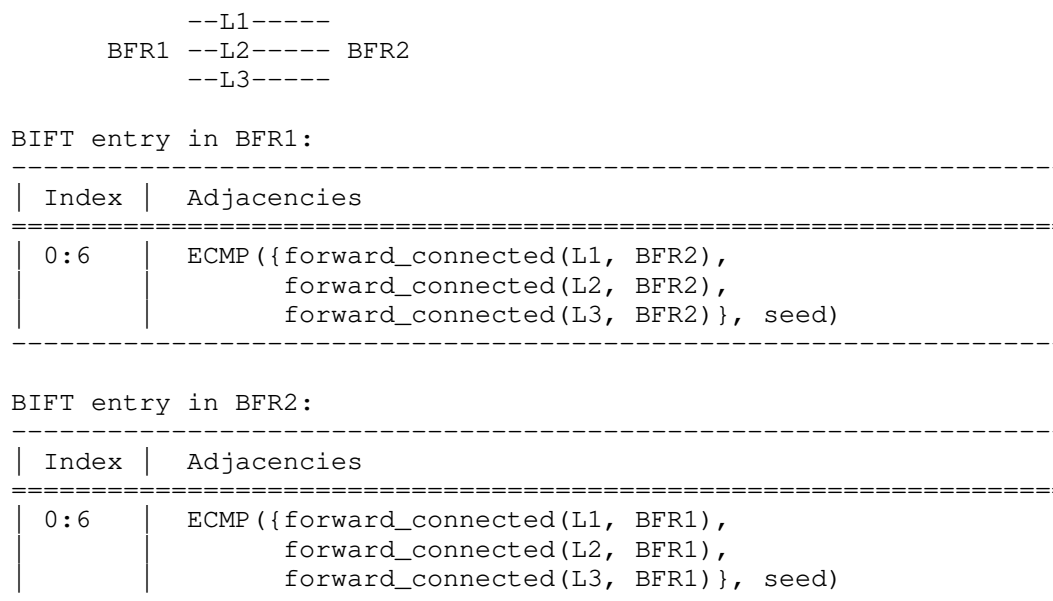


Figure 13: ECMP Example

This document does not standardize any ECMP algorithm because it is sufficient for implementations to document their freely chosen ECMP algorithm. This allows the BIER-TE Controller to calculate ECMP paths and seeds. Figure 14 shows an example ECMP algorithm:

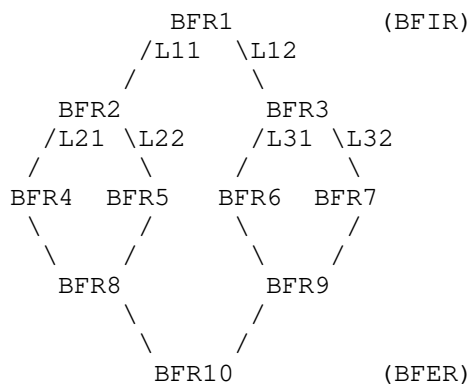
```

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

```

Figure 14: ECMP algorithm Example

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



BIFT entry in BFR1:

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

BIFT entry in BFR2:

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

BIFT entry in BFR3:

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

BIFT entry in BFR4, BFR5:

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

BIFT entry in BFR6, BFR7:

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

BIFT entry in BFR8, BFR9:

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

Figure 15: Polarization Example

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

With the setup of ECMP in the topology above, traffic would not be equally load-split. Instead, links L22 and L31 would see no traffic at all: BFR2 will only see traffic from BFR1 for which the ECMP hash in BFR1 selected the first adjacency in the list of 2 adjacencies given as parameters to the ECMP. It is link L11-to-BFR2. BFR2 performs again ECMP with two adjacencies on that subset of traffic using the same seed1, and will therefore again select the first of its two adjacencies: L21-to-BFR4. And therefore L22 and BFR5 sees no traffic. Likewise for L31 and BFR6.

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

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

5.1.8. Forward Routed adjacencies

5.1.8.1. Reducing bit positions

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

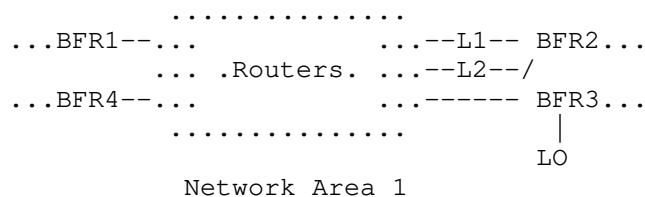


Figure 16: Forward Routed Adjacencies Example

Assume the requirement in Figure 16 is to explicitly steer traffic flows that have arrived at BFR1 or BFR4 via a shortest path in the routing underlay "Network Area 1" to one of the following three next segments: (1) BFR2 via link L1, (2) BFR2 via link L2, or (3) via BFR3.

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

5.1.8.2. Supporting nodes without BIER-TE

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

5.1.9. Reuse of bit positions (without DNC)

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

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

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

An example of (A) was given in Figure 15, where BP 0:7, BP 0:8 and BP 0:9 are each reused across multiple BFRs because a single packet/path would never be able to reach more than one BFR sharing the same BP.

Assume the example was changed: BFR1 has no ECMP adjacency for BP 0:6, but instead BP 0:5 with `forward_connected()` to BFR2 and BP 0:6 with `forward_connected()` to BFR3. Packets with both BP 0:5 and BP 0:6 would now be able to reach both BFR2 and BFR3 and the still existing re-use of BP 0:7 between BFR2 and BFR3 is a case of (B) where reuse of BP is perfect because it does not limit the set of useful path choices:

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

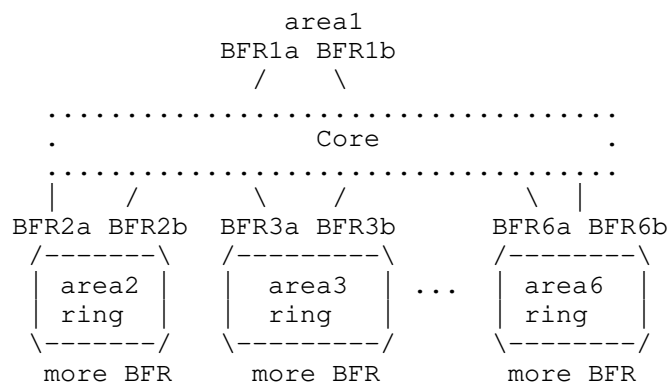


Figure 17: Reuse of BP

Reuse may also save BPs in larger topologies. Consider the topology shown in Figure 17. A BFIR/sender (e.g.: video headend) is attached to area 1, and area 2...6 contain receivers/BFER. Assume each area had a distribution ring, each with two BPs to indicate the direction (as explained before). These two BPs could be reused across the 5 areas. Packets would be replicated through other BPs for the Core to the desired subset of areas, and once a packet copy reaches the ring of the area, the two ring BPs come into play. This reuse is a case of (B), but it limits the topology choices: Packets can only flow around the same direction in the rings of all areas. This may or may not be acceptable based on the desired path steering options: If resilient transmission is the path engineering goal, then it is likely a good optimization, if the bandwidth of each ring was to be optimized separately, it would not be a good limitation.

5.1.10. Summary of BP optimizations

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

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

The optimizations described are then as follows:

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

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

5.2. Avoiding duplicates and loops

5.2.1. Loops

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

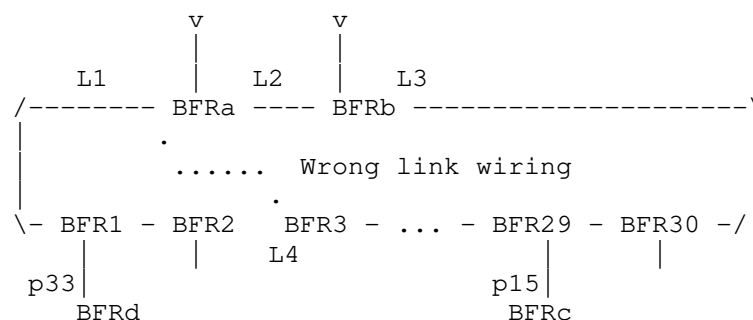


Figure 18: Miswired Ring Example

With DNC set, looping can happen. Consider in Figure 18 that link L4 from BFR3 is (inadvertently) plugged into the L1 interface of BFRa (instead of BFR2). This creates a loop where the rings clockwise bit position is never cleared for copies of the packets traveling clockwise around the ring.

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

5.2.2. Duplicates

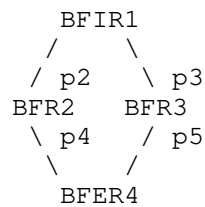


Figure 19: Duplicates Example

Duplicates happen when the graph expressed by a BitString is not a tree but redundantly connecting BFRs with each other. In Figure 19, a BitString of p2,p3,p4,p5 would result in duplicate packets to arrive on BFER4. The BIER-TE Controller must therefore ensure to only create BitStrings that are trees.

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

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

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

When the number of bits required to represent the necessary hops in the topology and BFER exceeds the supported BitStringLength (BSL), multiple SIs and/or sub-domains must be used. This section discusses how.

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

5.3.1. Why SI and sub-domains

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

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

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

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

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

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

To be able to easily reuse (and modify as little as possible) existing BIER procedures including flow-overlay and routing underlay, when BIER-TE forwarding is added, we therefore reuse SI and sub-domain logically in the same way as they are used in BIER: All necessary BFIR/BFER for a service use a single BIER-TE BIFT and are split across as many SIs as necessary (see Section 5.3.2). Different services may use different sub-domains that primarily exist to provide more efficient replication (and for BIER-TE desirable path steering) for different subsets of BFIR/BFER.

5.3.2. Assigning bits for the BIER-TE topology

In BIER, BitStrings only need to carry bits for BFERs, which leads to the model that BFR-ids map 1:1 to each bit in a BitString.

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

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

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

5.3.3. Assigning BFR-id with BIER-TE

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

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

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

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

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

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

5.3.4. Mapping from BFR to BitStrings with BIER-TE

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

In BIER-TE this mapping needs to be equally supported for flow overlays. This section outlines two core options, based on how "complex" the Tree Engineering is that the BIER-TE controller performs for a particular application.

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

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

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

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

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

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

5.3.5. Assigning BFR-ids for BIER-TE

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

- A. BIER and BIER-TE have different BFR-id in the same sub-domain. This allows higher replication efficiency for BIER because their BFR-id can be assigned sequentially, while the BitStrings for BIER-TE will have also the additional bits for the topology. There is no relationship between a BFR BIER BFR-id and BIER-TE BFR-id.
- B. BIER and BIER-TE share the same BFR-id. The BFR-ids are assigned as explained above for BIER-TE and simply reused for BIER. The replication efficiency for BIER will be as low as that for BIER-TE in this approach.

5.3.6. Example bit allocations

5.3.6.1. With BIER

Consider a network setup with a BSL of 256 for a network topology as shown in Figure 20. The network has 6 areas, each with 170 BFERs, connecting via a core with 4 (core) BFRs. To address all BFERs with BIER, 4 SIs are required. To send a BIER packet to all BFER in the network, 4 copies need to be sent by the BFIR. On the BFIR it does not make a difference how the BFR-ids are allocated to BFER in the network, but for efficiency further down in the network it does make a difference.

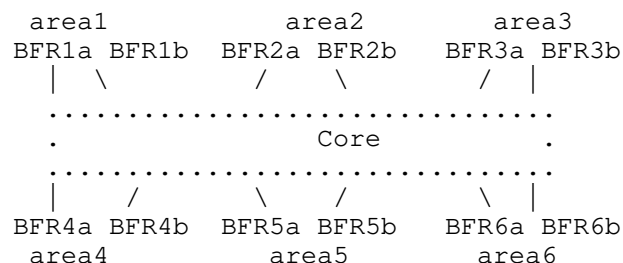


Figure 20: Scaling BIER-TE bits by reuse

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

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

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

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

5.3.6.2. With BIER-TE

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

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

Every time that different sub-topologies have overlap, bits need to be repeated across the BitStrings, increasing the overall amount of bits required across all BitString/SIs. In the worst case, random

subsets of BFERs are assigned to different SIs. This is much worse than in (non-TE) BIER because it not only reduces replication efficiency with the same number of overall bits, but even further - because more bits are required due to duplication of bits for topology across multiple SIs. Intelligent BFER to SI assignment and selecting specific "desired" subtopologies can minimize this problem.

To set up BIER-TE efficiently for the topology of Figure 20, the following bit allocation method can be used. This method can easily be expanded to other, similarly structured larger topologies.

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

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

On all BFIRs in an area $j|j=2\dots6$, bia in each BIFT:SI is populated with the same forward_routed(BFRja), and bib with forward_routed(BFRjb). On all area edge BFR, bea in BIFT:SI= $k|k=2\dots6$ is populated with forward_routed(BFRka) and beb in BIFT:SI= k with forward_routed(BFRkb). For this setup we do not consider area 1 because we assume the BIER-TE setup is just for sending traffic from area 1 into area 2...6, for example because the broadcast headends are in area 1 for an IPTV BIER-TE setup.

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

5.3.7. Summary

BIER-TE can, like BIER, support multiple SIs within a sub-domain to allow re-using the concept of BFR-id and therefore minimize BIER-TE specific functions in any possible BIER layer control plane used in conjunction with BIER-TE, flow overlay methods and BIER headers.

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

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

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

6. BIER-TE and Segment Routing

SR aims to enable lightweight path steering via loose source routing. Compared to its more heavy-weight predecessor RSVP-TE, SR does for example not require per-path signaling to each of these hops.

BIER-TE supports the same design philosophy for multicast. Like in SR, it relies on source-routing - via the definition of a BitString. Like SR, it only requires to consider the "hops" on which either replication has to happen, or across which the traffic should be steered (even without replication). Any other hops can be skipped via the use of routed adjacencies.

BIER-TE bit position (BP) can be understood as the BIER-TE equivalent of "forwarding segments" in SR, but they have a different scope than SR forwarding segments. Whereas forwarding segments in SR are global or local, BPs in BIER-TE have a scope that is the group of BFR(s) that have adjacencies for this BP in their BIFT. This can be called "adjacency" scoped forwarding segments.

Adjacency scope could be global, but then every BFR would need an adjacency for this BP, for example a `forward_routed()` adjacency with encapsulation to the global SR SID of the destination. Such a BP would always result in ingress replication though (as in [RFC7988]). The first BFR encountering this BP would directly replicate to it. Only by using non-global adjacency scope for BPs can traffic be steered and replicated on non-ingress BFR.

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

Note that (non-TE) BIER itself can also be seen to be similar to SR. BIER BPs act as global destination Node-SIDs and the BIER BitString is simply a highly optimized mechanism to indicate multiple such SIDs and let the network take care of effectively replicating the packet

hop-by-hop to each destination Node-SID. What BIER does not allow is to indicate intermediate hops, or in terms of SR the ability to indicate a sequence of SID to reach the destination. This is what BIER-TE and its adjacency scoped BP enables.

7. Security Considerations

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

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

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

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

Whereas IGP routing protocols are most often not well secured through cryptographic authentication and confidentiality, communications between controllers and routers such as those to be considered for the BIER-TE controller/control-plane can be and are much more commonly secured with those security properties, for example by using Secure SHell (SSH), [RFC4253] for NetConf ([RFC6241]), or via Transport Layer Security (TLS), such as [RFC8253] for PCEP, [RFC5440], or [RFC7589] for NetConf. BIER-TE controllers SHOULD use security equal to or better than these mechanisms.

For additional, BIER-TE independent security considerations for the use of a central BIER-TE controller, the security section of the protocols and security options in the previous paragraph apply. In addition, the security considerations of [RFC4655] apply.

The most important attack vector in BIER-TE is misconfiguration, either on the BFR themselves or via the BIER-TE controller. Forwarding entries with DNC could be set up to create persistent loops, in which packets only expire because of TTL. To minimize the impact of such attacks (or more likely unintentional misconfiguration

by operators and/or bad BIER-TE controller software), the BIER-TE forwarding rules are defined to be as strict in clearing bits as possible. The clearing of all bits with an adjacency on a BFR prohibits that a looping packet creates additional packet amplification through the misconfigured loop on the packet's second or further times around the loop, because all relevant adjacency bits would have been cleared on the first round through the loop. In result, BIER-TE has the same degree of looping packets as possible with unintentional or malicious loops in the routing underlay with BIER or even with unicast traffic.

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

8. IANA Considerations

This document requests no action by IANA.

9. Acknowledgements

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

10. Change log [RFC Editor: Please remove]

draft-ietf-bier-te-arch:

11: IESG review Ben Kaduk, summary:

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

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

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

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

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

10: AD review Alvaro Retana, summary:

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

Summary:

1. restructuring: merged all controller sections under common controller ops main section, moved unfitting stuff out to other parts of doc. Split Intro section into Overview and Intro. Shortened Abstract, moved text into Overview, added sections overview.
2. enhanced/rewrote: 2.3 Comparison with -> Relationship to BIER-TE
3. enhanced/rewrote: 3.2 BIER-TE controller -> BIER-TE control plane, 3.2.1 BIER-TE controller, for consistency with rfc8279
4. additional subsections for Alvaros asks
5. added to: 3.3 BIER-TE forwarding plane (consistency with rfc8279)
6. Enhanced description of 4.3/encap considerations to better explain how BIER/BIER-TE can run together.

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

Details:.

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

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

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

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

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

Renamed to Overview because that is now more factually correct.

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

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

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

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

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

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

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

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

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

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

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

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

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

3.4 Alvaro does not mind additional example, fixed bugs.

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

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

4.2.3 enhanced ECMP text (p).

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

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

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

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

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

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

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

co-authors internal review:

4.4 Added xref to Figure 5.

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

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

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

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

7. Added new security considerations (u).

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

Added references for Bloom Filters and Rate Controlled Service Disciplines.

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

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

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

3.1 Added explanation for VRF and its use in adjacencies.

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

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

07: Further reworking text for Lou.

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

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

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

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

Changed title "Traffic Engineering" to "Path Engineering"

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

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

Other:

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

05: Review Jeffrey Zhang.

Part 2:

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

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

4.9 added section discussing generic reuse of BP methods.

4.10 added section summarizing BP optimizations of section 4.

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

Part 1:

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

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

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

4.8.1 Simplified example picture for routed adjacency, explanatory text.

Review from Dirk Trossen:

Fixed up explanation of ICC paper vs. bloom filter.

04: spell check run.

Added remaining fixes for Sandys (Zhang Zheng) review:

4.7 Enhance ECMP explanations:

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

Review from Dirk Trossen:

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

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

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

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

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

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

many typos fixed. Thanks a lot.

03: Last call textual changes by authors to improve readability:

removed Wolfgang Braun as co-authors (as requested).

Improved abstract to be more explanatory. Removed mentioning of FRR (not concluded on so far).

Added new text into Introduction section because the text was too difficult to jump into (too many forward pointers). This primarily consists of examples and the early introduction of the BIER-TE Topology concept enabled by these examples.

Amended comparison to SR.

Changed syntax from [VRF] to {VRF} to indicate its optional and to make idnits happy.

Split references into normative / informative, added references.

02: Refresh after IETF104 discussion: changed intended status back to standard. Reasoning:

Tighter review of standards document == ensures arch will be better prepared for possible adoption by other WGs (e.g. DetNet) or std. bodies.

Requirement against the degree of existing implementations is self defined by the WG. BIER WG seems to think it is not necessary to apply multiple interoperating implementations against an architecture level document at this time to make it qualify to go to standards track. Also, the levels of support introduced in -01 rev. should allow all BIER forwarding engines to also be able to support the base level BIER-TE forwarding.

01: Added note comparing BIER and SR to also hopefully clarify BIER-TE vs. BIER comparison re. SR.

- added requirements section mandating only most basic BIER-TE forwarding features as MUST.

- reworked comparison with BIER forwarding section to only summarize and point to pseudocode section.

- reworked pseudocode section to have one pseudocode that mirrors the BIER forwarding pseudocode to make comparison easier and a second pseudocode that shows the complete set of BIER-TE forwarding options and simplification/optimization possible vs. BIER forwarding. Removed MyBitsOfInterest (was pure optimization).

- Added captions to pictures.

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

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

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

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

draft-eckert-bier-te-arch:

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

05: Author affiliation change only.

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

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

- Linked FRR information to new draft in Overview/Introduction
- Removed BTAFT/FRR from "Changes in the network topology"
- Linked new draft in "Link/Node Failures and Recovery"
- Removed FRR from "The BIER-TE Forwarding Layer"
- Moved FRR section to new draft
- Moved FRR parts of Pseudocode into new draft
- Left only non FRR parts
- removed FrrUpDown(..) and //FRR operations in ForwardBierTePacket(..)
- New draft contains FrrUpDown(..) and ForwardBierTePacket(Packet) from bier-arch-03
- Moved "BIER-TE and existing FRR to new draft
- Moved "BIER-TE and Segment Routing" section one level up
- Thus, removed "Further considerations" that only contained this section
- Added Changes for version 04

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

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

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

02: Added Section 5.3 to explain the use of SI, sub-domains and BFR-id in BIER-TE and to give an example how to efficiently assign bits for a large topology requiring multiple SI.

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

01: Fixed BFIR -> BFER for section 4.3.

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

00: Initial version.

11. References

11.1. Normative References

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

11.2. Informative References

- [Bloom70] Bloom, B. H., "Space/time trade-offs in hash coding with allowable errors", Comm. ACM 13(7):422-6, July 1970, <<http://gnunet.org/papers/p422-bloom.pdf>>.

[I-D.eckert-bier-te-frr]

Eckert, T., Cauchie, G., Braun, W., and M. Menth,
"Protection Methods for BIER-TE", Work in Progress,
Internet-Draft, draft-eckert-bier-te-frr-03, 5 March 2018,
<<https://www.ietf.org/archive/id/draft-eckert-bier-te-frr-03.txt>>.

[I-D.ietf-bier-multicast-http-response]

Trossen, D., Rahman, A., Wang, C., and T. Eckert,
"Applicability of BIER Multicast Overlay for Adaptive
Streaming Services", Work in Progress, Internet-Draft,
draft-ietf-bier-multicast-http-response-06, 10 July 2021,
<<https://www.ietf.org/archive/id/draft-ietf-bier-multicast-http-response-06.txt>>.

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

Wijnands, I., Mishra, M., Xu, X., and H. Bidgoli, "An
Optional Encoding of the BIFT-id Field in the non-MPLS
BIER Encapsulation", Work in Progress, Internet-Draft,
draft-ietf-bier-non-mpls-bift-encoding-04, 30 May 2021,
<<https://www.ietf.org/archive/id/draft-ietf-bier-non-mpls-bift-encoding-04.txt>>.

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

Zhang, Z., Wang, C., Chen, R., Hu, F., Sivakumar, M., and
H. Chen, "A YANG data model for Tree Engineering for Bit
Index Explicit Replication (BIER-TE)", Work in Progress,
Internet-Draft, draft-ietf-bier-te-yang-04, 7 November
2021, <<https://www.ietf.org/archive/id/draft-ietf-bier-te-yang-04.txt>>.

[I-D.ietf-roll-ccast]

Bergmann, O., Bormann, C., Gerdes, S., and H. Chen,
"Constrained-Cast: Source-Routed Multicast for RPL", Work
in Progress, Internet-Draft, draft-ietf-roll-ccast-01, 30
October 2017, <<https://www.ietf.org/archive/id/draft-ietf-roll-ccast-01.txt>>.

[I-D.ietf-teas-rfc3272bis]

Farrel, A., "Overview and Principles of Internet Traffic
Engineering", Work in Progress, Internet-Draft, draft-
ietf-teas-rfc3272bis-13, 8 November 2021,
<<https://www.ietf.org/archive/id/draft-ietf-teas-rfc3272bis-13.txt>>.

[ICC]

Reed, M. J., Al-Naday, M., Thomos, N., Trossen, D.,
Petropoulos, G., and S. Spirou, "Stateless multicast
switching in software defined networks", IEEE

- International Conference on Communications (ICC), Kuala Lumpur, Malaysia, 2016, May 2016,
<<https://ieeexplore.ieee.org/document/7511036>>.
- [RCSD94] Zhang, H. and D. Domenico, "Rate-Controlled Service Disciplines", Journal of High-Speed Networks, 1994, May 1994, <<https://dl.acm.org/doi/10.5555/2692227.2692232>>.
- [RFC4253] Ylonen, T. and C. Lonvick, Ed., "The Secure Shell (SSH) Transport Layer Protocol", RFC 4253, DOI 10.17487/RFC4253, January 2006, <<https://www.rfc-editor.org/info/rfc4253>>.
- [RFC4655] Farrel, A., Vasseur, J.-P., and J. Ash, "A Path Computation Element (PCE)-Based Architecture", RFC 4655, DOI 10.17487/RFC4655, August 2006, <<https://www.rfc-editor.org/info/rfc4655>>.
- [RFC5440] Vasseur, JP., Ed. and JL. Le Roux, Ed., "Path Computation Element (PCE) Communication Protocol (PCEP)", RFC 5440, DOI 10.17487/RFC5440, March 2009, <<https://www.rfc-editor.org/info/rfc5440>>.
- [RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011, <<https://www.rfc-editor.org/info/rfc6241>>.
- [RFC7589] Badra, M., Luchuk, A., and J. Schoenwaelder, "Using the NETCONF Protocol over Transport Layer Security (TLS) with Mutual X.509 Authentication", RFC 7589, DOI 10.17487/RFC7589, June 2015, <<https://www.rfc-editor.org/info/rfc7589>>.
- [RFC7752] Gredler, H., Ed., Medved, J., Previdi, S., Farrel, A., and S. Ray, "North-Bound Distribution of Link-State and Traffic Engineering (TE) Information Using BGP", RFC 7752, DOI 10.17487/RFC7752, March 2016, <<https://www.rfc-editor.org/info/rfc7752>>.
- [RFC7988] Rosen, E., Ed., Subramanian, K., and Z. Zhang, "Ingress Replication Tunnels in Multicast VPN", RFC 7988, DOI 10.17487/RFC7988, October 2016, <<https://www.rfc-editor.org/info/rfc7988>>.

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

Authors' Addresses

Toerless Eckert (editor)
Futurewei Technologies Inc.
2330 Central Expy
Santa Clara, 95050
United States of America

Email: tte+ietf@cs.fau.de

Gregory Cauchie
Bouygues Telecom

Email: GCAUCHIE@bouyguestelecom.fr

Michael Menth
University of Tuebingen

Email: menth@uni-tuebingen.de

PCE Working Group
Internet-Draft
Intended status: Standards Track
Expires: April 22, 2022

H. Li
A. Wang
China Telecom
H. Chen
Futurewei
R. Chen
ZTE Corporation
October 19, 2021

PCE based BIER Procedures and Protocol Extensions
draft-li-pce-based-bier-02

Abstract

This document describes extensions to Path Computation Element (PCE) communication Protocol (PCEP) for supporting the PCE based Bit Index Explicit Replication (BIER) deployment.

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 22, 2022.

Copyright Notice

Copyright (c) 2021 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. Terminology	3
4. Overview of PCE based BIER solution	4
4.1. Example of PCE based BIER Topology	4
4.2. Basic Procedures	5
5. Capability Advertisement	5
6. PCEP message	6
6.1. PCRpt message	6
6.2. PCUpd message	7
7. Object formats	8
7.1. Multicast Source Registration Object	8
7.1.1. Multicast Source Address TLV	9
7.1.2. BIER Information TLV	10
7.1.3. VPN Information TLV	10
7.2. Multicast Receiver Information Object	11
7.2.1. Multicast Group Address TLV	12
7.3. Forwarding Indication Object	12
7.4. Multicast Receiver Status Object	13
8. Procedures	14
8.1. Multicast source registration and revocation	14
8.2. Joining and leaving of multicast receivers	15
8.3. BitString management	15
8.4. Receiver information synchronization	15
9. Deployment Considerations	16
10. Security Considerations	16
11. IANA Considerations	16
11.1. BIER-MULTICAST-CAPABILITY	16
11.2. PCEP-ERROR Object	16
11.3. New Objects	16
11.4. New TLVs	16
12. Contributor	17
13. Acknowledgement	17
14. Normative References	17
Authors' Addresses	18

1. Introduction

[RFC8279] defines a Bit Index Explicit Replication (BIER) architecture where all intended multicast receivers are encoded as a bitmask in the multicast packet header within different encapsulations such as described in [RFC8296]. A router that receives such a packet will forward the packet based on the bit

position in the packet header towards the receiver(s) following a precomputed tree for each of the bits in the packet. Each receiver is represented by a unique bit in the bitmask.

Currently, multicast management information is mainly signaled by PIM [RFC2362] or BGP [RFC6514], which have some limitations in the deployment and process.

[RFC4655] defines a stateful PCE to be one in which the PCE maintains "strict synchronization between the PCE and not only the network states (in term of topology and resource information), but also the set of computed paths and reserved resources in use in the network." [RFC8231] specifies a set of extensions to PCEP to support state synchronization between PCCs and PCEs.

This document specifies PCEP protocol extensions to optimize the implementation of multicast source registration or revocation, receiver automatic discovery, and forwarding control of multicast data by using PCEP messages to transmit multicast management signaling, combining with the forwarding characteristics of BIER.

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. Terminology

The following terms are used in this document:

- o BFR-id: BFR Identifier. It is a number in the range [1,65535]
- o BGP: Border Gateway Protocol
- o BIER: Bit Index Explicit Replication
- o BIFT: Bit Index Forwarding Table
- o FI: Forwarding indication
- o IGMP: Internet Group Management Protocol
- o IGP: Interior Gateway Protocols
- o MLD: Multicast Listener Discover

- o MRI: Multicast Receiver Information
- o MSR: Multicast Source Registration
- o PCC: Path Computation Client
- o PCE: Path Computation Element
- o PCEP: PCE communication Protocol
- o PIM: Protocol Independent Multicast

4. Overview of PCE based BIER solution

PCE based BIER includes multicast source registration information management, multicast receiver information management and multicast data forwarding control.

Multicast source registration information includes registration and processing of multicast source information.

Multicast receiver information includes requesting multicast group, multicast source and BitPosition information of receiver-side PCC.

Multicast data forwarding control includes BitString processing and data forwarding.

PCRpt message and PCUpd message, described in [RFC8231], are used in the PCE based BIER processing.

This document specifies PCEP protocol extensions for multicast group management, including Multicast Source Registration (MSR) object, Multicast Receiver Information (MRI) object, Forwarding Indication (FI) object and Multicast Receiver Status (MRS) object.

4.1. Example of PCE based BIER Topology

An example of PCE based BIER topology for a BIER domain with a controller as PCE is shown in Figure 1. In this domain, node R1 and R7 are Bit-Forwarding Ingress Router (BFIR) and Bit-Forwarding Egress Router (BFER), respectively.

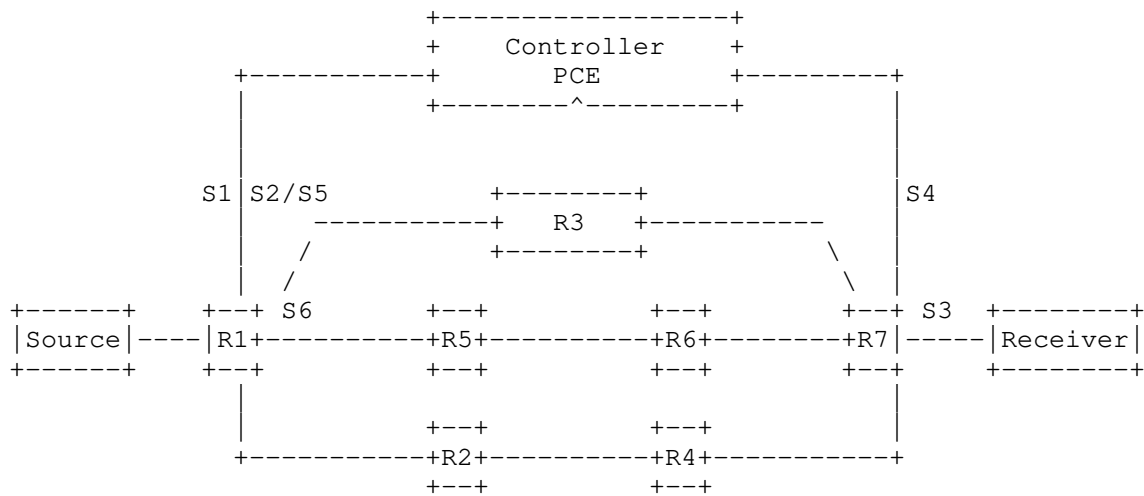


Figure 1: Example of PCE based BIER Topology(controller as PCE)

4.2. Basic Procedures

Step 1(S1): R1 sends multicast source information and authentication information to the controller about multicast information registration via PCRpt message.

Step 2(S2): The controller sends PCUpd message to R1, carrying authentication result.

Step 3(S3): Receivers send IGMP or MLD messages to R7 requesting to join or leave a multicast group.

Step 4(S4): R7 converts the IGMP or MLD messages into PCRpt message and sends it to the controller.

Step 5(S5): If the multicast group and multicast source information requested by the receiver has registered, the controller will send PCUpd message to R1 to start or stop forwarding, carrying BitString.

Step 6(S6): If R1 is ready to start forwarding, it will encapsulate BIER header and forward them based on BIFT and BitString when receiving multicast packets.

5. Capability Advertisement

During the PCEP initialization phase, PCEP speakers advertise stateful capability via the STATEFUL-PCE-CAPABILITY TLV in the OPEN

object. Various flags are defined for the STATEFUL-PCE-CAPABILITY TLV defined in [RFC8231] and updated in [RFC8232] and [RFC8281].

A new flag is added in this document, whose code point is TBD1:

B (BIER-MULTICAST-CAPABILITY, 1 bit): If set to 1 by a PCEP speaker, it indicates that the PCEP speaker supports the capability of these new flag as specified in this document.

If a PCEP speaker receives PCEP message with the newly defined object, but without the B bit set in STATEFUL-PCE-CAPABILITY TLV in the OPEN object, it MUST:

- o Send a PCErr message with Error-Type=10 (Reception of an invalid object) and Error-Value TBD2 (BIER-MULTICAST-CAPABILITY bit is not set).
- o Terminate the PCEP session.

6. PCEP message

6.1. PCRpt message

MSR objectSection 7.1 should be included in the PCRpt message when PCC registers multicast source information with PCE.

MRI objectSection 7.2 should be included in the PCRpt message when PCC sends multicast join messages to PCE.

MRS objectSection 7.4 should be included in the PCRpt message when PCC inform PCE of the number of receivers.

The definition of the PCRpt message from [RFC8231] is extended to optionally include MSR object, MRI object and MRS object after the path object. The encoding from [RFC8231] will become:

```
<PCRpT Message> ::= <Common Header>  
                        <state-report-list>
```

Where:

```
<state-report-list> ::= <state-report> [<state-report-list>]
```

```
<state-report> ::= [<SRP>]  
                  <LSP>  
                  <path>  
                  [<MSR>]  
                  [<MRI>]  
                  [<MRS>]
```

Where:

<path> is as per [RFC8231] and the LSP and SRP object are also defined in [RFC8231].

6.2. PCUpd message

MSR objectSection 7.1 should be included in the PCUpd message when PCE responds to the registration request.

FI objectSection 7.3 should be included in the PCUpd message when PCE sends the BitString to PCC to indicate the path of multicast data packets forwarding for PCC.

MRS objectSection 7.4 should be included in the PCUpd message when PCE inform PCC of the number of receivers.

The definition of the PCUpd message from [RFC8231] is extended to optionally include MSR object, FI object and MRS object after the path object. The encoding from [RFC8231] will become:


```
<PCUpd Message> ::= <Common Header>
                        <update-request-list>
```

Where:

```
<update-request-list> ::= <update-request> [<update-request-list>]
```

```
<update-request> ::= <SRP>
                        <LSP>
                        <path>
                        [<MSR>]
                        [<FI>]
                        [<MRS>]
```

Where:

<path> is as per [RFC8231] and the LSP and SRP object are also defined in [RFC8231].

7. Object formats

7.1. Multicast Source Registration Object

The MSR object is optional and specifies multicast source information in multicast registration information management. The MSR object should be carried within a PCRpt message sent by PCC to PCE for registration. The MSR object should be carried within a PCUpd message sent by PCE to PCC in response to registration.

MSR Object-Class is TBD3. MSR Object-Type is 1.

The format of the MSR object body is:

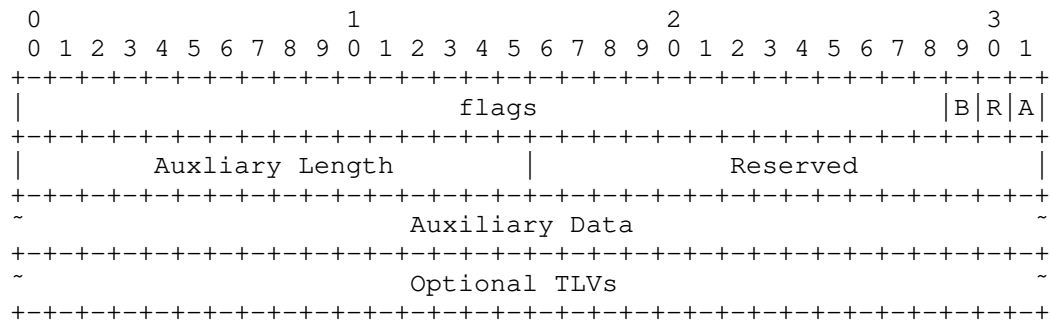


Figure 2: MSR Object Body Format

B(BIER multicast flag, 1 bit): The R flag set to 1 indicates that multicast protocol is BIER. The R flag set to 0 indicates that multicast protocol is not BIER.

R (Register flag, 1 bit): The R flag set to 1 indicates that the PCC is registering multicast information to the PCE. The R flag set to 0 indicates that the PCC revokes the register.

A (Authentication flag, 1 bit): The A flag set to 1 indicates success of registration. The A flag set to 0 indicates failure of registration or cancellation of registration. R and A cannot both be set to 0 or 1 in PCRpt message.

Auxiliary Length(8 bits): indicates the length of Auxiliary Data.

Auxiliary Data(Variable length): contains functional data such as authentication information.

MSR object could include three types of TLVs, namely Multicast Source Address TLV, BIER Information TLV, VPN Information TLV, as defined follows:

7.1.1. Multicast Source Address TLV

The format of the Multicast Source Address TLV is:

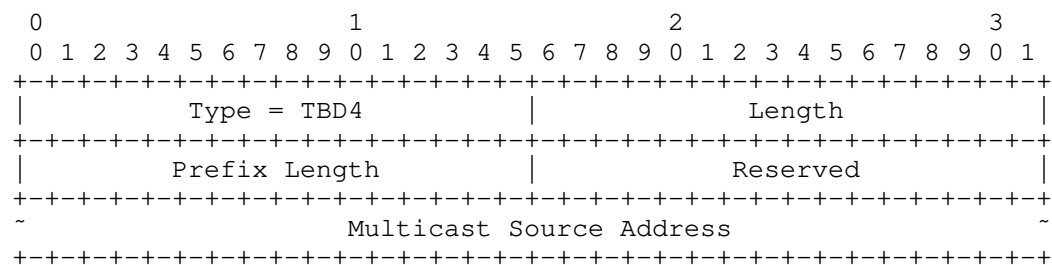


Figure 3: Multicast Source Address TLV Format

Type(16 bits): TBD4 is to be assigned by IANA.

Length: Variable.

Prefix Length(16 bits): indicates the length of multicast source address.

Multicast Source Address(Variable length): contains IPv4 or IPv6 address of the multicast source.

7.1.2. BIER Information TLV

BIER Information TLV is used to report router location information in the BIER domain. When the multicast flag in MSR, MRI, FI objects is set, BIER Information TLV should be included. The format of the BIER Information TLV is:

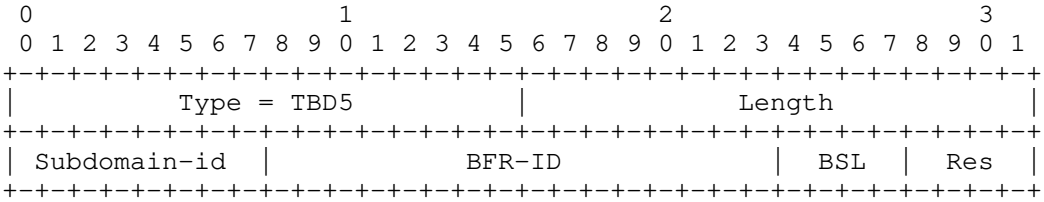


Figure 4: BIER Information TLV Format

Type(16 bits): TBD5 is to be assigned by IANA.

Length: Variable.

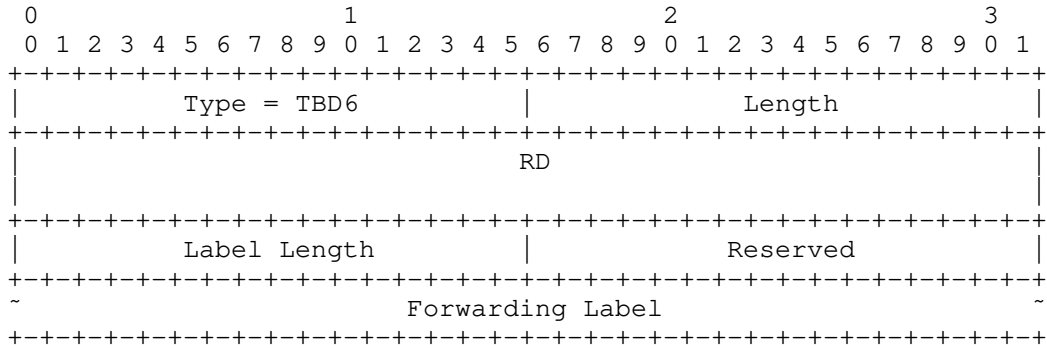
Subdomain-id(8 bits): Unique value identifying the BIER subdomain.

BFR-ID (16 bits): Identification of BFR in a subdomain.

BSL(BitString Length, 4 bits): encodes the length in bits of the BitString as per[RFC8296] , the maximum length of the BitString is 7, it indicates the length of BitString is 4096. It is used to refer to the number of bits in the BitString.

7.1.3. VPN Information TLV

VPN Information TLV is used to report VPN information about multicast sources and receivers. When the multicast flag in MSR, MRI, FI objects is set, VPN Information TLV should be included. The format of the VPN Information TLV is:



Type(16 bits): TBD6 is to be assigned by IANA.

Length: Variable.

RD(Route Distinguisher, 8 bytes): indicates the VPN which the receiver used.

Label Length(16 bits): indicates the length of forwarding label Data, the length should be 0 ,32 bits or 128 bits.

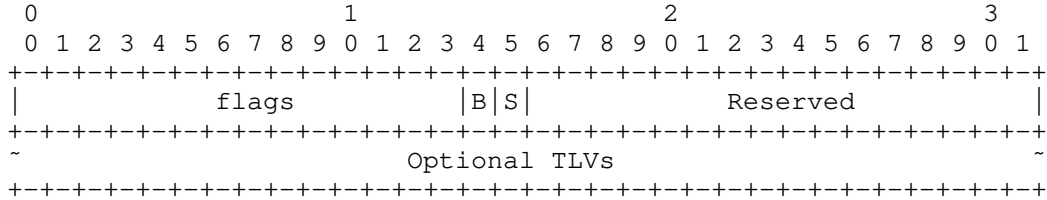
Forwarding Label(Variable Length): contains MPLS label with 32 bit or IPv6 Segment Identifier with 128 bits.

7.2. Multicast Receiver Information Object

The MRI object is optional and specifies receivers' information for matching the multicast registration information. The MRI object should be carried within a PCRpt message sent by PCC to PCE in muticast joining or leaving.

MRI Object-Class is TBD7. MRI Object-Type is 1.

The format of the MRI object body is:



B(BIER multicast flag, 1 bit): The R flag set to 1 indicates that multicast protocol is BIER. The R flag set to 0 indicates that multicast protocol is not BIER.

S(Subscribe flag, 1 bit): The S flag set to 1 indicates that PCC delivers the message requesting to join PCE. The S flag set to 0 indicates that PCC delivers the message requesting to leave to PCE.

MRI object could include four types of TLVs, namely Multicast Source Address TLV Section 7.1.1, BIER INFO TLV Section 7.1.2, VPN Information TLV Section 7.1.3 and Multicast Group Address TLV. Multicast Group Address TLV is defined as follows:

7.2.1. Multicast Group Address TLV

The format of the Multicast Group Address TLV is:

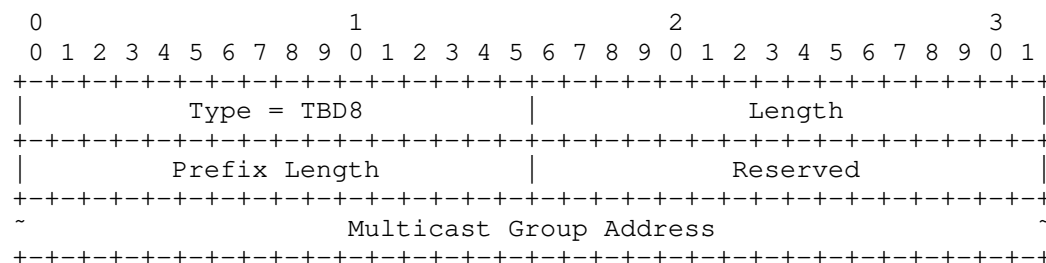


Figure 7: Multicast Group Address TLV Format

Type(16 bits): TBD8 is to be assigned by IANA.

Length: Variable.

Prefix Length(16 bits): indicates the length of multicast group address.

Multicast Group Address(Variable length): contains IPv4 or IPv6 address of the multicast group.

7.3. Forwarding Indication Object

The FI object is optional and used to indicate to the headend how to forward multicast data packets in the form of BitString. The FI object should be carried within a PCUpd message sent by PCE to PCC in multicast scenarios.

FI Object-Class is TBD9. FI Object-Type is 1.

The format of the FI object body is:

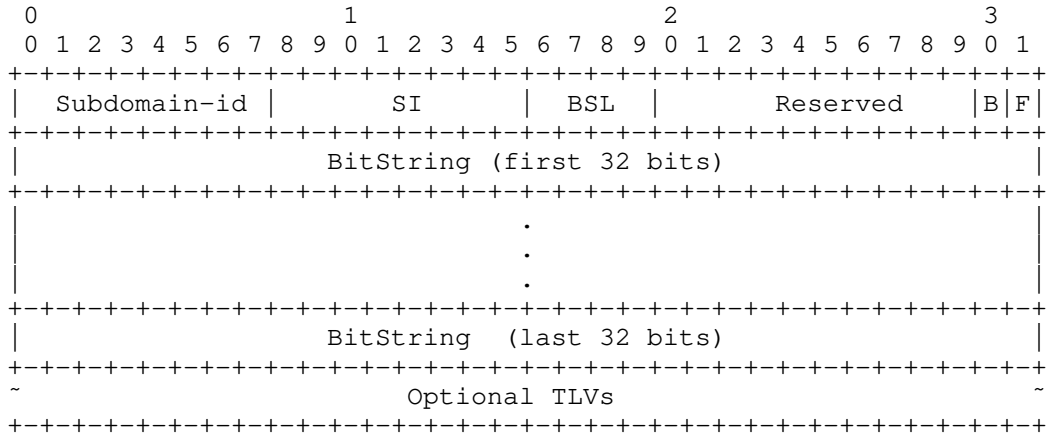


Figure 8: FI Object Body Format

Subdomain-id(8 bits): Unique value identifying the BIER subdomain.

SI (Set Identifier, 8 bits): encoding the Set Identifier used in the encapsulation for this BIER subdomain for this BitString length..

BSL(BitString Length, 4 bits): encodes the length in bits of the BitString as per[RFC8296] , the maximum length of the BitString is 7, it indicates the length of BitString is 4096. It is used to refer to the number of bits in the BitString.

B(BIER multicast flag, 1 bit): The R flag set to 1 indicates that multicast protocol is BIER. The R flag set to 0 indicates that multicast protocol is not BIER.

F(Forwarding flag, 1 bit): The F flag set to 1 indicates that the router may start forwarding multicast packets. The F flag set to 0 indicates that the router should stop forwarding multicast packets.

BitString(Variable length): indicates the path of multicast data packets forwarding for headend.

FI object should include three types of TLVs, namely Multicast Source Address TLVSection 7.1.1, VPN Information TLVSection 7.1.3 and Multicast Group Address TLVSection 7.2.1.

7.4. Multicast Receiver Status Object

The MRS object is optional and used to inform PCE of the number of receivers. The MRS object should be carried within a PCRpt or a PCUpd message for synchronize receiver information periodically, or PCRpt message for the leaving of receivers.

MRS Object-Class is TBD10. MRS Object-Type is 1.

The format of the MRS object body is:

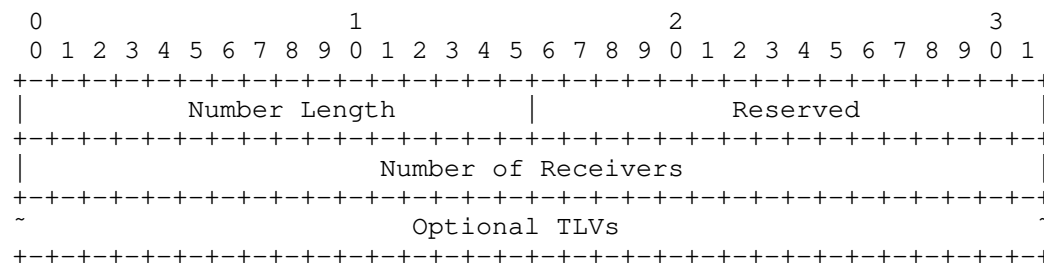


Figure 9: MRS Object Body Format

Number Length(16 bits): indicates the length of receiver number.

Number of Receivers(32 bits): indicates the number of receivers for a particular (S,G) tuple.

MRS object should include two types of TLVs, namely Multicast Source Address TLV Section 7.1.1 and Multicast Group Address TLV Section 7.2.1.

8. Procedures

8.1. Multicast source registration and revocation

For PCC-Registered multicast source, an ingress node sends a PCRpt message with MSR object to a stateful PCE, where R flag is set and A flag is not set. The registered authentication information can be passed through auxiliary data in MSR object.

Upon receiving the registration via PCRpt message, the stateful PCE MUST match local authentication rules based on the multicast information and auxiliary data in PCRpt message. If authenticated successfully, the PCE stores the multicast registration information into the database. In response, PCE MUST send a PCUpd message with MSR object to ingress node, where R flag is set. A flag is set only if authentication is successful.

For PCC-revoked multicast source registration, an ingress node sends a PCRpt message with MSR object to a stateful PCE, where R flag is not set and A flag is set.

Upon receiving the revocation via PCRpt message, in response, PCE MUST send a PCUpd message with MSR object to ingress node, where neither R nor A is set.

8.2. Joining and leaving of multicast receivers

When an egress node receives an IGMP or MLD message from a multicast receiver to join, the egress node should send a PCRpt message with MRI object to the PCE if no other receiver has sent the same request to it before.

If it is not the first time the PCE has received the same PCRpt message for join from the same egress node, this message should be ignored.

When an egress node receives an IGMP or MLD message from a multicast receiver to leave, the egress node should send a PCRpt message with MRI object and MRS object to the PCE if there are no other members in the requested multicast group. In MRS object, the number of receivers is zero.

8.3. BitString management

Upon receiving the join or leave request via PCRpt message, PCE needs to combine the BFR-id and SI of the egress node carried in PCRpt message with the BFR-id and SI of the ingress node and existed BitStrings in the database to create or update BitString. If there are members in the multicast group, the PCE should send a PCUpd message with FI object carrying the latest BitString to the ingress node, where F flag is set.

When receiving multicast packets, the ingress node encapsulates BIER header and forwards them based on BIFT and BitString. Encapsulation of Forwarding Label is not in the scope of this document.

If there is no member in the multicast group, the PCE should send a PCUpd message with FI object to the ingress node, where F flag is not set.

8.4. Receiver information synchronization

Upon receiving multicast packets from a particular multicast group, egress node will synchronize the number of receivers in this multicast group with the PCE via PCRpt message with MRS object periodically.

After sending a PCUpd message with FI object to an ingress node for a particular multicast group, the PCE will synchronize the total number of receivers in this multicast group with the ingress node via PCUpd message with MRS object periodically.

If there is no member in the multicast group, the synchronization of receiver number information ends.

9. Deployment Considerations

10. Security Considerations

11. IANA Considerations

11.1. BIER-MULTICAST-CAPABILITY

IANA is requested to allocate a new code point within registry "STATEFUL-PCE-CAPABILITY TLV Flag Field" under "Path Computation Element Protocol (PCEP) Numbers" as follows:

Value	Description	Reference
TBD1	BIER-MULTICAST-CAPABILITY	This document

11.2. PCEP-ERROR Object

IANA is requested to allocate code-points in the "PCEP-ERROR Object Error Types and Values" subregistry for the following new error-type and error-value:

Error-Type	Description	Reference
10	Error-value = TBD2 B bit is not set	This document

11.3. New Objects

IANA is requested to allocate the following Object-Class Values in the "PCEP Objects" subregistry under the "Path Computation Element Protocol (PCEP) Numbers" registry:

Object-Class Value	Description	Reference
TBD3	Multicast Receiver Information	This document
TBD7	Multicast Receiver Information	This document
TBD9	Forwarding Indication	This document
TBD10	Multicast Receiver Status	This document

11.4. New TLVs

IANA is requested to allocate the following Object-Class Values in the "PCEP Objects" subregistry under the "Path Computation Element Protocol (PCEP) Numbers" registry:

Type	Description	Reference
TBD4	Multicast Source Address	This document
TBD5	Multicast Group Address	This document
TBD6	BIER Information TLV	This document
TBD8	VPN Information	This document

12. Contributor

13. Acknowledgement

14. 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>>.
- [RFC2362] Estrin, D., Farinacci, D., Helmy, A., Thaler, D., Deering, S., Handley, M., Jacobson, V., Liu, C., Sharma, P., and L. Wei, "Protocol Independent Multicast-Sparse Mode (PIM-SM): Protocol Specification", RFC 2362, DOI 10.17487/RFC2362, June 1998, <<https://www.rfc-editor.org/info/rfc2362>>.
- [RFC4655] Farrel, A., Vasseur, J., and J. Ash, "A Path Computation Element (PCE)-Based Architecture", RFC 4655, DOI 10.17487/RFC4655, August 2006, <<https://www.rfc-editor.org/info/rfc4655>>.
- [RFC6514] Aggarwal, R., Rosen, E., Morin, T., and Y. Rekhter, "BGP Encodings and Procedures for Multicast in MPLS/BGP IP VPNs", RFC 6514, DOI 10.17487/RFC6514, February 2012, <<https://www.rfc-editor.org/info/rfc6514>>.
- [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>>.
- [RFC8231] Crabbe, E., Minei, I., Medved, J., and R. Varga, "Path Computation Element Communication Protocol (PCEP) Extensions for Stateful PCE", RFC 8231, DOI 10.17487/RFC8231, September 2017, <<https://www.rfc-editor.org/info/rfc8231>>.

- [RFC8232] Crabbe, E., Minei, I., Medved, J., Varga, R., Zhang, X., and D. Dhody, "Optimizations of Label Switched Path State Synchronization Procedures for a Stateful PCE", RFC 8232, DOI 10.17487/RFC8232, September 2017, <<https://www.rfc-editor.org/info/rfc8232>>.
- [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>>.
- [RFC8281] Crabbe, E., Minei, I., Sivabalan, S., and R. Varga, "Path Computation Element Communication Protocol (PCEP) Extensions for PCE-Initiated LSP Setup in a Stateful PCE Model", RFC 8281, DOI 10.17487/RFC8281, December 2017, <<https://www.rfc-editor.org/info/rfc8281>>.
- [RFC8296] Wijnands, IJ., Ed., Rosen, E., Ed., Dolganow, A., Tantsura, J., Aldrin, S., and I. Meilik, "Encapsulation for Bit Index Explicit Replication (BIER) in MPLS and Non-MPLS Networks", RFC 8296, DOI 10.17487/RFC8296, January 2018, <<https://www.rfc-editor.org/info/rfc8296>>.

Authors' Addresses

Huanan Li
China Telecom
Beiqijia Town, Changping District
Beijing, Beijing 102209
China

Email: lihn6@foxmail.com

Aijun Wang
China Telecom
Beiqijia Town, Changping District
Beijing, Beijing 102209
China

Email: wangaj3@chinatelecom.cn

Huaimo Chen
Futurewei
Boston
USA

Email: Huaimo.chen@futurewei.com

Ran Chen
ZTE Corporation
50 Software Avenue, Yuhua District
Nanjing, Jiangsu 210012
China

Email: chen.ran@zte.com.cn

BIER Working Group
Internet-Draft
Intended status: Standards Track
Expires: September 22, 2022

W. Wang
A. Wang
China Telecom
H. Chen
Futurewei
G. Mishra
Verizon Inc.
B. Xu
Huawei Technologies (2012Lab)
March 21, 2022

Routing Header Based BIER Information Encapsulation
draft-wang-bier-rh-bier-05

Abstract

This draft proposes one new encapsulation schema of Bit Index Explicit Replication (BIER) information to transfer the multicast packets within the IPv6 network. By using a new type of IPv6 Routing Header to forward the packet, the original source address and destination address of the multicast packet is kept unchanged along the forwarding path. Such encapsulation schema can make full use of the existing IPv6 quality assurance solutions to provide high-quality multicast service.

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 22, 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 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. BIER Routing Header	3
4. Multicast Packet Forwarding Procedures	4
4.1. All nodes in BIER domain support BIER Routing Header	5
4.2. Some nodes in BIER domain do not support BIER Routing Header	7
5. Security Considerations	9
6. IANA Considerations	9
7. References	9
7.1. Normative References	9
7.2. Informative References	9
Authors' Addresses	10

1. Introduction

Bit Index Explicit Replication (BIER) is a new multicast technology based on IPv6 defined in [RFC8279]. In BIER domain, the set of destination nodes of multicast message is mapped into a BitString and encapsulated into the BIER header. The position of each bit in the BitString represents an BFER. Compared with the traditional multicast technologies, the nodes in BIER domain do not need to maintain a multicast tree and keep the multicast flow state for each multicast flow.

Currently, there are two methods for encapsulating BIER information based on IPv6 in IETF: BIERin6([I-D.ietf-bier-bierin6]) and BIERv6([I-D.xie-bier-ipv6-encapsulation]).

BIERin6 carries BIER information by defining a new IPv6 next header type. During the forwarding process, the source address and destination address in the header will be changed.

BIERv6 carries bier related information by defining an new type of destination options header (i.e. bier option). The source address in

the header remains unchanged but the destination address will be changed along the forwarding path.

The differences between the above two BIER encapsulation and forwarding schemes are unfavorable for the development of BIER and its derivatives. In addition, when there is error in the forward process of the multicast packet, the change of source address and destination address during transmission will increase the difficulty of fault location and traceability.

This draft proposes a BIER information transmission scheme without changing the multicast source and destination addresses in the outer IPv6 header. The relevant BIER information is encapsulated within the newly defined IPv6 Routing Header type, each intermediate BIER router will route the multicast packet based on the BitString information and its associated BIFT. The multicast source and destination address are not changed along the forwarding path.

The characteristics of such schema are helpful to the rapid fault location and traceability, and can make full use of the existing IPv6 quality assurance technologies to provide high-quality multicast service.

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. BIER Routing Header

One new type of IPv6 Routing Header is defined according to [RFC8200]. The message format is shown in Figure 1.

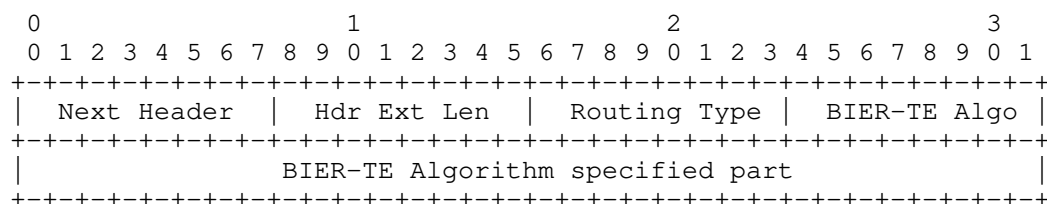


Figure 1: The format of BIER Routing Header

Where:

- o Next Header (8 bits): indicating the message header type immediately after the Routing Header.

- o HDR Ext Len (8 bits): indicating the length of the Routing Header.
- o Routing Type (8 bits): TBD. Identifying the newly defined Routing Header to encode BIER information.
- o BIER-TE Algo (8 bits): indicating the BIER-TE Algorithm for RH-BIER packets. Different values of this field refer to different BIER-TE Algorithm:
 - * Value 0: reserved.
 - * Value 1: IGP
 - * Value 2: CGM2 (see [I-D.eckert-bier-cgm2-rbs])
 - * Value 3: MRH (see [I-D.chen-pim-mrh6])
 - * Value 4-127: Expert Reviews
 - * Value 128-255: Flexible Algorithms
- o BIER-TE Algorithm specified part (variable): the encoding of this part depends on the value of BIER-TE Algo field:
 - * Value 1: The BIER-TE Algorithm is IGP, the encoding format of this part is described in [RFC8296].
 - * Value 2: The BIER-TE Algorithm is CGM2, the encoding format of this part is described in [I-D.eckert-bier-cgm2-rbs].
 - * Value 3: The BIER-TE Algorithm is MRH6, the encoding format of this part is described in [I-D.chen-pim-mrh6].

4. Multicast Packet Forwarding Procedures

Based on the newly defined BIER Routing Header, the nodes support BIER Routing Header will perform the following steps to forward the multicast packets:

1) When a BFIR receive a multicast packet, it will find out the destination address and RD that relate to the source interface of the packet. BFIR looks up its End.MVPN mapping table to find the associated End.MVPN, and encapsulate a IPv6 Header with BIER Routing Header. The payload is user data, the source address is the IPv6 address of BFIR, and destination address is End.MVPN. BitString in BIER Routing Header indicates the BFERs that want to receives such multicast packet.

2) BFIR checks whether there is BIFT corresponding to the BIFT-id locally. If not, it will discard the packet; otherwise, it will check whether the direct-connected node support BIER Routing Header. If the direct-connected node supports BIER Routing Header, proceeding to step 3). If the direct-connected node doesn't support BIER Routing Header, proceeding to step 2.1) .

2.1) BFIR Calculates the IPv6 address of next hop that support BIER Routing Header.

2.2) Encapsulating an outer IPv6 Header to the multicast packet. The calculated IPv6 address is used as the destination address of the outer IPv6 Header, and its own IPv6 address is used as the source address of the outer IPv6 Header. BitString will not be changed.

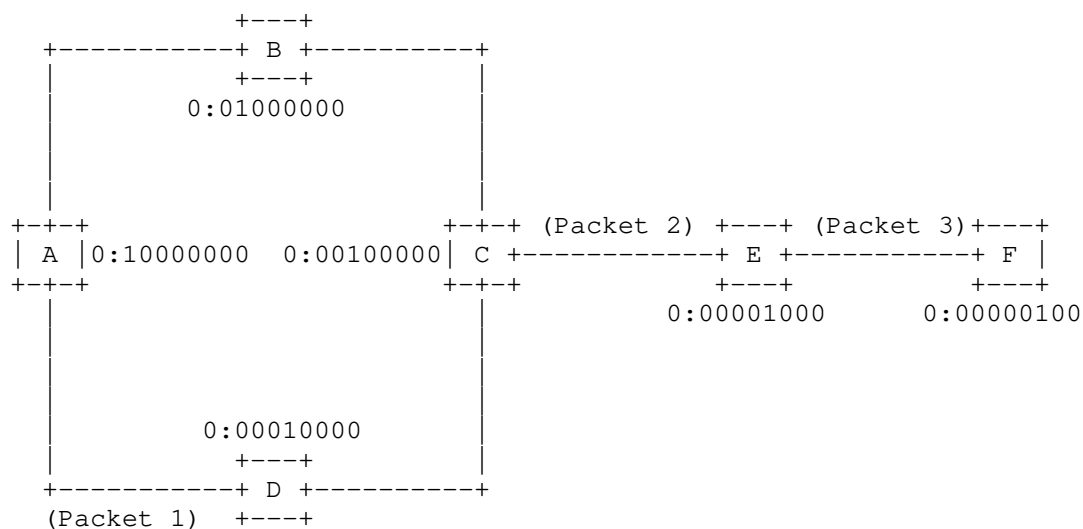
2.3) Sending the encapsulated packet to the direct-connected node, the node will perform normal IPv6 forwarding according to the outer IPv6 Header.

3) Performing the normal BIER forwarding process as described in [RFC8279].

For a BFR, it performs as described in Section 4.2.

The detail procedures for forwarding the multicast packets based on the newly defined Routing Header are described in the following sections.

4.1. All nodes in BIER domain support BIER Routing Header



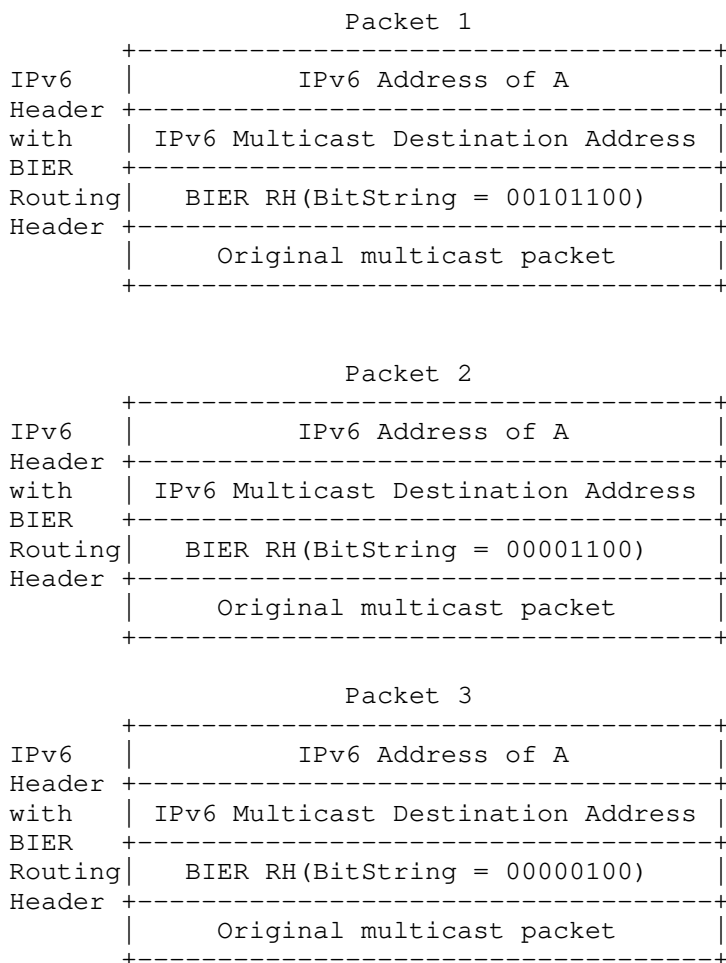


Figure 2: All nodes in BIER domain support BIER Routing Header

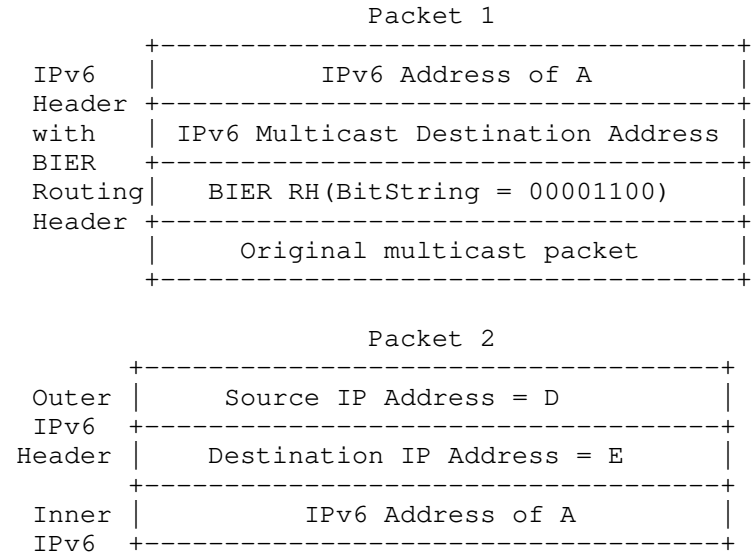
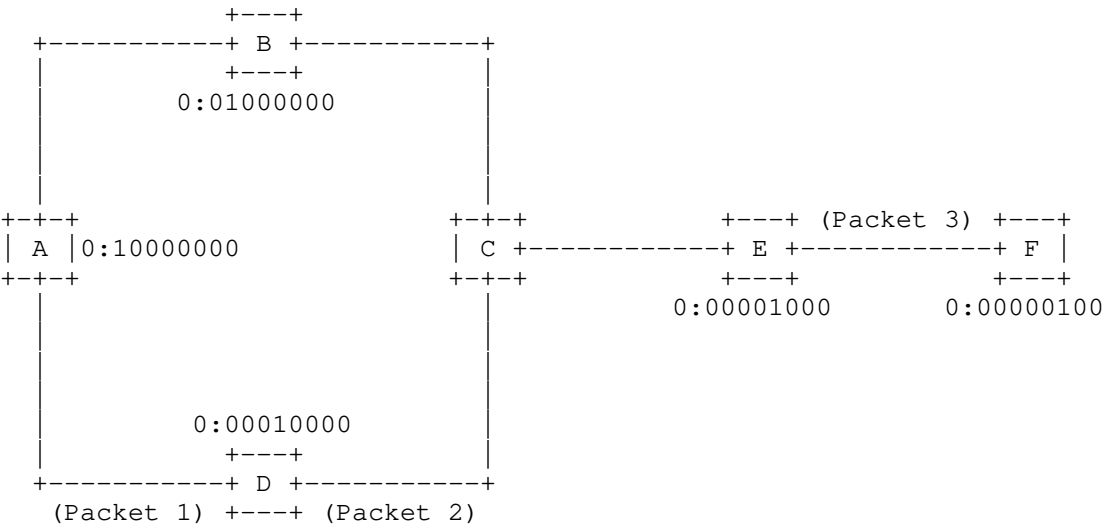
The topology is shown in Figure 2, node A-F support BIER Routing Header. The packet need to be transmitted from A to F. The changes of the Routing Header have been given in Figure 2.

- 1). Node A is BFIR, when it receives a multicast packet, it will encapsulate a IPv6 Header with BIER Routing Header to the packet.
- 2). Node A checks whether there is BIFT corresponding to the BIFT-id locally. If not, discarding the packet; otherwise, forwarding the packet according to the BIFT related to the BIFT-id.
- 3). Node D-E repeat the step 2).

4). Node F looks up the associated table and submits the packet to the new multicast downstreams.

During the forwarding procedures, the source & destination address in IPv6 header are not changed, only the BitString in BIER Routing Header is updated.

4.2. Some nodes in BIER domain do not support BIER Routing Header



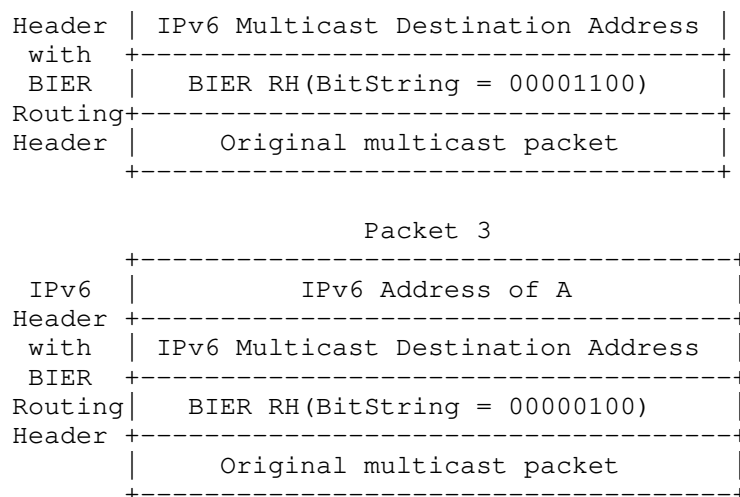


Figure 3: Some nodes in BIER domain do not support BIER Routing Header

The topology is shown in Figure 3, all nodes expect node C support BIER Routing Header. The packet need to be transmitted from A to F. The change of the Header has been given in the Figure 3.

- 1). After receiving a multicast packet, node A encapsulates a IPv6 Header with BIER Routing Header to it, and forwards the packet to node D according to the BIFT.
- 2). Node D calculates the IPv6 address of next hop node(Node E) that supports BIER Routing Header, and encapsulates an outer IPv6 Header to the packet. The source IPv6 address is the IPv6 address of itself, and the destination IPv6 address is the IPv6 address of node E. Then, sending the packet to node C.
- 3). Node C performs normal IPv6 forwarding according to the outer IPv6 header and sends the packet to node E.
- 4). Node E decapsulates the outer IPv6 header and forwards the packet according to the BIFT to node F.
- 5). Node F looks up the associated table and submits the packet to the new multicast downstreams.

In the forwarding procedures, the source address and destination address in the Inner IPv6 Header are not changed, only the BitString in BIER Routing Header is updated.

5. Security Considerations

TBD

6. IANA Considerations

This document defines a new type of IPv6 Routing Header - BIER Routing Header. The code point is from the "Internet Protocol Version 6 (IPv6) Parameters - Routing Types". It is recommended to set the code point of BIER Routing Header to 7.

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>>.
- [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>>.

7.2. Informative References

- [I-D.chen-pim-mrh6] Chen, H., McBride, M., Fan, Y., Li, Z., Geng, X., Toy, M., Mishra, G. S., Liu, Y., Wang, A., Liu, L., and X. Liu, "Multicast using Multicast Routing Header", draft-chen-pim-mrh6-01 (work in progress), March 2022.

[I-D.eckert-bier-cgm2-rbs]

Eckert, T. and B. (. Xu, "Carrier Grade Minimalist Multicast (CGM2) using Bit Index Explicit Replication (BIER) with Recursive BitString Structure (RBS) Addresses", draft-eckert-bier-cgm2-rbs-01 (work in progress), February 2022.

[I-D.ietf-bier-bierin6]

Zhang, Z., Zhang, Z., Wijnands, I., Mishra, M., Bidgoli, H., and G. Mishra, "Supporting BIER in IPv6 Networks (BIERin6)", draft-ietf-bier-bierin6-04 (work in progress), March 2022.

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

Xie, J., Geng, L., McBride, M., Asati, R., Dhanaraj, S., Zhu, Y., Qin, Z., Shin, M., Mishra, G., and X. Geng, "Encapsulation for BIER in Non-MPLS IPv6 Networks", draft-xie-bier-ipv6-encapsulation-10 (work in progress), February 2021.

Authors' Addresses

Wei Wang
China Telecom
Beiqijia Town, Changping District
Beijing, Beijing 102209
China

Email: weiwang94@foxmail.com

Aijun Wang
China Telecom
Beiqijia Town, Changping District
Beijing, Beijing 102209
China

Email: wangaj3@chinatelecom.cn

Huaimo Chen
Futurewei
Beiqijia Town, Changping District
Boston, MA
USA

Email: Huaimo.chen@futurewei.com

Gyan S. Mishra
Verizon Inc.
13101 Columbia Pike
Silver Spring MD 20904
United States of America

Phone: 301 502-1347
Email: gyan.s.mishra@verizon.com

Bing (Robin) Xu
Huawei Technologies (2012Lab)
Huawei Building, No.156 Beiqing Rd.
Beijing, Beijing 100095
China

Email: bing.xu@huawei.com

bier
Internet-Draft
Intended status: Standards Track
Expires: April 2, 2022

Z. Zhang
A. Przygienda
Juniper Networks
September 29, 2021

BIER with Network Slicing and Flow Differentiation
draft-zzhang-bier-slicing-and-differentiation-00

Abstract

This document specifies how BIER works in the context of IETF Network slicing, with or without fined-grained traffic differentiation.

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, 2022.

Copyright Notice

Copyright (c) 2021 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. BIER with IETF Network Slicing	3
3. BIER with Slice Aggregates	4
4. Specifications	4
4.1. ISIS Signaling	4
4.1.1. OSPF Signaling	5
4.1.2. BGP Signaling	5
4.2. BIER Extension Header	5
5. Security Considerations	5
6. IANA Considerations	6
7. References	6
7.1. Normative References	6
7.2. Informative References	6
Authors' Addresses	7

1. Introduction

Network slicing has been a topic widely discussed in and beyond IETF. According to [I-D.ietf-teas-ietf-network-slices]:

"An IETF Network Slice is a logical network topology connecting a number of endpoints using a set of shared or dedicated network resources that are used to satisfy specific Service Level Objectives (SLOs).

An IETF Network Slice combines the connectivity resource requirements and associated network behaviors such as bandwidth, latency, jitter, and network functions with other resource behaviors such as compute and storage availability."

It is expected that traffic associated with an IETF network slice is identified with a slice identifier (e.g. an MPLS label) and each node in the path uses the slice identifier to identify the slice in which the traffic is forwarded.

[I-D.bestbar-teas-ns-packet] introduces the notion of Slice Aggregate which comprises of one or more IETF network slice traffic streams. A Slice Aggregate is identified by a Slice Selector (SS), and packets carry the SS so that associated forwarding treatment or S-PHB (Slice policy Per Hop Behavior - the externally observable forwarding behavior applied to a specific packet belonging to a slice aggregate) - can be applied along the path.

[I-D.li-apn-problem-statement-usecases] describes challenges faced by network operators when attempting to provide fine-grained traffic operations to satisfy the various requirements demanded by new

applications that require differentiated service treatment and [I-D.li-apn-framework] proposes a framework for solution:

"... proposes a new framework, named Application-aware Networking (APN), where application-aware information (i.e. APN attribute) including APN identification (ID) and/or APN parameters (e.g. network performance requirements) is encapsulated at network edge devices and carried in packets traversing an APN domain in order to facilitate service provisioning, perform fine-granularity traffic steering and network resource adjustment."

The authors of this document believe that the IETF Network Slicing framework, when augmented by the Slice Aggregate, addresses the APN problem domain very well. This document describes how BIER [RFC8279] works together with IETF network slicing, with or without Slice Aggregate to provide fine granularity traffic differentiation (e.g. down to per-flow level) that is demanded in the APN problem statement.

2. BIER with IETF Network Slicing

Since an IETF Network Slice is a logical network topology, each slice may have its BIRT (which maps to a set of BIFTs when BitStringLength and SetID are considered). While it is tempting and seems logical to map a slice to a BIER sub-domain, and it is straightforward to do so when the number of slices is smaller than 256 (the max number of sub-domains), this document allows to map a slice directly to a BIRT instead of a sub-domain.

Now a BIRT corresponds to a <sub-domain, slice> tuple, and each BIFT corresponds to a <subdomain-id, slice-id, bitstring length, set-id> tuple. In forwarding plane a BIFT is only identified by a 20-bit opaque number locally on a BFR, which could be an MPLS label or just a plain number in case of non-MPLS data plane. Therefore, it is feasible to have many slices in the same sub-domain - each slice will have its own BIRT so that the same BFER in the same sub-domain can be reached via different nexthop BFRs according to different BIRTs (i.e. different set of corresponding BIFTs) for different slices.

With this, up to 2^{20} slices could be supported in theory - the only limit is the number of BIFT entries that a BFR can hold.

Mapping a slice directly to a BIRT instead of a sub-domain not only allows more than 256 slices but also reduces the burden of sub-domain related provisioning (e.g. a BFER-ID is needed for each <sub-domain, BFER/BFER>). Of course, as mentioned earlier, if the number of slices is smaller than 256 then a slice can map to a sub-domain as well.

3. BIER with Slice Aggregates

Per [I-D.bestbar-teas-ns-packet], a Slice Aggregate may be the aggregation of several entire slices, or just a particular flow in a slice. With a Slice Aggregate for several entire slices, the different slices (of the same Slice Aggregate) also map to the same BIRT. In that case, for the same destination BFER, traffic in those different slices are forwarded to the same (set of ECMP) nexthop BFER according to the shared BIRT, yet other forwarding treatment (e.g. queuing) could still be different.

In [RFC8279], a sub-domain is associated with only one topology and each sub-domain has its own BIRT calculated using the topology information. When multiple slices are associated with a single sub-domain, each slice (or a set of slices) also has its own BIRT calculated based on the slice's (or the set of slices') topology information. Therefore, having a sub-domain with multiple slices does not violate the underlying principle of BIER architecture, i.e., a BIRT is calculated on a corresponding topology, whether the topology is for a sub-domain as in [RFC8279] or for a <sub-domain, slice or set of slices> tuple as in this document.

The BIER header has a 6-bit DSCP field. If that is not enough to identify different slices or slice aggregates that share the same BIRT, an explicit Slice Selector can be carried in "BIER Extension Header" [I-D.zzhang-intarea-generic-delivery-functions].

This means that, even for a transit BFR, if provisioned to support slice aggregates identified by a Slice Selector in the extension header, it must check if the "Proto" field is set to a value for BIER Extension Header.

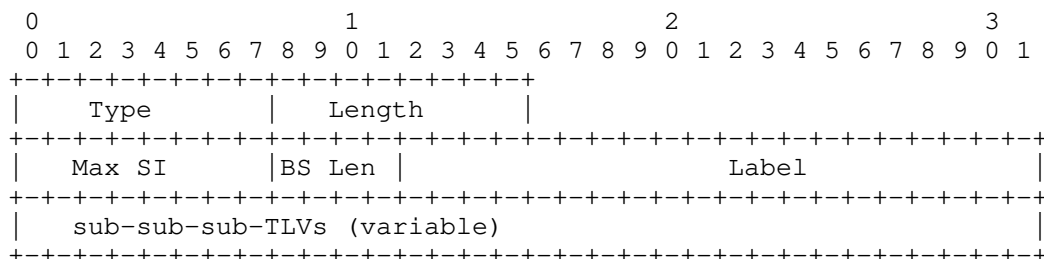
Note: while the concept of "BIER Extension Header" is first brought up in that Generic Delivery Functions draft [I-D.zzhang-intarea-generic-delivery-functions] in intarea WG, it is expected that BIER specific work will be brought to the BIER WG.

4. Specifications

BIER signaling for OSPF/ISIS/BGP is extended to include slice information so that slice-specific BIRTs can be built.

4.1. ISIS Signaling

A BIER MPLS Encapsulation Extended Sub-sub-TLV is defined with a new type to allow sub-sub-sub-TLVs in it. Besides the new type and additional sub-sub-sub-TLVs, the rest are the same as original BIER MPLS Encapsulation Sub-sub-TLV [RFC8401].



Type: Value of TBD indicating Extended sub-sub-TLV for MPLS

Length: Variable

Sub-sub-sub-TLVs: for information like Slice Selector

Sub-sub-sub-TLVs will be defined to include Slice Selector information [I-D.bestbar-teas-ns-packet] that identifies a slice or a Slice Aggregate, and potentially other information. Note that the Slice Aggregate here is for a set of slices instead of a flow in a slice. Future revisions will have more details.

Similar encoding will be defined for non-MPLS encapsulation in future revisions.

4.1.1. OSPF Signaling

Similar encoding will be defined for OSPF signaling in future revisions.

4.1.2. BGP Signaling

Similar encoding will be defined for BGP signaling in future revisions.

4.2. BIER Extension Header

This will be tracked by a separate BIER draft. For now, please refer to [I-D.zzhang-intarea-generic-delivery-functions].

5. Security Considerations

To be provided.

6. IANA Considerations

To be provided.

7. References

7.1. Normative References

[I-D.bestbar-teas-ns-packet]

Saad, T., Beeram, V. P., Wen, B., Ceccarelli, D., Halpern, J., Peng, S., Chen, R., Liu, X., Contreras, L. M., and R. Rokui, "Realizing Network Slices in IP/MPLS Networks", draft-bestbar-teas-ns-packet-03 (work in progress), July 2021.

[I-D.zzhang-intarea-generic-delivery-functions]

Zhang, Z., Bonica, R., Kompella, K., and G. Mirsky, "Generic Delivery Functions", draft-zzhang-intarea-generic-delivery-functions-02 (work in progress), August 2021.

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

[RFC8401] Ginsberg, L., Ed., Przygienda, T., Aldrin, S., and Z. Zhang, "Bit Index Explicit Replication (BIER) Support via IS-IS", RFC 8401, DOI 10.17487/RFC8401, June 2018, <<https://www.rfc-editor.org/info/rfc8401>>.

7.2. Informative References

[I-D.ietf-teas-ietf-network-slices]

Farrel, A., Gray, E., Drake, J., Rokui, R., Homma, S., Makhijani, K., Contreras, L. M., and J. Tantsura, "Framework for IETF Network Slices", draft-ietf-teas-ietf-network-slices-04 (work in progress), August 2021.

[I-D.li-apn-framework]

Li, Z., Peng, S., Voyer, D., Li, C., Liu, P., Cao, C., Mishra, G., Ebisawa, K., Previdi, S., and J. N. Guichard, "Application-aware Networking (APN) Framework", draft-li-apn-framework-03 (work in progress), May 2021.

[I-D.li-apn-problem-statement-usecases]

Li, Z., Peng, S., Voyer, D., Xie, C., Liu, P., Qin, Z.,
Mishra, G., Ebisawa, K., Previdi, S., and J. N. Guichard,
"Problem Statement and Use Cases of Application-aware
Networking (APN)", draft-li-apn-problem-statement-
usecases-04 (work in progress), June 2021.

Authors' Addresses

Zhaohui Zhang
Juniper Networks

Email: zzhang@juniper.net

Antoni Przygienda
Juniper Networks

Email: prz@juniper.net