Multicast VPN Using MPLS P2MP and BIER
draft-xie-bier-mvpn-mpls-p2mp-01

Abstract

The MVPN specifications allow the use of several different kinds of P-tunnel technology, such as mLDP P2MP, RSVP-TE P2MP and PIM SSM. It is common for such a P-tunnel having a multicast-specific path. Bit Index Explicit Replication (BIER) is an architecture that provides optimal multicast forwarding without requiring intermediate routers to maintain any per-flow state by using a multicast-specific BIER header.

[I-D.ietf-bier-mvpn] delivers a solution of MVPN using SPF based BIER defined in [RFC8279]. It can not, however, support a multicast-specific path well, something common in legacy MVPN deployment. [RFC8279] provides a solution to support mid nodes without BIER-capability. It can not, however, support deployment on a network that has edge nodes without BIER-capability, which is common in some SP-networks.

This document introduces a seamless transition mechanism from legacy MVPN to MVPN using P2MP based BIER while preserving existing features such as multicast-specific PATH and Live-Live protection. It also introduces a seamless Live-Live protection mechanism by re-using the Entropy field of the BIER header, and two methods to deploy BIER when edge nodes and/or mid nodes don’t have BIER-capability.

Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].
1. Introduction ................................................. 3
2. Terminology .................................................. 4
3. Applicability Statement ................................. 4
4. MVPN using P2MP based BIER .......................... 5
   4.1. Overview .............................................. 5
   4.2. MVPN Transition from P2MP to P2MP based BIER ....... 6
       4.2.1. Use of the PTA in x-PMSI A-D Routes .......... 7
       4.2.2. Use of the PTA in Leaf A-D routes ............ 9
   4.3. Building P2MP based BIER forwarding state .......... 10
   4.4. Inheriting and Developing of Live-Live protection .... 11
5. P2MP based BIER Forwarding Procedures .............. 11
   5.1. Overview .............................................. 12
   5.2. P2MP based BIER forwarding customization ........... 13
   5.3. When Mid, Leaf or Bud nodes do not support P-CAPABILITY ... 15

Xie, et al.  Expires September 6, 2018
1. Introduction

[RFC6513] and [RFC6514] specify the protocols and procedures that a Service Provider (SP) can use to provide Multicast Virtual Private Network (MVPN) service to its customers. Multicast tunnels are created through an SP’s backbone network; these are known as "P-tunnels". The P-tunnels are used for carrying multicast traffic across the backbone. The MVPN specifications allow the use of several different kinds of P-tunnel technology, such as mLDP P2MP, RSVP-TE P2MP and PIM SSM. It is common for such a P-tunnel having a multicast-specific path.

Bit Index Explicit Replication (BIER) [RFC8279] is an architecture that provides optimal multicast forwarding through a "multicast domain", without requiring intermediate routers to maintain any per-flow state, by using a multicast-specific BIER header (per [RFC8296]).

[I-D.ietf-bier-mvpn] delivers a solution of MVPN using SPF based BIER defined in [RFC8279]. It can not, however, support a multicast-specific path well, something common in legacy MVPN deployment.

[RFC8279] provides a solution to support mid nodes without BIER-capability. It cannot, however, support deployment on a network that has edge nodes without BIER-capability, which may be common in some SP-networks, especially when most of the nodes in a network or part of a network are edge or service nodes.

This document introduces a seamless transition mechanism from legacy MVPN to MVPN using P2MP based BIER, by applying a BIER encapsulation in data-plane to eliminate per-flow states, while preserving existing features such as multicast-specific PATH and Live-Live protection by using existing protocols.

It also introduces a seamless Live-Live protection developped from existing Live-Live protection scheme, by re-using the Entropy field of the BIER header, for the ECMP/Entropy feature is not supported in P2MP (per RFC6790).
It also introduces a seamless deployment solution on networks with Non-BIER-capability Edge nodes and/or Mid nodes, by exploring the P2MP/tree based BIER forwarding procedure in detail. Such a P2MP/tree based BIER is mentioned but not explored in detail in RFC8279.

2. Terminology

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

- **LSP**: Label Switch Path
- **LSR**: Label Switching Router
- **P2MP**: Point to Multi-point
- **P-tunnel**: A multicast tunnel through the network of one or more SPs. P-tunnels are used to transport MVPN multicast data.
- **PMSI**: Provider Multicast Service Interface
- **x-PMSI A-D route**: a route that is either an I-PMSI A-D route or an S-PMSI A-D route.
- **PTA**: PMSI Tunnel attribute. A type of BGP attribute known as the PMSI Tunnel attribute.
- **P2MP based BIER**: BIER using P2MP LSP as topology
- **P-CAPABILITY**: A capability to Process BitString in BIER Header of a packet.
- **D-CAPABILITY**: A capability to Disposit BIER Header of a packet, including or excluding the BIER Label.
- **BSL**: Bit String Length, that is 64, 128, 256, etc (per RFC8279).

3. Applicability Statement

The BIER architecture document [RFC8279] describes how each node forwards BIER packets hop by hop to neighboring nodes without generating duplicate packets. This forwarding is for the case where a form of underlay called "many to many" and built by IGP is used. Obviously, the case of underlay of "one to many" or P2MP is a simpler scenario, and the forwarding procedure naturally applies. However, as is well-known, such a forwarding procedure requires the support of
hardware. The usage of the same forwarding method for both complex scenarios and simple scenarios will inevitably require complex hardware forwarding.

This document describes how BIER forwarding can be customized and simplified with an underlay of "one to many" or P2MP (see chapter 5). This customization and simplification eliminates some of the unnecessary data plane processing and so is easier to implement with existing hardware. Based on this customization of the forwarding method for P2MP-based BIER, a variety of Partial Deployment methods are given for the different capabilities of the hardware to support BIER forwarding. Compared with RFC8279, when there is no BIER forwarding capability on edge nodes, Partial Deployment can be carried out; For the case where the intermediate node has no BIER forwarding capability, P2MP forwarding can be used without the need for unicast replication.

This document also describes a MVPN Transition solution that eliminates the per-flow state by introducing BIER MPLS encapsulation and forwarding in data-plane, while preserving the original control-plane protocol and its features, especially when some sort of path customizing being used. The said path customization include RSVP-TE P2MP using an explicit path, PIM-SSM tree using an explicit path, and MLDP P2MP where static route was used. These features can continue to retain, making the transition process seamless.

This document also describes a seamless redundancy mechanism for the widely deployed MVPN Live-Live protection, by using the added information in the BIER header as a sequence-number of per packet. This will bring additional benefit to the transition process from traditional MVPN using PIM-SSM/mLDP/RSVP-TE to MVPN using P2MP based BIER.

4. MVPN using P2MP based BIER

4.1. Overview

According to [RFC8279], the P2MP based BIER is a BIER which using a form of tree as the underlay. The P2MP LSP is not only a LSP, but also a topology as the BIER underlay. The P2MP based BIER is P-tunnel, which is used for bearing multicast flows. Every flow can think as binding to an independent tunnel, which is constructed by the BitString in the BIER header of every packet of the flow. Multicast flows are transported in SPMSI-only mode, on P2MP based BIER tunnels, and never directly on P2MP LSP tunnel.

Section 4.2 describes the overall principle of transitioning a Legacy MVPN using P2MP to a MVPN using BIER. We call it a tick-tock
transitioning. It also describes the detail use of new types of PTA in BGP MVPN routes to indicate PEs to initialize the building of P2MP based BIER forwarding.

Section 4.3 describes the Underlay protocols to build P2MP based BIER forwarding briefly.

Section 4.4 describes how the widely deployed Live-Live protection in MVPN can be inherited and developed. This will introduce a new function to BIER Layer by re-using the Entropy field of the BIER encapsulation.

4.2. MVPN Transition from P2MP to P2MP based BIER

This section describes a MVPN transitioning solution that eliminates the per-flow state by introducing BIER MPLS encapsulation and forwarding procedure in data-plane, while preserving the originally deployed control-plane protocol and its features, especially when some sort of path customizing being used.

When transitioning a MVPN using mLDP P2MP P-tunnel, then continue using mLDP to build a P2MP based BIER forwarding, preserving the original mLDP features. For example, mLDP uses static route to specify a path other than the path of IGP.

When transitioning a MVPN using RSVP-TE P2MP P-tunnel, then continue using RSVP-TE to build a P2MP based BIER forwarding, preserving the original RSVP-TE features. For example, RSVP-TE use explicit path to specify a path other than the path of IGP.

When transitioning a MVPN using PIM-SSM p-tunnel, then continue using PIM-SSM to build a P2MP based BIER forwarding, preserving the original PIM features. For example, PIM use explicit vector to specify a path other than the path of IGP.

It is called a tick-tock transitioning, that is to introduce a new standard BIER encapsulation defined in [RFC8296], while preserving old control-plane protocols, features, and most of the devices. When all or most of the devices in a network support BIER forwarding, then we can do a further tick-tock transitioning, that is to introducing a new control-plane protocol while preserving old data-plane encapsulation, when the control-plane protocol can provide all the needed features of the old ones.
4.2.1. Use of the PTA in x-PMSI A-D Routes

As defined in [RFC6514], the PMSI Tunnel attribute (PTA) carried by an x-PMSI A-D route identifies the P-tunnel that is used to instantiate a particular PMSI. If a PMSI is to be instantiated by P2MP LSP based BIER, the PTA is constructed by a BFIR, which is also an Ingress LSR. This document defines the following Tunnel Types:

+ TBD - RSVP-TE built P2MP BIER
+ TBD - mLDP built P2MP BIER
+ TBD - PIM-SSM built P2MP BIER

Allocation is expected from IANA for two new tunnel type codepoints from the "P-Multicast Service Interface Tunnel (PMSI Tunnel) Tunnel Types" registry. These codepoints will be used to indicate that the PMSIs is instantiated by MLDP or RSVP-TE or PIM extension with support of BIER.

When the Tunnel Type is set to RSVP-TE built P2MP BIER, the Tunnel Identifier include two parts, as follows:

```
+---------------------------------------------------------------+
| BS Len | Max SI | Must Be Zero | P2MP ID |
|---------------------------------------------------------------|
| MUST be zero | Tunnel Range Base | Extended Tunnel ID |
+---------------------------------------------------------------+
```

Figure 1: PTA of RSVP-TE built P2MP BIER

BS Len: A 4 bits field. The values allowed in this field are specified in section 2 of [RFC8296].

Max SI: A 1 octet field. Maximum Set Identifier (section 1 of [RFC8279]) used in the encapsulation for this BIER sub-domain.

<Extended Tunnel ID, Reserved, Tunnel Range Base, P2MP ID>: A ID as carried in the RSVP-TE P2MP LSP SESSION Object defined in [RFC4875].

The "Tunnel Range" is the set of P2MP LSPs beginning with the Tunnel Range base and ending with ((Tunnel Range base)+(Tunnel Number)- 1).
A unique Tunnel Range is allocated for the BSL and a Sub-domain-ID implicited by the P2MP.

The size of the Tunnel Range is determined by the number of Set Identifiers (SI) (section 1 of [RFC8279]) that are used in the topology of the P2MP-LSP. Each SI maps to a single Tunnel in the Tunnel Range. The first Tunnel is for SI=0, the second Tunnel is for SI=1, etc.

When the Tunnel Type is set to mLDP built P2MP BIER, the Tunnel Identifier include two parts, as follows:

```
<table>
<thead>
<tr>
<th>BS Len</th>
<th>Max SI</th>
<th>Must Be Zero</th>
</tr>
</thead>
<tbody>
<tr>
<td>P2MP Type(0x06)</td>
<td>Address Family</td>
<td>Address Length</td>
</tr>
<tr>
<td>~</td>
<td>Root Node Address</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Opaque Length(0x0007)</td>
<td>OV Type(0x01)</td>
</tr>
<tr>
<td></td>
<td>(Low 8b)(0x04)</td>
<td>Tunnel Range Base(High 24b)</td>
</tr>
<tr>
<td></td>
<td>(Low 8b)</td>
<td></td>
</tr>
</tbody>
</table>
```

Figure 2: PTA of MLDP built P2MP BIER

BS Len: A 4 bits field. The values allowed in this field are specified in section 2 of [RFC8296].

Max SI: A 1 octet field. Maximum Set Identifier (section 1 of [RFC8279]) used in the encapsulation for this BIER sub-domain.

<Type=0x06, AF, AL, RootNodeAddr, Opaque Length=0x0007, OV Type=0x01, OV Len=0x04, Tunnel Range Base>: A P2MP Forwarding Equivalence Class (FEC) Element, with a Generic LSP Identifier TLV as the opaque value element, defined in [RFC6388].

The "Tunnel Range" is the set of P2MP LSPs beginning with the Tunnel Range base and ending with ((Tunnel Range base)+(Tunnel Number)-1). A unique Tunnel Range is allocated for the BSL and a Sub-domain-ID implicited by the P2MP.

The size of the Tunnel Range is determined by the number of Set Identifiers (SI) (section 1 of [RFC8279]) that are used in the
topology of the P2MP-LSP. Each SI maps to a single Tunnel in the Tunnel Range. The first Tunnel is for SI=0, the second Tunnel is for SI=1, etc.

When the Tunnel Type is set to PIM-SSM built P2MP BIER, the Tunnel Identifier include two parts, as follows:

```
+-----------------+-----------------+-----------------+-----------------+
| BS Len | Max SI | Must Be Zero | BS Len | Max SI | Must Be Zero |
+-----------------+-----------------+-----------------+-----------------+
| P-Root Node Address | ~ | P-Multicast Group Base | ~ |
+-----------------+-----------------+-----------------+-----------------+
```

Figure 3: PTA of PIM-SSM built P2MP BIER

BS Len: A 4 bits field. The values allowed in this field are specified in section 2 of [RFC8296].

Max SI: A 1 octet field. Maximum Set Identifier (section 1 of [RFC8279]) used in the encapsulation for this BIER sub-domain.

<P-Root Node Address, P-Multicast Group Base>: A ID to build a PIM-SSM tree when build the P2MP BIER information for a specified SI. Use a consecutive number of Groups from P-Multicast Group Base, multiple PIM-SSM trees will be built for multiple SIs, and multiple P2MP BIER information will be built accordingly.

When the Tunnel Type is any of the above, The "MPLS label" field OPTIONAL contain an upstream-assigned non-zero MPLS label. It is assigned by the router (a BFIR) that constructs the PTA. Absence of an MPLS Label is indicated by setting the MPLS Label field to zero.

When the Tunnel Type is any of the above, two of the flags, LIR and LIR-pF, in the PTA "Flags" field are meaningful. Details about the use of these flags can be found in [RFC6513], [I-D.ietf-bess-mvpn-expl-track] and [I-D.ietf-bier-mvpn]].

4.2.2. Use of the PTA in Leaf A-D routes

Before an egress PE can receive a (C-S,C-G) flow from a given ingress PE via RSVP-TE/MLDP P2MP LSP based BIER, the egress PE must have received one of the following x-PMSI A-D routes from the ingress PE:

A "less specific" x-PMSI A-D route, (C-*,C-*) or (C-S,C-*) S-PMSI A-D route.

In which, the PTA tunnel Type is "RSVP-TE P2MP LSP based BIER" or "MLDP P2MP LSP based BIER".

The rules for determining which x-PMSI A-D route is the match for reception are given in [RFC6625]. If such a route is found, we refer to it as the "matching x-PMSI A-D route." If no matching x-PMSI A-D route for (C-S,C-G) is found, the egress PE cannot receive the (C-S,C-G) flow from the ingress PE via RSVP-TE/MLDP P2MP LSP based BIER until such time as a matching route is received.

When an egress PE determines that it needs to receive a (C-S,C-G) flow from a particular ingress PE via RSVP-TE/MLDP P2MP LSP based BIER, it originates a Leaf A-D route. Construction of the Leaf A-D route generally follows the procedures specified in [RFC6514], or optionally, the procedures specified in [I-D.ietf-bess-mvpn-expl-track]. However, when RSVP-TE/MLDP P2MP LSP based BIER is being used, the Leaf A-D route MUST carry a PTA that is constructed as follows:

1. The tunnel type MUST be set to RSVP-TE/MLDP P2MP LSP based BIER, corresponding to the PTA of the matching x-PMSI A-D route.
2. The MPLS label field SHOULD be set to zero.
3. The BFR-Prefix field of the Tunnel Identifier field MUST be set to the egress PE's IP-Address. This IP-Address is the same as the Originating Router’s IP Addr field of the NLRI of the Leaf A-D route.

When an ingress PE receives such a Leaf A-D route, it learns the BFR-Prefix of the egress PE from the PTA. The ingress PE does not make any use the value of the PTA’s MPLS label field.

Failure to properly construct the PTA cannot always be detected by the protocol, and will cause improper delivery of the data packets.

4.3. Building P2MP based BIER forwarding state

When P2MP based BIER are used, then it is not necessary to use IGP or BGP to build the BIER routing table and forwarding table. Instead, the BIER layer information is carried by MLDP or RSVP-TE or PIM, when they building the P2MP tree or PIM tree.

The detail procedure for building P2MP based BIER forwarding state using mLDP, RSVP-TE or PIM is outside the scope of this document.
4.4. Inheriting and Developing of Live-Live protection

Multicast has its special service protection requirement, especially when multicast service is compressed or uncompressed video. Accordingly, there are some multicast-specific methods of protection, such as Live-Live. [RFC7431] defines a method of detecting failure locally by comparing the packets received from live-live paths, but it depends on an APP level encapsulation, which may not always be satisfied in any deployment. Furthermore, it is inconsequential and inefficient to do such a deep detecting in a data-plane forwarding procedure. [I-D.ietf-bess-mvpn-fast-failover] also defines a Live-Live method for protecting Multicast in MVPN, in which a method of determining the status of a tunnel using "(S,G) counter information" is defined, but it does not describe how to get such a (S,G) counter.

This document specifies one OPTIONAL extension to enhance Live-Live protection, re-using the Entropy field of BIER header as a Sequence number of multicast packet, on the condition that the field is not used for ECMP, such as in the P2MP LSP topology. Currently ECMP is not used for P2MP LSP, as per [RFC6790].

This is an optional function of BIER Layer. If this function is enabled, every BFR of the domain is required to support, which means:

1. The BFIR (and Ingress LSR) will push a sequence-number in the Entropy field, per-flow per-packet.
2. The middle BFR will ignore the Entropy field, and not do the selection of multi-tables.
3. The BFER (and Egress LSR) will do packet check from live-live paths, and do forward packet with zero packet loss, on a per-flow basis.

5. P2MP based BIER Forwarding Procedures

The MVPN application plays the role of the "multicast flow overlay" as described in [RFC8279].

This section specifies some OPTIONAL rules for forwarding a BIER-encapsulated data packet within a P2MP topology underlay. It is part of the "BIER Layer" function, which is mentioned as an alternative deployment of BIER on some sort of multicast-specific tree, but not detailedly explored in [RFC8279].

These OPTIONAL rules are some sort of customization and simplification to the common BIER forwarding procedures, and will
produce the same results as the procedures in [RFC8279], on condition that the underlay topology is a P2MP.

These OPTIONAL rules will lead to some new methods to deploy when some nodes do not support BIER forwarding. These new methods and its effects will be introduced as well.

5.1. Overview

As [RFC8279] describes:

1. BIER support using the default topology of the unicast IGP as the routing underlay. To quote from [RFC8279]: "By default, each sub-domain uses the default topology of the unicast IGP as the routing underlay."

2. BIER also support using other topologies as the routing underlay, including a tree topology. To quote from [RFC8279]: "Alternatively, one could deploy a routing underlay that creates a multicast-specific tree of some sort. Then BIER could be used to forward multicast data packets along the multicast-specific tree, while unicast packets follow the ‘ordinary’ OSPF best path."

This document specifies one OPTIONAL Forwarding Procedure of BIER encapsulation packet, on the condition that the BIER underlay topology is P2MP LSP, as describes in the above sections. It is in fact a customized forwarding procedure, and a detail exploration of BIER forwarding along a multicast-specific tree. Comparing to the common Forwarding Procedure described in [RFC8279], there is some considerable simplification:

1. Not need to Edit the BitString when forwarding packet to Neighbor, for the underlay P2MP topology is already loop-free and duplicate-free. This can further lead to a method to by-pass the BIER encapsulation packet when a node does not support the BitString process.

2. Not need to do a disposition function by parsing the BitString, for a P2MP can identify a disposition function by a node’s Label when the P2MP is built. This can further reduce the complex BitString processing for legacy hardware on edge, and lead to a method to deploy on exist network when an edge node does not support BitString process.

3. Not need to use Entropy in the BIER Header, for current P2MP topology is ECMP-excluding as per [RFC6790]. This can make it possible to re-use the field for other function, and lead to a
method of inheriting and developing of the Live-Live protection, as described in chapter 4.

The main principle of the optional forwarding procedure of the P2MP-based BIER is, on the basis of P2MP forwarding procedure according to the BIER-MPLS label, to use the BitString to prune the undesired P2MP downstream. This is an enhancement to the standard P2MP forwarding.

The enhancement to the P2MP forwarding is to add a Forwarding BitMask to existing NHLFE defined in [RFC3031], for checking with the BitString in a packet, to determine whether the packet is to be forwarded or pruned. If the checking result by AND’ing a packet’s BitString with the F-BM of the NHLFE (i.e., Packet->BitString & F-BM) is non-zero, then forward the packet to the next-hop indicated by the NHLFE entry, and the Label is switched to the proper one in the NHLFE. If the result is zero, then do not forward the packet to the next-hop indicated by the NHLFE entry.

5.2. P2MP based BIER forwarding customization

For a P2MP tree, every node has a role of Root, Branch, Leaf, or Bud, as specified in [RFC4611].

EXAMPLE 1: Take the following figure as an example.

```
  ( A ) ------------ ( B ) ------------ ( C ) ------------ ( D )
    (Root)                \                  \            1 (0:0001)
                  \                  \          2 (0:0010)
         ( E )            ( F )            3 (0:0100)
```

Figure 4: P2MP-based BIER Topology without BUD nodes

Forwarding Table on A:

```
NHLFE(TreeID, OutInterface<toB>, OutLabel<alloc by B>, F-BM<0111>)
```

Forwarding Table on C:

```
ILM (inLabel<alloc by C>, action<Replication to TreeID>,
     Flag=Branch|CheckBS, BSL)
NHLFE(TreeID, OutInterface<toD>, OutLabel<alloc by D>, F-BM<0001>)
NHLFE(TreeID, OutInterface<toF>, OutLabel<alloc by F>, F-BM<0010>)
```

For Node C, the ability to receive a MPLS-encapsulation BIER packet, match ILM and get a TreeID, replicate to NHLFE Entries of the TreeID
according to the result of AND’ing the BitString of packet and the F-BM of a NHLFE Entry, is called a P-CAPABILITY, which means to Process BitString in each packet.

Forwarding Table on B is the same to C.

Forwarding Table on D:

ILM (inLabel<alloc by D>, action<Replication to TreeID>, Flag=Leaf|CheckBS, BSL)

LEAF(TreeID, F-BM<0001>, flag=PopBIERincluding)

When Node D receive a MPLS-encapsulation BIER packet, it get the Label and match ILM, then do a replication according to the LEAF and check whether to proceed by AND’ing the BitString in the replicated packet and the F-BM in the LEAF entry. When the AND’ing result is non-zero then do a POP to the packet to disposit the whole BIER header Including the BIER Label, which has a length of (12+BSL/8) octets.

Node D need to have a P-CAPABILITY, for it need to Process BitString in each packet to determin whether to replicate to a special LEAF, and then disposit the whole BIER header Including the BIER Label and forward the IP multicast packet further. Node D also need to do the disposition as well, which is called a D-CAPABILITY. D-CAPABILITY means to disposit the BIER header including or excluding the BIER Label in the begining. Here PopBIERincluding means pop the BIER header including the BIER Label, while PopBIERexcluding means pop the BIER header excluding the BIER Label.

Forwarding Tables on E and F are same to D.

Comparing to the forwarding procedure defined in [RFC8279], there are two benefits of using the customized P2MP based BIER forwarding:

1. Not need to walk every physical neighbor, but only need to walk downstream neighbors on a P2MP tree.

2. Not need to edit the BitString in every packet, but only need to swap the BIER Label.

EXAMPLE 2: Another example with P2MP BUD Nodes.
When Node E receives a MPLS-encapsulation BIER packet, it gets the label and matches the ILM, then does a replication according to the NHLFEs and checks whether to proceed by AND’ing the BitString in the replicated packet and the F-BM in the NHLFE/LEAF entry. When the AND’ing result is non-zero for the second LEAF, then do a POP to the packet to dispose the whole BIER header, which has a length of \((12 + \text{BSL}/8)\) octets.

Node E, which is a BUD Node, has both the two capacities: P-CAPABILITY and D-CAPABILITY. P-CAPABILITY is needed to be used for every NHLFE/LEAF, and D-CAPABILITY is needed for the NHLFE that has a PopBIERincluding flag.

5.3. When Mid, Leaf or Bud nodes do not support P-CAPABILITY

The procedures of Section 5.2 presuppose that, within a given BIER domain, all the nodes adjacent to a given BFR in a given routing underlay are also BFRs. However, it is possible to use BIER even when this is not the case. In this section, we describe procedures
that can be used if the routing underlay is a P2MP tree with BIER information in the domain.

For a P2MP tree, every node has a role of Root, Branch, Leaf, or Bud. The role is determined when the tree is built. The method is suitable for conditions when Mid, Leaf or Bud nodes do not support P-CAPABILITY.

EXAMPLE 1: Take Figure 4 as an example.

If D, F, E support BIER, and C don’t support BIER, then we can configure on C to indicate it to use P2MP for BIER packets forwarding. Then C build a P2MP forwarding entry, while still pass the BIER information in control-plane. For example, D send a P2MP FEC Mapping message to C with a BitMask 0001, F send a P2MP FEC Mapping message to C with a BitMask 0010, and C send a P2MP FEC Mapping message to B with a BitMask, but C build a P2MP forward entry like this:

ILM (inLabel<alloc by C>, action<Replication to TreeID>, Flag=Branch)

NHLFE(TreeID, OutInterface<toD>, OutLabel<alloc by D>)

NHLFE(TreeID, OutInterface<toF>, OutLabel<alloc by F>)

If D don’t support BIER P-CAPABILITY, but it support BIER D-CAPABILITY, then the above method is still valid.

Forwarding Table on D when D don’t have a P-CAPABILITY:

ILM (inLabel<alloc by D>, action<Replication to TreeID>, Flag=Leaf, BSL)

NHLFE(TreeID, flag=PopBIERincluding)

When Node D receive a MPLS-encapsulation BIER packet, it get the Label and match ILM, then do a replication according to the NHLFE but don’t do the check by AND’ing the BitString in the replicated packet and the F-BM in the NHLFE entry. And then do a POP to the packet to dispose the whole BIER header, which has a length of (12+BSL/8) octets.

Another alternative form of Forwarding Table on D can also be the following when D don’t have a P-CAPABILITY:

ILM (inLabel<alloc by D>, action<PopBIERincluding>, Flag=Leaf, BSL)
When Node D receive a MPLS-encapsulation BIER packet, it get the Label and match ILM, then do a POP action according to the ILM to pop the whole (12+BSL/8) octets from the Label position.

EXAMPLE 2: Take BUD Node E in Figure 5 as another example.

Forwarding Table on Bud Node E when E don’t have a P-CAPABILITY:

Forwarding Table on E when E don’t have a P-CAPABILITY:

<table>
<thead>
<tr>
<th>ILM</th>
<th>NHLFE</th>
<th>LEAF</th>
</tr>
</thead>
<tbody>
<tr>
<td>(inLabel&lt;alloc by E&gt;, action&lt;Replication to TreeID&gt;, Flag=Bud, BSL)</td>
<td>NHLFE(TreeID, OutInterface&lt;toF&gt;, OutLabel&lt;alloc by F&gt;)</td>
<td>LEAF(TreeID, flag=PopBIERincluding)</td>
</tr>
</tbody>
</table>

One can see that, this method can support widely Non-BIER Nodes in a network, no matter the node has a Mid, Leaf or Bud role, and would never result in any ingress-replication through unicast tunnel, which may cause a overload on a link.

One can also see that, [RFC8279] only support Non BIER-capability nodes being the Mid nodes, and never allow a BFER nodes to be Non BIER-capability.

5.4. When Leaf or Bud nodes do not support D-CAPABILITY

A more tolerant variant of the above, when Leaf or Bud nodes do not support D-CAPABILITY, would be the following:

EXAMPLE 1: Take Figure 4 as an example.

If D even don’t support BIER P-CAPABILITY or D-CAPABILITY, then POP the whole BIER Header except the first four octets Label field of a packet before it come to D. This requires C to build a forwarding table like this:

Forwarding Table on C (Branch Node):

<table>
<thead>
<tr>
<th>ILM</th>
<th>NHLFE</th>
<th>NHLFE</th>
</tr>
</thead>
<tbody>
<tr>
<td>(inLabel&lt;alloc by E&gt;, action&lt;Replication to TreeID&gt;, Flag=Branch</td>
<td>CheckBS, BSL)</td>
<td>NHLFE(TreeID, OutInterface&lt;toD&gt;, OutLabel&lt;alloc by D&gt;, F-BM&lt;0001&gt;, Flag=PopBIERexcluding)</td>
</tr>
</tbody>
</table>
The Flag PopBIERexcluding means POP the BIER Header excluding the first 4 octets BIER Label in a packet, that is a Length of (8+BSL/8).

If D don’t support BIER P-CAPABILITY or D-CAPABILITY, and C don’t support BIER P-CAPABILITY, then it requires B to build a forwarding table, to ensure the BIER Header except the first four octets Label field of a packet is popped before replicated to C, and requires C to build a forwarding table of a pure P2MP branch, and requires F to build a forwarding table of a pure P2MP Leaf. Their forwarding tables are like below:

Forwarding Table on B (Branch Node):

- ILM (inLabel<alloc by B>, action<Replication to TreeID>, Flag=Branch|CheckBS, BSL)
- NHLFE(TreeID, OutInterface<toC>, OutLabel<alloc by C>, F-MB<0011>, Flag=PopBIERexcluding)
- NHLFE(TreeID, OutInterface<toE>, OutLabel<alloc by E>, F-BM<0100>)

Forwarding Table on C (Branch Node):

- ILM (inLabel<alloc by C>, action<Replication to TreeID>, Flag=Branch)
- NHLFE(TreeID, OutInterface<toF>, OutLabel<alloc by F>)
- NHLFE(TreeID, OutInterface<toF>, OutLabel<alloc by F>)

Forwarding Table on D (Branch Node):

- ILM (inLabel<alloc by D>, action<PopLabel>, Flag=Leaf)

Here PopLabel mean to pop the Label, which is in fact a P2MP LSP Label. It is a basic capability of any LSR.

Forwarding Table on F (Branch Node):

- ILM (inLabel<alloc by F>, action<PopLabel>, Flag=Leaf)

Here PopLabel mean to pop the Label, which is in fact a P2MP LSP Label. It is a basic capability of any LSR, and the Forwarding table on F is in fact a P2MP one.

Note that, although F support BIER, which means it can deal with a BIER packet, but it must downshift its forwarding table to a pure P2MP one, because the packet it received doesn’t include a BIER...
Header but a P2MP Label packet due to the POP behaving of its upstream node.

EXAMPLE 2: Take Figure 5 as another example.

If E even don't support BIER P-CAPABILITY or D-CAPABILITY, then POP the whole BIER Header Except the first four octets Label field of a packet before it come to D. This requires B to build a forwarding table like this:

Forwarding Table on B (Branch Node):

- ILM (inLabel<alloc by B>, action<Replication to TreeID>, Flag=Branch|CheckBS, BSL)
- NHLFE(TreeID, OutInterface<toC>, OutLabel<alloc by C>, F-MB<0011>)
- NHLFE(TreeID, OutInterface<toE>, OutLabel<alloc by E>, F-BM<0100>, Flag=PopBIERexcluding)

Forwarding Table on E (Bud Node):

- ILM (inLabel<alloc by E>, action<Replication to TreeID>, Flag=Bud)
- NHLFE(TreeID, OutInterface<toF>, OutLabel<alloc by F>)
- LEAF(TreeID, flag=PopLabel)

Forwarding Table on F (Branch Node):

- ILM (inLabel<alloc by F>, action<PopLabel>, Flag=Leaf)

Note that, althought F support BIER, which means it can deal with a BIER packet, but it must downshift its forwarding table to a pure P2MP Leaf, because the packet it received doesn’t include a BIER Header but a P2MP Label packet due to the POP behaving of its upstream node.

One can see that, when some Leaf or Bud nodes even don’t have a D-CAPABILITY, we can do a POP action to dispositing the BIER header excluding the BIER Label in the begining before the packet arrive the node. This is similar to a Penultimate Hop Popping in a P2P LSP, and we call it Upstream Hop Popping in P2MP based BIER.
6. Provisioning Considerations

P2MP based BIER use concepts of both P2MP and BIER. Some provisioning considerations list below:

Sub-domain:

In P2MP based BIER, every P2MP is a specific BIER underlay topology, and an implicit Sub-domain. RSVP-TE/MLDP/PIM build the BIER information of the implicit sub-domain when building the P2MP LSP or PIM tree. MVPN get the implicit sub-domain by provisioning.

In the following conditions, there may be requirements to configure an explicit sub-domain ID for P2MP based BIER:

1. P2MP LSP based BIER, use the native procedure of forwarding described in [RFC8279], which require Consistent Per-Sub-domain BIFT.

2. P2MP LSP based BIER is shared by multiple VPNs, and an explicit sub-domain ID is configured as anchor for using by these VPNs.

When explicitly configuring a sub-domain ID for P2MP LSP based BIER, the ID should be greater than 255. For the [0-255] has been defined to use by IGP, BGP and MVPN, as specified in [I-D.ietf-bier-ospf-bier-extensions], [I-D.ietf-bier-isis-extensions], [I-D.ietf-bier-idr-extensions] and [I-D.ietf-bier-mvpn].

BFR-prefix:

In P2MP LSP based BIER, every BFR is also a LSR. So the BFR-prefix in the sub-domain is by default identified by LSR-id. Additionally, When BFR/LSR is also a MVPN PE, BFR-prefix is also the same as Originating Router’s IP Address of x-PMSI A-D route or Leaf A-D route.

BFR-id:

When using protocols like RSVP-TE, which initializes P2MP LSP from a specific Ingress Node, BFR-id which is unique in P2MP LSP scope, can be auto-provisioned by Ingress Node, or conventionally configure on every Egress Nodes.

BSL and BIER-MPLS Label Block Size:
In P2MP LSP based BIER, Every P2MP LSP or implicit sub-domain requires a single BSL, and a specific BIER-MPLS Label block size for this BSL.

VPN-Label:

The P2MP based BIER ‘P-tunnel’ can be shared by multiple VPNs or a single VPN. When a P2MP based BIER being shared by multiple VPNs, an Upstream-assigned VPN-Label is required. It can be auto-provisioned or manual configured by the BFIR or Ingress LSR.

In fact, [RFC6513] has defined the method of "Aggregating Multiple MVPNs on a Single P-Tunnel". But unfortunately it is not widely deployed because of the serious trade-off between state saving and bandwidth waste. The BIER encapsulation and forwarding method give it a chance to eliminate the trade-off while gaining a completely state saving.

Even when such an aggregating is not used, it is still adequate to use BIER to save state by sharing one P2MP based BIER "p-tunnel" for multi flows in one specific VPN.

For seamless transitioning from legacy MVPN deployment and existing network, it is recommended not to use such an aggregating, as well as to use such an aggregating.

7. IANA Considerations

Allocation is expected from IANA for two new tunnel type codepoints for "RSVP-TE built P2MP based BIER", "MLDP built P2MP based BIER" and "PIM built P2MP based BIER" from the "P-Multicast Service Interface Tunnel (PMSI Tunnel) Tunnel Types" registry.

8. Security Considerations

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

9. Acknowledgements

TBD

10. References
10.1. Normative References

[I-D.ietf-bess-mvpn-expl-track]
Dolganow, A., Kotalwar, J., Rosen, E., and Z. Zhang,
"Explicit Tracking with Wild Card Routes in Multicast VPN",
draft-ietf-bess-mvpn-expl-track-08 (work in progress), February 2018.

[I-D.ietf-bess-mvpn-fast-failover]
Morin, T. and R. Kebler, "Multicast VPN fast upstream failover",
draft-ietf-bess-mvpn-fast-failover-02 (work in progress), March 2017.

[I-D.ietf-bier-oidr-extensions]
Xu, X., Chen, M., Patel, K., Wijnands, I., and T.
Przygienda, "BGP Extensions for BIER",
draft-ietf-bier-oidr-extensions-05 (work in progress), March 2018.

[I-D.ietf-bier-isis-extensions]
Ginsberg, L., Przygienda, T., Aldrin, S., and Z. Zhang,
"BIER support via ISIS",
draft-ietf-bier-isis-extensions-09 (work in progress), February 2018.

[I-D.ietf-bier-mvpn]
Rosen, E., Sivakumar, M., Aldrin, S., Dolganow, A., and T.
Przygienda, "Multicast VPN Using BIER",
draft-ietf-bier-mvpn-10 (work in progress), February 2018.

[I-D.ietf-bier-ospf-bier-extensions]
Psenak, P., Kumar, N., Wijnands, I., Dolganow, A.,
Przygienda, T., Zhang, Z., and S. Aldrin,
"OSPF Extensions for BIER",

RFC 3031, DOI 10.17487/RFC3031, January 2001,

Yasukawa, Ed., "Extensions to Resource Reservation Protocol - Traffic Engineering (RSVP-TE) for Point-to-Multipoint TE Label Switched Paths (LSPs)",
RFC 4875, DOI 10.17487/RFC4875, May 2007,


10.2. Informative References

Authors’ Addresses

Jingrong Xie
Huawei
Q15 Huawei Campus, No.156 Beiqing Rd.
Beijing  100095
China
Email: xiejingrong@huawei.com

Mike McBride
Huawei Technologies
Email: mmcbride7@gmail.com

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

Liang Geng
China Mobile
Beijing 100053
China
Email: gengliang@chinamobile.com