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

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.
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 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 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 BFER 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.

```plaintext
+-----------------+-----------------+-----------------+-----------------+
|     Type (TBD2)  |     Length      |   Reserved      |
|   +-----------------+-----------------+-----------------+-----------------+ |
| BFR-id of backup egress node | Sub-TLVs (Optional) |
|                                                  +-----------------+
```

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 BFERs. 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 BFERs.

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 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.

<table>
<thead>
<tr>
<th>BFR-id (SI:BitString)</th>
<th>F-BM</th>
<th>BFR-NBR</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 (0:00001)</td>
<td>10111</td>
<td>C</td>
</tr>
<tr>
<td>2 (0:00010)</td>
<td>10111</td>
<td>C</td>
</tr>
<tr>
<td>3 (0:00100)</td>
<td>10111</td>
<td>C</td>
</tr>
<tr>
<td>4 (0:01000)</td>
<td>01000</td>
<td>H</td>
</tr>
<tr>
<td>5 (0:10000)</td>
<td>10111</td>
<td>C</td>
</tr>
</tbody>
</table>

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.

```
<table>
<thead>
<tr>
<th>BFR-id (SI:BitString)</th>
<th>F-BM</th>
<th>BFR-NBR</th>
<th>BEA</th>
<th>SC</th>
<th>BE-BFER</th>
<th>BF-BM</th>
<th>BBFR-NBR /P-FIB</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 (0:00001)</td>
<td>1011</td>
<td>C</td>
<td>0</td>
<td>0</td>
<td>H(01000)</td>
<td>00001</td>
<td>P-FIB-4D</td>
</tr>
<tr>
<td>2 (0:00010)</td>
<td>1011</td>
<td>C</td>
<td>0</td>
<td>0</td>
<td>NULL</td>
<td></td>
<td>NULL</td>
</tr>
<tr>
<td>3 (0:00100)</td>
<td>1011</td>
<td>C</td>
<td>0</td>
<td>0</td>
<td>NULL</td>
<td></td>
<td>NULL</td>
</tr>
<tr>
<td>4 (0:01000)</td>
<td>0100</td>
<td>H</td>
<td>0</td>
<td>0</td>
<td>NULL</td>
<td></td>
<td>NULL</td>
</tr>
<tr>
<td>5 (0:10000)</td>
<td>1011</td>
<td>C</td>
<td>0</td>
<td></td>
<td>NULL</td>
<td></td>
<td>NULL</td>
</tr>
</tbody>
</table>
```

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 BFIT 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.

<table>
<thead>
<tr>
<th>BFR-id (SI:BitString)</th>
<th>F-BM</th>
<th>BFR-NBR</th>
<th>SC</th>
<th>BE-BFER</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 (0:00001)</td>
<td>01001</td>
<td>H</td>
<td>1</td>
<td>H(01000)</td>
</tr>
<tr>
<td>2 (0:00010)</td>
<td>00110</td>
<td>F</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3 (0:00100)</td>
<td>00110</td>
<td>F</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4 (0:01000)</td>
<td>01001</td>
<td>H</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5 (0:10000)</td>
<td>10000</td>
<td>B</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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.

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

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     |  BFR-Prefix  |  BFR-NBR   |
| (SI:BitString) | of Dest BFER | (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.

<table>
<thead>
<tr>
<th>BFR-id (SI:BitString)</th>
<th>F-BM</th>
<th>BFR-NBR (Next Hop)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 (0:00001)</td>
<td>00001</td>
<td>D</td>
</tr>
<tr>
<td>2 (0:00010)</td>
<td>00110</td>
<td>F</td>
</tr>
<tr>
<td>3 (0:00100)</td>
<td>00110</td>
<td>F</td>
</tr>
<tr>
<td>4 (0:01000)</td>
<td>01000</td>
<td>H</td>
</tr>
<tr>
<td>5 (0:10000)</td>
<td>10000</td>
<td>B</td>
</tr>
</tbody>
</table>

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 (BFERs 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.

<table>
<thead>
<tr>
<th>BFR-id (SI:BitString)</th>
<th>BFR-Prefix of Dest BFER</th>
<th>BFR-NBR (Next Hop)</th>
<th>BE-BFER</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 (0:00001)</td>
<td>D</td>
<td>H</td>
<td>D</td>
</tr>
<tr>
<td>2 (0:00010)</td>
<td>F</td>
<td>F</td>
<td>0</td>
</tr>
<tr>
<td>3 (0:00100)</td>
<td>E</td>
<td>F</td>
<td>0</td>
</tr>
<tr>
<td>4 (0:01000)</td>
<td>H</td>
<td>H</td>
<td>0</td>
</tr>
<tr>
<td>5 (0:10000)</td>
<td>A</td>
<td>B</td>
<td>0</td>
</tr>
</tbody>
</table>

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).

<table>
<thead>
<tr>
<th>BFR-id (SI:BitString)</th>
<th>F-BM</th>
<th>BFR-NBR (Next Hop)</th>
<th>SC</th>
<th>BE-BFER</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 (0:00001)</td>
<td>01001</td>
<td>H</td>
<td>1</td>
<td>H</td>
</tr>
<tr>
<td>2 (0:00010)</td>
<td>00110</td>
<td>F</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3 (0:00100)</td>
<td>00110</td>
<td>F</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4 (0:01000)</td>
<td>01001</td>
<td>H</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5 (0:10000)</td>
<td>10000</td>
<td>B</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

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 sends 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


9.2. Informative References

[I-D.ietf-rtgwg-segment-routing-ti-lfa]


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

Mike McBride
Futurewei
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
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.

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 of 5 bits. BFR-ids 1, 2, 3, 4, 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.
### Figure 3: BIRT on BFR B

<table>
<thead>
<tr>
<th>BFR-id (SI:BitString)</th>
<th>BFR-Prefix of Dest BFER</th>
<th>BFR-NBR (Next Hop)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 (0:00001)</td>
<td>D</td>
<td>C</td>
</tr>
<tr>
<td>2 (0:00010)</td>
<td>F</td>
<td>C</td>
</tr>
<tr>
<td>3 (0:01000)</td>
<td>E</td>
<td>E</td>
</tr>
<tr>
<td>4 (0:01000)</td>
<td>H</td>
<td>C</td>
</tr>
<tr>
<td>5 (0:10000)</td>
<td>A</td>
<td>A</td>
</tr>
</tbody>
</table>

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.
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.
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     |  F-BM   |  BFR-NBR   |
| (SI:BitString) |         | (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.

<table>
<thead>
<tr>
<th>BFR-id</th>
<th>F-BM</th>
<th>BFR-NBR</th>
</tr>
</thead>
<tbody>
<tr>
<td>(SI:BitString)</td>
<td>(Next Hop)</td>
<td></td>
</tr>
<tr>
<td>1, 4 (0:01001)</td>
<td>01001</td>
<td>G</td>
</tr>
<tr>
<td>2, 3 (0:00110)</td>
<td>00110</td>
<td>E</td>
</tr>
<tr>
<td>5 (0:10000)</td>
<td>10000</td>
<td>A</td>
</tr>
</tbody>
</table>

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


7.2. Informative References

[I-D.ietf-rtgwg-segment-routing-ti-lfa]

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

[I-D.merling-bier-frr]
Merling, D. and M. Menth, "BIER Fast Reroute", draft-merling-bier-frr-00 (work in progress), March 2019.

[RFC8296]


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
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.
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.
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 8 nodes/BFRs A, B, C, D, E, F, G and H. Nodes/BFRs D, F, E, H and A are BFERs and have local decap adjacency BitPositions (BPs for short) 1, 2, 3, 4, and 5 respectively. For simplicity, these BPs are represented by (SI:BitString), where SI = 0 and BitString is of 8 bits. BPs 1, 2, 3, 4, and 5 are represented by 1 (0:00000001), 2 (0:00000010), 3 (0:00000100), 4 (0:00001000) and 5 (0:00010000) 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 16. 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 16) are represented by 1'(6:00000001), 2'(6:00000010), 3'(6:00000010), 4'(6:00000100), 5'(6:00001000), 6'(6:00010000), 7'(6:01000000), 8'(6:10000000), 9'(7:00000001), 10'(7:00000010), . . . , 16'(7:10000000).
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 13', 14', 15' and 16' 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 8 nodes/BFRs A, B, C, D, E, F, G and H 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 two adjacencies of A. One is the forward connected adjacency from A to B (represented by BP 2'); the other is the local decap adjacency (represented by BP 5) for BFER (i.e., egress) A. The BIFT on A has two forwarding entries.
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 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.

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.

<table>
<thead>
<tr>
<th>Adjacency BP (SI:BitString)</th>
<th>Action</th>
<th>BFR-NBR (Next Hop)</th>
</tr>
</thead>
<tbody>
<tr>
<td>5’ (6:00010000)</td>
<td>fw-connected</td>
<td>B</td>
</tr>
<tr>
<td>12’ (7:00001000)</td>
<td>fw-connected</td>
<td>F</td>
</tr>
<tr>
<td>14’ (7:00100000)</td>
<td>fw-connected</td>
<td>G</td>
</tr>
<tr>
<td>15’ (7:01000000)</td>
<td>fw-connected</td>
<td>H</td>
</tr>
<tr>
<td>16’ (7:10000000)</td>
<td>fw-connected</td>
<td>D</td>
</tr>
</tbody>
</table>
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.

<table>
<thead>
<tr>
<th>Adjacency BP (SI:BitString)</th>
<th>Action</th>
<th>BFR-NBR (Next Hop)</th>
</tr>
</thead>
<tbody>
<tr>
<td>13' (7:00010000)</td>
<td>fw-connected</td>
<td>C</td>
</tr>
<tr>
<td>14' (7:00100000)</td>
<td>fw-connected</td>
<td>G</td>
</tr>
<tr>
<td>15' (7:01000000)</td>
<td>fw-connected</td>
<td>H</td>
</tr>
<tr>
<td>1 (0:00000001)</td>
<td>local-decap</td>
<td></td>
</tr>
</tbody>
</table>

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 F (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.

<table>
<thead>
<tr>
<th>Adjacency BP (SI:BitString)</th>
<th>Action</th>
<th>BFR-NBR (Next Hop)</th>
</tr>
</thead>
<tbody>
<tr>
<td>7' (6:01000000)</td>
<td>fw-connected</td>
<td>B</td>
</tr>
<tr>
<td>10' (7:00000010)</td>
<td>fw-connected</td>
<td>F</td>
</tr>
<tr>
<td>3 (0:00000100)</td>
<td>local-decap</td>
<td></td>
</tr>
</tbody>
</table>

Figure 4: BIER-TE BIFT on BFR C

Figure 5: BIER-TE BIFT on BFR D

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 three adjacencies of F. Two of them are the forward connected adjacencies from F to E (represented by BP 9') and F to C (represented by BP 11') respectively; the other is the local decap adjacency (represented by BP 2) for BFER (i.e., egress) F. The BIFT on F has three forwarding entries for these adjacencies.

<table>
<thead>
<tr>
<th>Adjacency BP (SI:BitString)</th>
<th>Action</th>
<th>BFR-NBR (Next Hop)</th>
</tr>
</thead>
<tbody>
<tr>
<td>9'(7:00000001)</td>
<td>fw-connected</td>
<td>E</td>
</tr>
<tr>
<td>11'(7:00000100)</td>
<td>fw-connected</td>
<td>C</td>
</tr>
<tr>
<td>2 (0:00000010)</td>
<td>local-decap</td>
<td></td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Adjacency BP (SI:BitString)</th>
<th>Action</th>
<th>BFR-NBR (Next Hop)</th>
</tr>
</thead>
<tbody>
<tr>
<td>3'(6:00000100)</td>
<td>fw-connected</td>
<td>B</td>
</tr>
<tr>
<td>13'(7:00010000)</td>
<td>fw-connected</td>
<td>C</td>
</tr>
<tr>
<td>15'(7:01000000)</td>
<td>fw-connected</td>
<td>H</td>
</tr>
<tr>
<td>16'(7:10000000)</td>
<td>fw-connected</td>
<td>D</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Adjacency BP (SI:BitString)</th>
<th>Action</th>
<th>BFR-NBR (Next Hop)</th>
</tr>
</thead>
<tbody>
<tr>
<td>13'(7:00010000)</td>
<td>fw-connected</td>
<td>C</td>
</tr>
<tr>
<td>14'(7:00100000)</td>
<td>fw-connected</td>
<td>G</td>
</tr>
<tr>
<td>16'(7:10000000)</td>
<td>fw-connected</td>
<td>D</td>
</tr>
<tr>
<td>4 (0:00001000)</td>
<td>local-decap</td>
<td></td>
</tr>
</tbody>
</table>

Figure 9: BIER-TE BIFT on BFR H

2.3. Example P2MP Path with LAN

This section presents the forwarding behaviors along an explicit P2MP
path in Figure 1 going through the LAN in the figure.

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 H (indicated by BP 15’), and the link/
adjacency from C to F (indicated by BP 12’). This path is
represented by {2’, 4’, 6’, 12’, 15’, 2, 4}. 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’, 15’, 2, 4}.
The packet received by C has path {12’, 15’, 2, 4}.

After receiving the packet from B, G sends the packet to H according
to the forwarding entry for BP 15’ 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 15’ 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 10.

![Figure 10: Example BIER-TE Topology with BPs for LAN](image-url)
The connection/adjacency from Px to C is assigned BP 13’, and the connection/adjacency from C to Px is assigned BP 14’.

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

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

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

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 10.

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 11. It comprises the forwarding entry for the LAN adjacency from C to Px (indicated by BP 14’), 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 15’), Px to H (indicated by 17’), and Px to D (indicated by 19’) respectively.

<table>
<thead>
<tr>
<th>Adjacency BP (SI:BitString)</th>
<th>Action</th>
<th>BFR-NBR (Next Hop)</th>
</tr>
</thead>
<tbody>
<tr>
<td>15’(7:01000000)</td>
<td>fw-connected</td>
<td>G</td>
</tr>
<tr>
<td>17’(8:00000001)</td>
<td>fw-connected</td>
<td>H</td>
</tr>
<tr>
<td>19’(8:00000100)</td>
<td>fw-connected</td>
<td>D</td>
</tr>
</tbody>
</table>

Figure 11: Improved BIER-TE BIFT on BFR C

The improved BIER-TE BIFT on BFR G is illustrated in Figure 12. It comprises the forwarding entry for the LAN adjacency from G to Px (indicated by BP 16’), 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 13’), Px to H (indicated by 17’), and Px to D (indicated by 19’) respectively.

<table>
<thead>
<tr>
<th>Adjacency BP (SI:BitString)</th>
<th>Action</th>
<th>BFR-NBR (Next Hop)</th>
</tr>
</thead>
<tbody>
<tr>
<td>5’(6:00010000)</td>
<td>fw-connected</td>
<td>B</td>
</tr>
<tr>
<td>12’(7:00001000)</td>
<td>fw-connected</td>
<td>F</td>
</tr>
<tr>
<td>14’(7:00100000)</td>
<td>lan-connected</td>
<td>Px</td>
</tr>
</tbody>
</table>
The improved BIER-TE BIFT on BFR H is illustrated in Figure 13. It comprises the forwarding entry for the LAN adjacency from H to Px (indicated by BP 18’) 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 13’), Px to G (indicated by 15’) and Px to D (indicated by 19’) respectively.
### Improved BIER-TE BIFT on BFR H

<table>
<thead>
<tr>
<th>Adjacency BP (SI:BitString)</th>
<th>Action</th>
<th>BFR-NBR (Next Hop)</th>
</tr>
</thead>
<tbody>
<tr>
<td>13'(7:00010000)</td>
<td>fw-connected</td>
<td>C</td>
</tr>
<tr>
<td>15'(7:01000000)</td>
<td>fw-connected</td>
<td>G</td>
</tr>
<tr>
<td>19'(8:00000100)</td>
<td>fw-connected</td>
<td>D</td>
</tr>
</tbody>
</table>

**Secondary BIFT for Px on BFR H**

<table>
<thead>
<tr>
<th>Adjacency BP (SI:BitString)</th>
<th>Action</th>
<th>BFR-NBR (Next Hop)</th>
</tr>
</thead>
<tbody>
<tr>
<td>4 (0:00001000)</td>
<td>local-decap</td>
<td></td>
</tr>
<tr>
<td>18'(8:00000010)</td>
<td>lan-connected</td>
<td>Px</td>
</tr>
</tbody>
</table>

Figure 13: Improved BIER-TE BIFT on BFR H

The improved BIER-TE BIFT on BFR D is illustrated in Figure 14. It comprises the forwarding entry for the LAN adjacency from D to Px (indicated by BP 20’) 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 13’), Px to G (indicated by 15’) and Px to H (indicated by 17’) respectively.
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 15. 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 15: 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 10 going through the LAN in the figure.

The new BP assignments for the LAN in Figure 1 is shown in Figure 10. 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 Px (indicated by BP 16’), the link/adjacency from C to Px (indicated by BP 10’), the link/adjacency from D to Px (indicated by BP 12’), the link/adjacency from E to Px (indicated by BP 14’), the link/adjacency from F to Px (indicated by BP 16’), and the link/adjacency from G to Px (indicated by BP 16’).
from C to F (indicated by BP 12’), and the link/adjacency from Px to H (indicated by BP 17’). This path is represented by {2’, 4’, 6’, 12’, 16’, 17’, 2, 4}. 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’, 16’, 17’, 2, 4}. The packet received by C has path {12’, 16’, 17’, 2, 4}.

After receiving the packet from B, G "sends" the packet to Px according to the forwarding entry for BP 16’ in G’s improved BIER-TE BIFT. After receiving the packet from G, which has path {12’, 16’, 17’, 2, 4}, Px "sends" the packet to H according to the forwarding entry for BP 17’ 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’, 2, 4}, 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’, 16’, 17’, 2, 4}, 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 {16’, 17’, 2, 4}, 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]


8.2. Informative References

[I-D.eckert-bier-te-frr]  

[I-D.ietf-rtgwg-segment-routing-ti-lfa]  
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
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 BFER/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](image)

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 CGM2 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.

```
+----------+-----+---------------+---------+
| TotalLen | Rsv | RecursiveUnit | Padding |
+----------+-----+---------------+---------+
.... TotalLen .......
```

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, Padding (bits) = (8 - TotalLen % 8).

3.3.1. RecursiveUnit

The RecursiveUnit field is structured as shown in Figure 4.

```
+-----------+------------------+-
| BitString...| AddressingField...| RecursiveUnit 1...M|
+-----------+------------------+-
```

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...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.

```
+---------------------------------+  +---------------------------------+
|       L1     |   L2     |...|   L(M-1)   |
+---------------------------------+  +---------------------------------+
```

Figure 5: RBS AddressingField

The AddressingField consists of one or more fields Li, i=1...(M-1). Li 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.
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.
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.

![Figure 9: BIER-RBS BIFT on B](image)

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](image)

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.

.............. RecursiveUnit ..............
  +------------------------------------------------+
  | Tlen:32 R:011 L1:18 S:011 L1:3 C:001 D:0001 E:001 |
  +------------------------------------------------+
  8bit 3bit 8bit 3bit 8bit 3bit 4bit 3bit
  . RecursiveUnit 1...... .....  
  RecursiveUnit 2 ...

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.

.............. RecursiveUnit ..............
  +------------------------------------------------+
  | Tlen:32 R:011 L1:18 S:01100000110010001 E:001 |
  +------------------------------------------------+
  8bit 3bit 8bit 18bit 3bit
  . RecursiveUnit 1...... .....  
  RecursiveUnit 2 ...

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.
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 13: RBS BIFT on BFR R

Figure 14: RBS Address processed by BFR-S

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|
+-------+-----+-----+
```

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.
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.

Figure 18: RBS Address processed by BFR-D

<table>
<thead>
<tr>
<th>Tlen:4</th>
<th>D:0001</th>
<th>Pad:4</th>
</tr>
</thead>
<tbody>
<tr>
<td>8bit</td>
<td>4bit</td>
<td>4bit</td>
</tr>
</tbody>
</table>

Figure 19: RBS BIFT on BFR-D

<table>
<thead>
<tr>
<th>BP</th>
<th>Recursive</th>
<th>Adjacency</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>S</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>C</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>E</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>local_decap</td>
</tr>
</tbody>
</table>

Figure 20: RBS Address processed by BFR-E

<table>
<thead>
<tr>
<th>Tlen:3</th>
<th>E:001</th>
<th>Pad:5</th>
</tr>
</thead>
<tbody>
<tr>
<td>8bit</td>
<td>3bit</td>
<td>5bit</td>
</tr>
</tbody>
</table>

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);
        } 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.

```
B1
 /  \
B2  B3
 /  /  \
/  /  \
B4  B5  B6
/..|  /   |..\nB7..B99 B100..B200 B201...B300
```

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=>B2=>B4=>(B7..B99), for a first packet, an RBS address B1=>B3=>B5=>(B100..B200) for a second packet and a third RBS address B1=>B3=>B6=>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=>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 today's programmable forwarding planes such as P4.

Whereas BIER-TE clears bit on each hop's 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 document's 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]


10.2. Informative References


Author’s Address

Toerless Eckert
Futurewei Technologies USA
2220 Central Expressway
Santa Clara, CA 95050
United States of America
M-LDP Signaling Through BIER Core
draft-ietf-bier-mldp-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.
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

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 form IBBR. It should assign a unique "upstream assigned label" for each arriving FEC generated by IBBRs.

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

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

- The label can be assigned from a domain-wide Common Block (DCB) [draft-ietf-bess-mvpn-evpn-aggregation-label]

- The Interface ID TLV, as per [RFC6389] should includes a new BIER sub-domain sub- tlv (type TBD)
The EBBR will also generate a new label and FEC toward the ROOT on the LDP domain. The EBBR should stitch this generate label with the "upstream assigned label" to complete the P2MP or MP2MP LSP.

With the 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 downstream 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]

[draft-ietf-bier-pim-signaling]
"H. Bidgoli, F. Xu, J. Kotalwar, IJ. Wijnands, M. Mishra, Z. Zhang "PIM Signaling Through BIER Core"", February 2012.


[RFC6512] "IJ. Wijnands, E. Rosen, M. Napierala, N. Leymann "Using Multipoint LDP when the backbone has No route to the root"", February 2012.


Authors’ Addresses
Hooman Bidgoli (editor)
Nokia
Ottawa
Canada
Email: hooman.bidgoli@nokia.com

Jayant Kotalwar
Nokia
Montain 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
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:

- **BFR-id**: BFR Identifier. It is a number in the range [1,65535]
- **BGP**: Border Gateway Protocol
- **BIER**: Bit Index Explicit Replication
- **BIFT**: Bit Index Forwarding Table
- **FI**: Forwarding indication
- **IGMP**: Internet Group Management Protocol
- **IGP**: Interior Gateway Protocols
- **MLD**: Multicast Listener Discover
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.
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 PEC message with the newly defined object, but without the B bit set in STATEFUL-PCE-CAPABILITY TLV in the OPEN object, it MUST:

- Send a PCErr message with Error-Type=10 (Reception of an invalid object) and Error-Value TBD2 (BIER-MULTICAST-CAPABILITY bit is not set).
- Terminate the PCEP session.

6. PCEP message

6.1. PCRpt message

MSR object: Section 7.1 should be included in the PCRpt message when PCC registers multicast source information with PCE.

MRI object: Section 7.2 should be included in the PCRpt message when PCC sends multicast join messages to PCE.

MRS object: Section 7.4 should be included in the PCRpt message when PCC informs 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:

```
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                           flags                         |B|R|A|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|        Auxliary Length        |            Reserved           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
˜                        Auxiliary Data                         ˜
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
˜                        Optional TLVs                          ˜
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

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:

<table>
<thead>
<tr>
<th>Type = TBD4</th>
<th>Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prefix Length</td>
<td>Reserved</td>
</tr>
<tr>
<td>Multicast Source Address</td>
<td></td>
</tr>
</tbody>
</table>

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:

```
0                   1                   2                   3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          Type = TBD5          |             Length            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Subdomain-id  |            BFR-ID             |  BSL  |  Res  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

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:

```
```
Internet-Draft PCE based BIER Procedures and Protocol Extensions

Figure 5: VPN Information TLV Format

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 multicast joining or leaving.

MRI Object-Class is TBD7. MRI Object-Type is 1.

The format of the MRI object body is:

Figure 6: MRI 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.

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 PCE.

MRI object could include four types of TLVs, namely Multicast Source Address TLVSection 7.1.1, BIER INFO TLVSection 7.1.2, VPN Information TLVSection 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:

```
0                   1                   2                   3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          Type = TBD8          |             Length            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         Prefix Length         |            Reserved           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
˜                    Multicast Group Address                    ˜
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

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:
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:

```
0                   1                   2                   3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          Number Length        |            Reserved           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      Number of Receivers                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      Optional TLVs                          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

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 TLVSection 7.1.1 and Multicast Group Address TLVSection 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:

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>TBD1</td>
<td>BIER-MULTICAST-CAPABILITY</td>
<td>This document</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>Error-Type</th>
<th>Description</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>Error-value = TBD2</td>
<td>This document</td>
</tr>
<tr>
<td></td>
<td>B bit is not set</td>
<td></td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>Object-Class Value</th>
<th>Description</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>TBD3</td>
<td>Multicast Receiver Information</td>
<td>This document</td>
</tr>
<tr>
<td>TBD7</td>
<td>Multicast Receiver Information</td>
<td>This document</td>
</tr>
<tr>
<td>TBD9</td>
<td>Forwarding Indication</td>
<td>This document</td>
</tr>
<tr>
<td>TBD10</td>
<td>Multicast Receiver Status</td>
<td>This document</td>
</tr>
</tbody>
</table>

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:
<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>TBD4</td>
<td>Multicast Source Address</td>
<td>This document</td>
</tr>
<tr>
<td>TBD5</td>
<td>Multicast Group Address</td>
<td>This document</td>
</tr>
<tr>
<td>TBD6</td>
<td>BIER Information TLV</td>
<td>This document</td>
</tr>
<tr>
<td>TBD8</td>
<td>VPN Information</td>
<td>This document</td>
</tr>
</tbody>
</table>

12. Contributor

13. Acknowledgement

14. Normative References


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
Routing Header Based BIER Information Encapsulation
draft-wang-bier-rh-bier-02

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 IPv6 Routing Header type 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 April 28, 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
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 technology, 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: BIERn6([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 a new option 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. 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.

```
+---------------------------------+-
<table>
<thead>
<tr>
<th>Next Header</th>
<th>Hdr Ext Len</th>
<th>Routing Type</th>
<th>Segment Left</th>
</tr>
</thead>
</table>
+---------------------------------+-
| BIFT-id | Ver | TTL |
+---------------------------------+-
| BSL | Entropy | DSCP | OAM |
+---------------------------------+-
| BFIR-id | RsV | Reserved |
+---------------------------------+-
| . . | BitString . |
| +---------------------------------+-
```

Figure 1: The format of BIER Routing Header

Where:
- Next Header (8 bits): indicating the message header type immediately after the routing header.

- HDR Ext Len (8 bits): indicating the length of the routing header.

- Routing Type (8 bits): TBD. Identifying the newly defined Routing Header to encode BIER information.

- Segments Left (8 bits): indicating the number of explicitly listed intermediate nodes to be accessed before reaching the final destination. It is not used here for the time being, and all are set to 0.

- BIFT-id (20 bits): each < SD, Si, BSL > is assigned a BIFT-id.

- Ver (4 bits): identifying the version of the BIER header. When an unsupported BIER header version is received, the BFR needs to discard the packet and record the error.

- TTL (8 bits): indicating the lifetime of the message. It is used to prevent ring. The processing process is the same as that in non MPLS networks.

- BSL (4 bits): indicating the length of BitString.

- Entropy (20 bits): this field specifies an "entropy" for ECMP.

- DSCP (6 bits): this field is used to support different service codes.

- OAM (2 bits): by default, this value will be set to 0 by BFIR, and other BFRs will not be modified. Whether to use this field is optional.

- BFIR-id (16 bits): indicating BFR ID of BFIR.

- Rsv (2 bits): unused, set to 0.

- Reserved (14 bits): reserved field, set to 0.

- BitString (variable): the length must be reflected in the BSL field. The string saved in this field is used to identify the destination BFER of the packet.
4. Multicast Packet Forwarding Procedures

Based on the newly defined BIER Routing Header, the devices support BIER Routing Header will perform the following steps to forward the multicast packets:

1) When BFIR receives the IPv6 multicast packets from the multicast source, it will add BIER Routing Header to indicate the BFERs that want to receive such multicast packet. The encapsulated multicast packet will be forwarded according to the BIFT that identified by the BIFT-id.

2) Each BFR (includes BFIR) will check whether the direct-connected device support BIER Routing Header. If yes, proceed to step 3); otherwise, proceed to step 2.1).

2.1) Calculating 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 device, the device will perform normal IPv6 forwarding according to the outer IPv6 Header.

3) On the router that supports the BIER Routing Header, perform the normal BIER forwarding process as described in [RFC8279].

The detail procedures for forwarding the multicast packets based on the newly defined Routing Header are described in the following sections.

4.1. All devices in BIER domain support BIER Routing Header
Figure 2: All devices in BIER domain support BIER Routing Header

The topology is shown in Figure 2, device 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. Each device will perform the following steps after receiving the packet:

1). Checking whether there is BIFT corresponding to the BIFT-id locally. If yes, proceed to step 2); otherwise, discard the packet.
2). Checking whether the direct-connected device support BIER Routing Header. If yes, forwarding the packet according to the BIFT related to the BIFT-id; otherwise, see section 4.2 for detailed procedures.

During the forwarding procedures, the source address and destination address of the IPv6 multicast packet are not changed, only the BitString in BIER Routing Header is updated.

4.2. Some devices in BIER domain do not support BIER Routing Header

---

Packet 1

<table>
<thead>
<tr>
<th>IPv6 Header</th>
<th>IPv6 Address of Multicast Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>BIER Header</td>
<td>IPv6 Multicast Destination Address</td>
</tr>
<tr>
<td>BIER Routing Header</td>
<td>BitString = 00101100</td>
</tr>
</tbody>
</table>

---

Packet 2

| Outer IPv6 Header | Source IP Address = D |
| Inner IPv6 Header | Destination IP Address = E |
| IPv6 Header | IPv6 Address of Multicast Source |
The topology is shown in Figure 3, all devices expect device 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. When it is found that device C does not support BIER Routing Header, device D will perform the following steps after receiving the packet:

1. Calculating the IPv6 address of next hop device (Node E) that supports BIER Routing Header.

2. Encapsulating an outer IPv6 Header to the packet. The calculated IPv6 address (E) is used as the destination address of the outer IPv6 Header, and its own IPv6 address (D) is used as the source address of the outer IPv6 Header. BitString will not be changed.

3. Sending the packet to directed-connected device C.

After receiving the packet, device C will perform IPv6 forwarding according the information in outer IPv6 Header, and send the packet to device E. Device E will send it to device F according the information in BIER Routing Header. In the forwarding process, the source address and destination address in the Inner IPv6 Header are not changed.

5. Security Considerations

TBD

6. IANA Considerations

This document defines a new IPv6 Routing Header - BIER Routing Header. The code point is from the "Internet Protocol Version 6
It is recommended to set the code point of BIER Routing Header to 7.

7. References

7.1. Normative References


7.2. Informative References


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
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.
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 BFR-ID is needed for each <sub-domain, BFR/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]

[I-D.zzhang-intarea-generic-delivery-functions]

[RFC8279]

[RFC8401]

7.2. Informative References

[I-D.ietf-teas-ietf-network-slices]

[I-D.li-apn-framework]
Authors' Addresses

Zhaohui Zhang
Juniper Networks
Email: zzhang@juniper.net

Antoni Przygienda
Juniper Networks
Email: prz@juniper.net