Internet Engineering Task Force
Internet-Draft
Intended status: Experimental
Expires: April 30, 2018

IJ. Wijnands, Ed.
Cisco Systems, Inc.
E. Rosen, Ed.
Juniper Networks, Inc.
A. Dolganow
Nokia
J. Tantsura
Individual
S. Aldrin
Google, Inc.
I. Meilik
Broadcom

October 27, 2017

Encapsulation for Bit Index Explicit Replication in MPLS and non-MPLS Networks
draft-ietf-bier-mpls-encapsulation-12

Abstract

Bit Index Explicit Replication (BIER) is an architecture that provides optimal multicast forwarding through a "multicast domain", without requiring intermediate routers to maintain any per-flow state or to engage in an explicit tree-building protocol. When a multicast data packet enters the domain, the ingress router determines the set of egress routers to which the packet needs to be sent. The ingress router then encapsulates the packet in a BIER header. The BIER header contains a bitstring in which each bit represents exactly one egress router in the domain; to forward the packet to a given set of egress routers, the bits corresponding to those routers are set in the BIER header. The details of the encapsulation depend on the type of network used to realize the multicast domain. This document specifies a BIER encapsulation that can be used in an MPLS network, or with slight differences, in a non-MPLS network.

Status of This Memo

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

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

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any

time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on April 30, 2018.

Copyright Notice

Copyright (c) 2017 IETF Trust and the persons identified as the document authors. All rights reserved.

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

Table of Contents

1. Introduction 3
2. BIER Header 4
2.1. In MPLS Networks 5
2.1.1. Encapsulation Initial Four Octets 5
2.1.1.1. The BIER-MPLS Label 5
2.1.1.2. Other Fields of the Initial Four Octets 7
2.1.2. Remainder of Encapsulation 8
2.1.3. Further Encapsulating a BIER Packet 10
2.2. In Non-MPLS Networks 11
2.2.1. Encapsulation Initial Four Octets 11
2.2.1.1. The BIFT-id 11
2.2.1.2. Other Fields of the Initial Four Octets 12
2.2.2. Remainder of Encapsulation 12
2.2.3. Further Encapsulating a BIER Packet 13
3. Imposing and Processing the BIER Encapsulation 14
4. IANA Considerations 16
5. IEEE Considerations 17
6. Security Considerations 17
7. Acknowledgements 18
8. Contributor Addresses 18
9. References 20
9.1. Normative References 20
9.2. Informative References 21
Authors' Addresses 22

1. Introduction

[BIER_ARCH] describes a new architecture for the forwarding of multicast data packets. Known as "Bit Index Explicit Replication" (BIER), that architecture provides optimal forwarding of multicast data packets through a "multicast domain". It does so without requiring any explicit tree-building protocol and without requiring intermediate nodes to maintain any per-flow state.

This document will use terminology defined in [BIER_ARCH].

A router that supports BIER is known as a "Bit-Forwarding Router" (BFR). A "BIER domain" is a connected set of Bit-Forwarding Routers (BFRs), each of which has been assigned a BFR-prefix. A BFR-prefix is a routable IP address of a BFR, and is used by BIER to identify a BFR. A packet enters a BIER domain at an ingress BFR (BFIR), and leaves the BIER domain at one or more egress BFRs (BFERs). As specified in [BIER_ARCH], each BFR of a given BIER domain is provisioned to be in one or more "sub-domains" (SDs). In the context of a given SD, each BFIR and BFER must have a BFR-id that is unique within that SD. A BFR-id is just a number in the range [1,65535] that, relative to a BIER SD, identifies a BFR uniquely.

As described in [BIER_ARCH], BIER requires that multicast data packets be encapsulated with a header that provides the information needed to support the BIER forwarding procedures. This information includes the SD to which the packet has been assigned, a Set-Id (SI), a BitString, and a BitStringLength (BSL). Together these values are used to identify the set of BFERs to which the packet must be delivered.

This document defines an encapsulation that can be used in either MPLS networks or non-MPLS networks. However, the construction and processing of the BIER header is slightly different in MPLS networks than in non-MPLS networks. In particular:

- The handling of certain fields in the encapsulation header (the "BIER header") is different depending upon whether the underlying network is an MPLS network or not.

- In an MPLS network, the first four octets of a BIER header is also the bottom entry (the last four octets) of an MPLS label stack.

The MPLS-based encapsulation is explained in detail in Section 2.1. The differences between the MPLS-based encapsulation and the non-MPLS encapsulation is explained in Section 2.2.

Following the BIER header is the "payload". The payload may be an IPv4 packet, an IPv6 packet, an ethernet frame, an MPLS packet, or an OAM packet. (The use of BIER with other payload types is also possible, but is not further discussed in this document.) The BIER header contains information (the Next Protocol field) identifying the type of the payload.

If the payload is an MPLS packet, then an MPLS label stack immediately follows the BIER header. The top label of this MPLS label stack may be either a downstream-assigned label [RFC3032] or an upstream-assigned label [RFC5331].

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. BIER Header

The BIER header is shown in Figure 1.

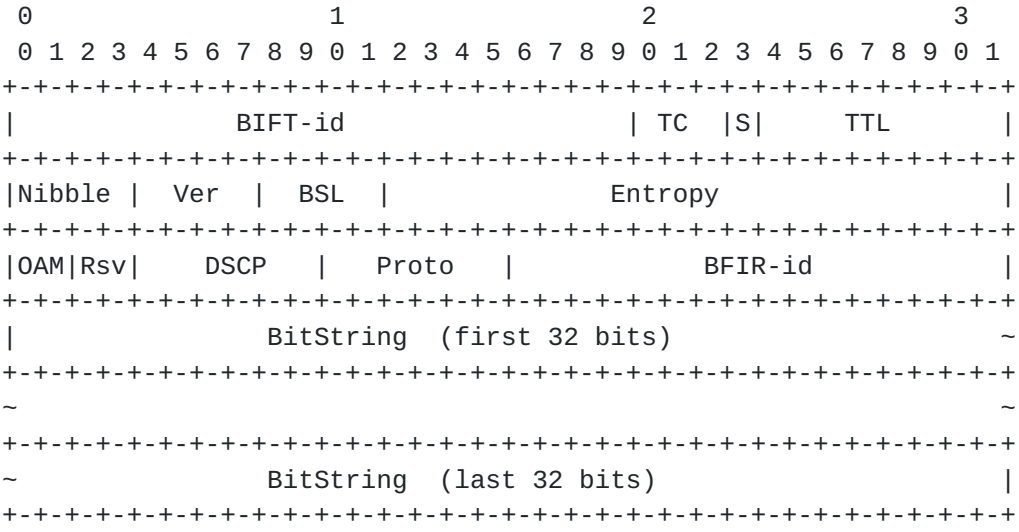


Figure 1: BIER Header

The BIFT-id represents a particular Bit Index Forwarding Table (BIFT); see Section 6.4 of [BIER_ARCH]. As explained in [BIER_ARCH], each BIFT corresponds to a particular combination of SD, BSL, and SI.

Section 2.1 explains how the fields of the encapsulation header are used in MPLS networks. For those fields that are used differently in non-MPLS networks, Section 2.2 explains the differences.

The default BitStringLength value for the encapsulations defined in this document is 256. See Section 3 of [BIER_ARCH] for a discussion of the default BitStringLength value.

2.1. In MPLS Networks

2.1.1. Encapsulation Initial Four Octets

2.1.1.1. The BIER-MPLS Label

As stated in [BIER_ARCH], when a BIER domain is also an IGP domain, IGP extensions can be used by each BFR to advertise the BFR-id and BFR-prefix. The extensions for OSPF are given in [OSPF_BIER_EXTENSIONS]. The extensions for IS-IS are given in [ISIS_BIER_EXTENSIONS].

When a particular BIER domain is both an IGP domain and an MPLS network, we assume that each BFR will also use IGP extensions to advertise a set of one or more "BIER-MPLS" labels. When the domain contains a single SD, a given BFR needs to advertise one such label for each combination of SI and BSL. If the domain contains multiple SDs, a BFR needs to advertise one such label per SI per BSL for each SD.

In some environments, the only routing protocol in a BIER domain might be BGP; in this case, the BGP extensions described in [BGP_BIER_EXTENSIONS] can be used to advertise the necessary set of BIER-MPLS labels.

The BIER-MPLS labels are locally significant (i.e., unique only to the BFR that advertises them) downstream-assigned MPLS labels. Penultimate hop popping ([RFC3031]) MUST NOT be applied to a BIER-MPLS label.

Suppose for example that there is a single SD (the default SD), that the network is using a BSL of 256, and that all BFERs in the SD have BFR-ids in the range [1,512]. Since each BIER BitString is 256 bits long, this requires the use of two SIs: SI=0 and SI=1. So each BFR will advertise, via IGP extensions, two MPLS labels for BIER: one corresponding to SI=0 and one corresponding to SI=1. The advertisements of these labels will also bind each label to the default SD and to the BSL 256.

As another example, suppose a particular BIER domain contains 2 SDs (SD 0 and SD 1), supports 2 BSLs (256 and 512), and contains 1024 BFRs. A BFR that is provisioned for both SDs, and that supports both BSLs, would have to advertise the following set of BIER-MPLS labels:

L1: corresponding to SD 0, BSL 256, SI 0.

L2: corresponding to SD 0, BSL 256, SI 1.

L3: corresponding to SD 0, BSL 256, SI 2.

L4: corresponding to SD 0, BSL 256, SI 3.

L5: corresponding to SD 0, BSL 512, SI 0.

L6: corresponding to SD 0, BSL 512, SI 1.

L7: corresponding to SD 1, BSL 256, SI 0.

L8: corresponding to SD 1, BSL 256, SI 1.

L9: corresponding to SD 1, BSL 256, SI 2.

L10: corresponding to SD 1, BSL 256, SI 3.

L11: corresponding to SD 1, BSL 512, SI 0.

L12: corresponding to SD 1, BSL 512, SI 1.

The above example should not be taken as implying that the BFRs need to advertise 12 individual labels. For instance, instead of advertising a label for <SD 1, BSL 512, SI 0> and a label for <SD 1, BSL 512, SI 1>, a BFR could advertise a contiguous range of labels (in this case, a range containing exactly two labels) corresponding to <SD 1, BSL 512>. The first label in the range could correspond to SI 0, and the second to SI 1. The precise mechanism for generating and forming the advertisements is outside the scope of this document. See [OSPF_BIER_EXTENSIONS] and [ISIS_BIER_EXTENSIONS].

The BIER-MPLS label corresponding to a particular combination of SD, SI, and BSL is interpreted as representing the BIFT that corresponds to that same combination of SD, SI, and BSL. That is, the BIER-MPLS label performs the function of a BIFT-id. This label value is carried in the BIFT-id field of the BIER encapsulation.

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

When a BFR receives an MPLS packet, and the next label to be processed is one of its BIER-MPLS labels, it will assume that the

remainder of the BIER header (see Section 2.1.2) immediately follows the stack.

Note that in practice, labels only have to be assigned if they are going to be used. If a particular BIER domain supports BSLs 256 and 512, but some SD, say SD 1, only uses BSL 256, then it is not necessary to assign labels that correspond to the combination of SD 1 and BSL 512.

2.1.1.2. Other Fields of the Initial Four Octets

TC:

The "Traffic Class" field ([RFC5462]) has its usual meaning in an MPLS label stack entry.

S bit:

When a BIER packet is traveling through an MPLS network, the high-order 20 bits of the initial four octets of the BIER encapsulation contain an MPLS label in the BIFT-id field. These four octets are treated as the final entry in the packet's MPLS label stack. Hence the S bit (see [RFC3032]) MUST be set to 1. If there are any MPLS label stack entries immediately preceding the BIER encapsulation, the S bit of those label stack entries MUST be set to 0.

TTL:

This is the usual MPLS "Time to Live" field ([RFC3032]). When a BIER packet is received, its "incoming TTL" (see below) is taken from this TTL field.

When a BIER packet is forwarded to one or more BFR adjacencies, the BIER-MPLS label carried by the forwarded packet MUST have a TTL field whose value is one less than that of the packet's incoming TTL.

If a BIER packet's incoming TTL is 1 or greater, and one of the bits in its BitString identifies the current BFR, then the current BFR is a BFER for the packet. Therefore the current BFR MUST process the packet as a BFER, e.g., by removing the BIER encapsulation and processing the payload based on the contents of the Proto field.

If the incoming TTL is 0, the packet is considered to be "expired". If the incoming TTL is 1, and the BitString has a bit set that does not identify the current BFR, the packet is also

considered to be expired. Expired packets SHOULD be passed to an error handling procedure. (Optional implementation-specific rate limiting may be applied to control the rate at which packets are passed to the error handling procedure.) Specification of the error handling procedure is outside the scope of this document.

Note that if a received BIER packet has an incoming TTL of 1, and its BitString has a bit set identifying the current BFR, the payload MUST be processed by the current BFR, but the packet MUST NOT be forwarded further, and the packet SHOULD also be passed to the error handling procedures for expired packets (subject to any implementation-specific rate limiting).

2.1.2. Remainder of Encapsulation

Nibble:

This field is set to the binary value 0101; this ensures that the MPLS ECMP logic will not confuse the remainder of the BIER header with an IP header or with the header of a pseudowire packet. In an MPLS network, if a BFR receives a BIER packet with any other value in the first nibble after the label stack, it SHOULD discard the packet and log an error.

Ver:

This 4-bit field identifies the version of the BIER header. This document specifies version 0 of the BIER header. If a packet is received by a particular BFR, and that BFR does not support the specified version of the BIER header, the BFR MUST discard the packet and log an error.

The value 0xF is reserved for experimental use; that value MUST NOT be assigned by any future IETF document or by IANA.

BSL:

This 4-bit field encodes the length in bits of the BitString.

Note: When parsing the BIER header, a BFR MUST infer the length of the BitString from the BIFT-id, and MUST NOT infer it from the value of this field. This field is present only to enable off-line tools (such as LAN analyzers) to parse the BIER header.

If k is the length of the BitString, the value of this field is log2(k)-5. However, only certain values are supported:

1: 64 bits

2: 128 bits

3: 256 bits

4: 512 bits

5: 1024 bits

6: 2048 bits

7: 4096 bits

The value of this field MUST NOT be set to any value other than those listed above. A received packet containing another value in this field SHOULD be discarded, and an error logged. If the value in this field is other than what is expected based on the BIER-MPLS label, the packet SHOULD be discarded and an error logged.

Entropy:

This 20-bit field specifies an "entropy" value that can be used for load balancing purposes. The BIER forwarding process may do equal cost load balancing, in which case the load balancing procedure MUST choose the same path for any two packets that have the same entropy value and the same BitString. Please see Section 6.7 ("Equal Cost Multi-path Forwarding") of [BIER_ARCH] for a more detailed discussion of BIER load balancing procedures.

If a BFIR is encapsulating (as the payload) MPLS packets that have entropy labels, the BFIR MUST ensure that if two such packets have the same MPLS entropy label, they also have the same value of the BIER entropy field.

OAM:

By default, these two bits are set to zero by the BFIR, and are not modified by other BFRs. These two bits have no effect on the path taken by a BIER packet, and have no effect on the quality of service applied to a BIER packet.

The use of these bits in other than the default manner is OPTIONAL. Specification of the non-default use or uses of these bits is outside the scope of this document. (See [BIER-PMM] for an example of such a specification.)

Rsv:

These 2 bits are currently unused. They SHOULD be set to zero upon transmission, and MUST be ignored upon reception.

DSCP:

By default, this 6-bit field is not used in MPLS networks. The default behavior is that all 6 bits SHOULD be set to zero upon transmission, and MUST be ignored upon reception.

Non-default use of this field in MPLS networks is outside the scope of this document.

Proto:

This 6-bit "Next Protocol" field identifies the type of the payload. (The "payload" is the packet or frame immediately following the BIER header.) IANA has been requested to create a registry of "BIER Next Protocol Identifiers". This field is to be populated with the appropriate entry from that registry.

If a BFER receives a BIER packet, but does not recognize (or does not support) the value of the Next Protocol field, the BFER SHOULD discard the packet and log an error.

BFIR-id:

By default, this is the BFR-id of the BFIR, in the SD to which the packet has been assigned. The BFR-id is encoded in the 16-bit field as an unsigned integer in the range [1,65535].

Certain applications may require that the BFIR-id field contain the BFR-id of a BFR other than the BFIR. However, that usage of the BFIR-id field is outside the scope of the current document.

BitString:

The BitString that, together with the packet's SI and SD, identifies the destination BFERs for this packet. Note that the SI and SD for the packet are not carried explicitly in the BIER header, as a particular BIFT-id always corresponds to a particular SI and SD.

2.1.3. Further Encapsulating a BIER Packet

Sending a BIER packet from one BFR to another may require the packet to be further encapsulated. For example: in some scenarios it may be necessary to encapsulate a BIER packet in an ethernet frame; in other scenarios it may be necessary to encapsulate a BIER packet in a UDP

packet. In such cases, the BIER packet itself is the payload of an "outer" encapsulation.

In this document, we assume that the frame or packet carrying a BIER packet as its payload is a unicast frame or packet. That is, although a BIER packet is a multicast packet, we assume that the frame or packet carrying the BIER packet as its payload is unicast from one BFR to the next.

Generally the outer encapsulation has a codepoint identifying the "next protocol". The outer encapsulation's "next protocol" codepoint for MPLS MUST be used. If a particular outer encapsulation has a codepoint for "MPLS with Downstream-Assigned Label" and a different codepoint for "MPLS with Upstream-Assigned Label", the codepoint for "MPLS with Downstream-Assigned Label" MUST be used.

For example, if a BIER packet is encapsulated in an ethernet frame, the ethertype MUST be 0x8847 ([RFC5332]), which is the ethertype for a unicast ethernet frame that carries an MPLS packet whose label stack beings with a downstream-assigned label.

In the special case where the outer encapsulation is MPLS, the outer encapsulation has no "next protocol" codepoint. All that is needed to encapsulate the BIER packet is to push more MPLS label stack entries (with S bit clear) on the BIER packet's label stack.

If two BIER packets have the same value in the entropy field of their respective BIER headers, and if both are placed in an outer encapsulation, it is desirable for the outer encapsulation to preserve the fact that the two packets have the same entropy. If the outer encapsulation is MPLS, and if the MPLS entropy label ([RFC6790]) is in use in a given deployment, one way to do this is to copy the value of the BIER header entropy field into an MPLS entropy label.

2.2. In Non-MPLS Networks

2.2.1. Encapsulation Initial Four Octets

2.2.1.1. The BIFT-id

In non-MPLS networks, a BIFT-id MUST be assigned for every combination of <SD, SI, BSL> that is to be used in that network. The correspondence between a BIFT-id and a particular <SD, SI, BSL> triple is unique throughout the BIER domain, and is known to all the BFRs in the BIER domain.

The means by which the BIFT-ids are assigned, and the means by which these assignments are made known to the BFRs, are outside the scope of this document.

In an MPLS network, since the BIFT-id is an MPLS label, its value may be changed as a BIER packet goes from BFR to BFR. In a non-MPLS network, since the BIFT-id is domain-wide unique, it is not expected to change as a BIER packet travels.

2.2.1.2. Other Fields of the Initial Four Octets

TC:

By default, the TC field has no significance in a non-MPLS network. The default behavior is that this field SHOULD be set to the binary value 000 upon transmission, and MUST be ignored upon reception.

Non-default use of this field in non-MPLS networks is outside the scope of this document.

S bit:

The S bit has no significance in a non-MPLS network. It SHOULD be set to 1 upon transmission, but it MUST be ignored upon reception.

TTL:

This is the BIER "Time to Live" field. Its purpose is to prevent BIER packets from looping indefinitely in the event of improper operation of the control plane. When a BIER packet is received, its "incoming TTL" (see below) is taken from this TTL field.

The effect of this field on the processing of a BIER packet is described in Section 2.1.1.2.

2.2.2. Remainder of Encapsulation

Nibble:

This field SHOULD be set to 0000 upon transmission, but MUST be ignored upon reception.

Ver:

See Section 2.1.2.

BSL:

See Section 2.1.2.

Entropy:

See Section 2.1.2.

OAM:

See Section 2.1.2.

Rsv:

See Section 2.1.2.

DSCP:

This 6-bit field MAY be used to hold a Differentiated Services Codepoint ([RFC2474]). The significance of this field is outside the scope of this document.

Proto:

See Section 2.1.2.

BFIR-id:

See Section 2.1.2.

BitString:

See Section 2.1.2.

2.2.3. Further Encapsulating a BIER Packet

Sending a BIER packet from one BFR to another may require the packet to be further encapsulated. For example: in some scenarios it may be necessary to encapsulate a BIER packet in an ethernet frame; in other scenarios it may be necessary to encapsulate a BIER packet in in a UDP packet. In such cases, the BIER packet itself is the payload of an "outer" encapsulation.

In this document, we assume that the frame or packet carrying a BIER packet as its payload is a unicast frame or packet. That is, although a BIER packet is a multicast packet, we assume that the frame or packet carrying the BIER packet as its payload is unicast from one BFR to the next.

Generally the outer encapsulation has a codepoint identifying the "next protocol". This codepoint MUST be set to a value that means "Non-MPLS BIER". In particular, a codepoint that means "MPLS" (with either upstream-assigned or downstream-assigned labels) MUST NOT be used.

By requiring the use of a distinct codepoint for "non-MPLS BIER", we allow for deployment scenarios where non-MPLS BIER can coexist with non-BIER MPLS. The BIFT-id values used by the former will not conflict with MPLS label values used by the latter.

Therefore, if a non-MPLS BIER packet is encapsulated in an ethernet header, the ethertype MUST NOT be 0x8847 or 0x8848 ([RFC5332]). IEEE has been requested to assign ethertype TBD1 for non-MPLS BIER packets.

In the special case where the outer encapsulation is MPLS, the outer encapsulation has no "next protocol" codepoint. If it is necessary to use MPLS as an outer encapsulation for BIER packets, it is RECOMMENDED to use the MPLS encapsulation for BIER. Procedures for encapsulating a non-MPLS BIER packet in MPLS are outside the scope of this document.

If two BIER packets have the same value in the entropy field of their respective BIER headers, and if both are placed in an outer encapsulation, it is desirable for the outer encapsulation to preserve the fact that the two packets have the same entropy.

3. Imposing and Processing the BIER Encapsulation

Each BFIR is expected to know the Maximum Transit Unit (MTU) of the BIER domain. This may be known by provisioning, or by some other method outside the scope of this document. Each BFIR also knows the size of the BIER encapsulation. Thus each BFIR can deduce the maximum size payload that can be encapsulated in a BIER packet. We will refer to this payload size as the BIER-MTU.

If a BFIR receives a multicast packet from outside the BIER domain, and the packet size exceeds the BIER-MTU, the BFIR takes whatever action is appropriate to take when receiving a multicast packet that is too large to be forwarded to all its next hops. If the appropriate action is to drop the packet and advertise an MTU to the source, then the BFIR drops the packet and advertises the BIER-MTU. If the appropriate action is to fragment the packet, then the procedures of this section are applied, in sequence, to each fragment.

When a BFIR processes a multicast packet (or fragment thereof) from outside the BIER domain, the BFIR carries out the following procedure:

1. By consulting the "multicast flow overlay" [BIER_ARCH], it determines the value of the "Proto" field.

2. By consulting the multicast flow overlay, it determines the set of BFERs that must receive the packet.

3. If more than one SD is supported, the BFIR assigns the packet to a particular SD. Procedures for determining the SD to which a particular packet should be assigned are outside the scope of this document.

4. The BFIR looks up the BFR-id, in the given SD, of each of the BFERs.

5. The BFIR converts each such BFR-id into (SI, BitString) format, as described in [BIER_ARCH].

6. All such BFR-ids that have the same SI can be encoded into the same BitString. Details of this encoding can be found in [BIER_ARCH]. For each distinct SI that occurs in the list of the packet's destination BFERs:

   a. The BFIR makes a copy of the multicast data packet, and encapsulates the copy in a BIER header (see Section 2). The BIER header contains the BitString that represents all the destination BFERs whose BFR-ids (in the given SD) correspond to the given SI. It also contains the BFIR's BFR-id in the SD to which the packet has been assigned.

      N.B.: For certain applications, it may be necessary for the BFIR-id field to contain the BFR-id of a BFR other than the BFIR that is creating the header. Such uses are outside the scope of this document.

   b. The BFIR then applies to that copy the forwarding procedure of [BIER_ARCH]. This may result in one or more copies of the packet (possibly with a modified BitString) being transmitted to a neighboring BFR.

   c. If the non-MPLS BIER encapsulation is being used, the BIFT-id field is set to the BIFT-id that corresponds to the packet's <SD, SI, BSL>. The TTL is set according to policy.

If the MPLS BIER encapsulation is being used, the BFIR finds the BIER-MPLS label that was advertised by the neighbor as corresponding to the given <SD, SI, BSL>. An MPLS label stack is then prepended to the packet. This label stack [RFC3032] will contain one label, the aforementioned BIER-MPLS label. The "S" bit MUST be set, indicating the end of the MPLS label stack. The TTL field of this label stack entry is set according to policy.

d. The packet may then be transmitted to the neighboring BFR. (In an MPLS network, this may result in additional MPLS labels being pushed on the stack. For example, if an RSVP-TE tunnel is used to transmit packets to the neighbor, a label representing that tunnel would be pushed onto the stack.)

When an intermediate BFR is processing a received MPLS packet, and one of the BFR's own BIER-MPLS labels rises to the top of the label stack, the BFR infers the BSL from the label. The SI and SD are also implicitly identified by the label. The BFR then follows the forwarding procedures of [BIER_ARCH]. If it forwards a copy of the packet to a neighboring BFR, it first swaps the label at the top of the label stack with the BIER-MPLS label, advertised by that neighbor, that corresponds to the same <SD, SI, BSL>. Note that when this swap operation is done, the TTL field of the BIER-MPLS label of the outgoing packet MUST be one less than the "incoming TTL" of the packet, as defined in Section 2.1.1.1.

When an intermediate BFR is processing a received non-MPLS BIER packet, the BFR infers the BSL from the BIFT-id. The SI and SD are also implicitly identified by the BIFT-id. The BFR then follows the forwarding procedures of [BIER_ARCH].

If the BIER payload is an MPLS packet, the BIER header is followed by an MPLS label stack. This stack is separate from any MPLS stack that may precede the BIER header. For an example of an application where it is useful to carry an MPLS packet as the BIER payload, see [BIER_MVPN]. If the BIER encapsulation's Proto field indicates that the payload is an MPLS packet with an upstream-assigned at the top of the stack, the upstream-assigned label is interpreted in the context of <BFIR-id,sub-domain-id>. Note that the sub-domain-id must be inferred from the BIFT-id.

4. IANA Considerations

IANA is requested to set up a registry called "BIER Next Protocol Identifiers". The registration policy for this registry is "IETF Review" ([RFC8126] and [RFC7120]).

Values in this registry must come from the range 0-63.

The initial values in the BIER Next Protocol Identifiers registry are:

0: Reserved (not to be assigned by IANA).

1: MPLS packet with downstream-assigned label at top of stack.

2: MPLS packet with upstream-assigned label at top of stack

3: Ethernet frame.

4: IPv4 packet.

5: OAM packet (reference: [BIER_PING])

6: IPv6 packet.

63: Reserved (not to be assigned by IANA).

5. IEEE Considerations

IEEE has been requested to assign ethertype TBD1 for non-MPLS BIER packets.

6. Security Considerations

Insofar as this document makes use of MPLS, it inherits any security considerations that apply to the use of the MPLS data plane.

If a BIER encapsulation header is modified in ways other than those specified in [BIER_ARCH] and in this document, packets may be lost, stolen, or otherwise misdelivered. Such modifications are likely to go undetected, as the BIER encapsulation does not provide cryptographic integrity protection.

Layer 2 encryption can be used to ensure that a BIER-encapsulated packet is not altered while in transit between adjacent BFRs. If a BFR itself is compromised, there is no way to prevent the compromised BFR from making illegitimate modifications to the BIER header, or to prevent it from misforwarding or misdelivering the a BIER-encapsulated packet.

If the routing underlay (see Section 4.1 of [BIER_ARCH]) is based on a unicast routing protocol, BIER assumes that the routers participating in the unicast routing protocol have not been compromised. BIER has no procedures to ensure that the unicast

routing adjacencies have not been compromised; that falls within the scope of whatever unicast routing protocols are being used.

BIER-encapsulated packets should generally not be accepted from untrusted interfaces or tunnels. For example, an operator may wish to have a policy of accepting BIER-encapsulated packets only from interfaces to trusted routers, and not from customer-facing interfaces.

There may be applications that require a BFR to accept a BIER-encapsulated packet from an interface to a system that is not controlled by the network operator. For instance, there may be an application in which a Virtual Machine in a Data Center submits BIER-encapsulated packets to a router. In such a case, it is desirable to verify that the packet is from a legitimate source, and that its BitString denotes only systems to which that source is allowed to send. However, the BIER encapsulation itself does not provide a way to verify that the source is legitimate, or that the source is really the system denoted by the BFIR-id, or that the source is allowed to set any particular set of bits in the BitString.

Insofar as this document relies upon IGP extensions, it inherits any security considerations that apply to the IGP.

The security considerations of [BIER_ARCH] also apply.

7. Acknowledgements

The authors wish to thank Rajiv Asati, John Bettink, Nagendra Kumar, Christian Martin, Neale Ranns, Greg Shepherd, Ramji Vaithianathan, Xiaohu Xu, and Jeffrey Zhang for their ideas and contributions to this work.

8. Contributor Addresses

Below is a list of other contributing authors in alphabetical order:

Mach (Guoyi) Chen
Huawei

Email: mach.chen@huawei.com

Arkadiy Gulko
Thomson Reuters
195 Broadway
New York NY 10007
United States

Email: arkadiy.gulko@thomsonreuters.com

Wim Henderickx
Nokia
Copernicuslaan 50
Antwerp 2018
Belgium

Email: wim.henderickx@nokia.com

Martin Horneffer
Deutsche Telekom
Hammer Str. 216-226
Muenster 48153
Germany

Email: Martin.Horneffer@telekom.de

Uwe Joorde
Deutsche Telekom
Hammer Str. 216-226
Muenster D-48153
Germany

Email: Uwe.Joorde@telekom.de

Tony Przygienda
Juniper Networks, Inc.
1194 N. Mathilda Ave.
Sunnyvale, California 94089
United States

Email: prz@juniper.net

9. References

9.1. Normative References

[BIER_ARCH] Wijnands, IJ., Rosen, E., Dolganow, A., Przygienda, T., and S. Aldrin, "Multicast using Bit Index Explicit Replication", internet-draft draft-ietf-bier-architecture-08, September 2017.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC2474] Nichols, K., Blake, S., Baker, F., and D. Black, "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers", RFC 2474, DOI 10.17487/RFC2474, December 1998, <https://www.rfc-editor.org/info/rfc2474>.

[RFC3031] Rosen, E., Viswanathan, A., and R. Callon, "Multiprotocol Label Switching Architecture", RFC 3031, DOI 10.17487/RFC3031, January 2001, <https://www.rfc-editor.org/info/rfc3031>.

[RFC3032] Rosen, E., Tappan, D., Fedorkow, G., Rekhter, Y., Farinacci, D., Li, T., and A. Conta, "MPLS Label Stack Encoding", RFC 3032, DOI 10.17487/RFC3032, January 2001, <https://www.rfc-editor.org/info/rfc3032>.

[RFC5331] Aggarwal, R., Rekhter, Y., and E. Rosen, "MPLS Upstream Label Assignment and Context-Specific Label Space", RFC 5331, DOI 10.17487/RFC5331, August 2008, <https://www.rfc-editor.org/info/rfc5331>.

[RFC5332] Eckert, T., Rosen, E., Ed., Aggarwal, R., and Y. Rekhter, "MPLS Multicast Encapsulations", RFC 5332, DOI 10.17487/RFC5332, August 2008, <https://www.rfc-editor.org/info/rfc5332>.

[RFC5462] Andersson, L. and R. Asati, "Multiprotocol Label Switching (MPLS) Label Stack Entry: "EXP" Field Renamed to "Traffic Class" Field", RFC 5462, DOI 10.17487/RFC5462, February 2009, <https://www.rfc-editor.org/info/rfc5462>.

[RFC7120] Cotton, M., "Early IANA Allocation of Standards Track Code Points", BCP 100, RFC 7120, DOI 10.17487/RFC7120, January 2014, <https://www.rfc-editor.org/info/rfc7120>.

[RFC8126] Cotton, M., Leiba, B., and T. Narten, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 8126, DOI 10.17487/RFC8126, June 2017, <https://www.rfc-editor.org/info/rfc8126>.

9.2. Informative References

[BGP_BIER_EXTENSIONS]
Xu, X., Chen, M., Patel, K., Wijnands, I., and A. Przygienda, "BGP Extensions for BIER", internet-draft draft-ietf-bier-idr-extensions-03.txt, August 2017.

[BIER-PMM]
Mirsky, G., Zheng, L., Chen, M., and G. Fioccola, "IP Flow Performance Measurement Framework", internet-draft draft-ietf-bier-pmmm-oam-03, October 2017.

[BIER_MVPN]
Rosen, E., Ed., Sivakumar, M., Wijnands, IJ., Aldrin, S., Dolganow, A., and T. Przygienda, "Multicast VPN Using Bier", internet-draft draft-ietf-bier-mvpn-07, July 2017.

[BIER_PING]
Kumar, N., Pignataro, C., Akiya, N., Zheng, L., Chen, M., and G. Mirsky, "BIER Ping and Trace", internet-draft draft-ietf-bier-ping-02.txt, July 2017.

[ISIS_BIER_EXTENSIONS]
Ginsberg, L., Przygienda, T., Aldrin, S., and Z. Zhang, "BIER Support via IS-IS", internet-draft draft-ietf-bier-isis-extensions-05.txt, July 2017.

[OSPF_BIER_EXTENSIONS]
Psenak, P., Kumar, N., Wijnands, IJ., Dolganow, A., Przygienda, T., Zhang, Z., and S. Aldrin, "OSPF Extensions for Bit Index Explicit Replication", internet-draft draft-ietf-ospf-bier-extensions-07.txt, July 2017.

[RFC6790] Kompella, K., Drake, J., Amante, S., Henderickx, W., and L. Yong, "The Use of Entropy Labels in MPLS Forwarding", RFC 6790, DOI 10.17487/RFC6790, November 2012, <https://www.rfc-editor.org/info/rfc6790>.

Authors' Addresses

IJsbrand Wijnands (editor)
Cisco Systems, Inc.
De Kleetlaan 6a
Diegem 1831
Belgium

Email: ice@cisco.com

Eric C. Rosen (editor)
Juniper Networks, Inc.
10 Technology Park Drive
Westford, Massachusetts 01886
United States

Email: erosen@juniper.net

Andrew Dolganow
Nokia
438B Alexandra Rd #08-07/10
Alexandra Technopark
Singapore 119968

Email: andrew.dolganow@nokia.com

Jeff Tantsura
Individual

Email: jefftant.ietf@gmail.com

Sam K Aldrin
Google, Inc.
1600 Amphitheatre Parkway
Mountain View, California
United States

Email: aldrin.ietf@gmail.com

Israel Meilik
Broadcom

Email: israel@broadcom.com