Internet Engineering Task Force
Internet-Draft
Intended status: Standards Track
Expires: March 26, 2015
IJ. Wijnands, Ed.
E. Rosen, Ed.
Cisco Systems, Inc.
A. Dolganow
Alcatel-Lucent
T. Przygienda
Ericsson
September 22, 2014

Multicast using Bit Index Explicit Replication
draft-wijnands-bier-architecture-00

Abstract

This document specifies a new architecture for the forwarding of multicast data packets. It provides optimal forwarding of multicast packets through a "multicast domain". However, it does not require any explicit tree-building protocol, nor does it require intermediate nodes to maintain any per-flow state. This architecture is known as "Bit Index Explicit Replication" (BIER). 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. Elimination of the per-flow state and the explicit tree-building protocols results in a considerable simplification.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

This document specifies a new architecture for the forwarding of multicast data packets. It provides optimal forwarding of multicast data packets through a "multicast domain". However, it does not require any explicit tree-building protocol, and does not require intermediate nodes to maintain any per-flow state. This architecture is known as "Bit Index Explicit Replication" (BIER).

A router that supports BIER is known as a "Bit-Forwarding Router" (BFR). A BIER domain is a connected set of BFRs, where each BFR has a "BFR-id" that is unique within the domain. A BFR-id is a small unstructured number. For instance, if a particular BIER domain contains 1,374 BFRs, each one could be given a BFR-id in the range 1-1374.

A multicast data packet enters a BIER domain at a "Bit-Forwarding Ingress Router" (BFIR), and leaves the BIER domain at one or more "Bit-Forwarding Egress Routers" (BFERs). A BFR that receives a multicast data packet from another BFR in the same BIER domain, and forwards the packet to another BFR in the same BIER domain, will be known as a "transit BFR" for that packet. A single BFR may be a BFIR for some multicast traffic while also being a BFER for some multicast traffic and a transit BFR for some multicast traffic. In fact, a BFR may be the BFIR for a given packet and may also be (one of) the BFER(s), for that packet; it may also forward that packet to one or more additional BFRs.

When a multicast packet arrives from outside the domain at a BFIR, the BFIR determines the set of BFERs to which the packet must be sent. It then encapsulates the packet in a "BIER header". The BIER header contains a bit string in which each bit represents a single BFR-id. To indicate that a particular BFER needs to receive a given packet, the BFIR sets the bit corresponding to that BFER's BFR-id. We will use term "BitString" to refer to the bit string field in the BIER header. We will use the term "payload" to refer to the packet

that has been encapsulated. Thus a "BIER-encapsulated" packet consists of a "BIER header" followed by a "payload".

The number of BFERs to which a given packet can be forwarded is limited only by the length of the BitString in the BIER header. Different deployments can use different BitString lengths. We will use the term "BitStringLength" to refer to the number of bits in the BitString. It is possible that some deployment will have more BFERs in a given domain than there are bits in the BitString. To accommodate this case, the BIER encapsulation includes both the BitString and a "Set Identifier" (SI). It is the BitString and the

SI together that determine the set of BFERs to which a given packet will be delivered:

- o by convention, the least significant (rightmost) bit in the BitString is "bit 1", and the most significant (leftmost) bit is "bit BitStringLength".

- o if a BIER-encapsulated packet has an SI of n, and a BitString with bit k set, then the packet must be delivered to the BFER whose BFR-id is n*BitStringLength+k.

For example, suppose the BIER encapsulation uses a BitStringLength of 256 bits. By convention, the least significant (rightmost) bit is "bit 1", and the most significant (leftmost) bit is "bit 256". Suppose that a given packet needs to be delivered to three BFERs, where those BFERs have BFR-ids of 13, 126, and 235 respectively. The BFIR would create a BIER encapsulation with the SI set to zero, and with bits 13, 126, and 235 of the BitString set. (All other bits of the BitString would be clear.) If the packet also needs to be sent to a BFER whose BFR-id is 257, the BFIR would have to create a second copy of the packet, and the BIER encapsulation would specify an SI of 1, and a BitString with bit 1 set and all the other bits clear.

Note that it is generally advantageous to assign the BFR-ids so that as many BFERs as possible can be represented in a single bit string.

Suppose a BFR, call it BFR-A, receives a packet whose BIER encapsulation specifies an SI of 0, and a BitString with bits 13, 26, and 235 set. Suppose BFR-A has two BFR neighbors, BFR-B and BFR-C,

such that the best path to BFERs 13 and 26 is via BFR-B, but the best path to BFER 235 is via BFR-C. Then BFR-A will replicate the packet, sending one copy to BFR-B and one copy to BFR-C. However, BFR-A will clear bit 235 in the BitString of the packet copy it sends to BFR-B, and will clear bits 13 and 26 in the BitString of the packet copy it sends to BFR-C. As a result, BFR-B will forward the packet only towards BFERs 13 and 26, and BFR-C will forward the packet only towards BFER 235. This ensures that each BFER receives only one copy of the packet.

With this forwarding procedure, a multicast data packet can follow an optimal path from its BFIR to each of its BFERs. Further, since the set of BFERs for a given packet is explicitly encoded into the BIER header, the packet is not sent to any BFER that does not need to receive it. This allows for optimal forwarding of multicast traffic. This optimal forwarding is achieved without any need for transit BFRs to maintain per-flow state, or to run a multicast tree-building protocol.

The idea of encoding the set of egress nodes into the header of a multicast packet is not new. For example, [Boivie_Feldman] proposes to encode the set of egress nodes as a set of IP addresses, and proposes mechanisms and procedures that are in some ways similar to those described in the current document. However, since BIER encodes each BFR-id as a single bit in a bit string, it can represent up to 128 BFERs in the same number of bits that it would take to carry the IPv6 address of a single BFER. Thus BIER scales to a much larger number of egress nodes per packet.

BIER does not require that each transit BFR look up the best path to each BFER that is identified in the BIER header; the number of lookups required in the forwarding path for a single packet can be limited to the number of neighboring BFRs; this can be much smaller than the number of BFERs. See Section 6 (especially Section 6.4) for details.

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. The BFR Identifier and BFR-Prefix

Each BFR MUST be assigned a "BFR-Prefix". A BFR's BFR-Prefix MUST be an IP address (either IPv4 or IPv6) of the BFR, and MUST be routable within the domain. It is RECOMMENDED that the BFR-prefix be a loopback address of the BFR. Two BFRs in the same BIER domain MUST NOT be assigned the same BFR-Prefix.

A "BFR Identifier" (BFR-id) is a number in the range [1,65535]. In general, each BFR in a given BIER domain must be assigned a unique number from this range (i.e., two BFRs in the same BIER domain MUST NOT have the same BFR-id). However, if it is known that a given BFR will never need to function as a BFER, then it is not necessary to assign a BFR-id to that BFR.

The procedure for assigning a particular BFR-id to a particular BFR is outside the scope of this document. However, it is RECOMMENDED that the BFR-ids be assigned "densely" from the numbering space, as this will result in a more efficient encoding (see Section 3). That is, if there are 256 or fewer BFERs, it is RECOMMENDED to assign all the BFR-ids from the range [1,256]. If there are more than 256 BFERs, but less than 512, it is RECOMMENDED to assign all the BFR-ids from the range [1,512], with as few "holes" as possible in the earlier range. However, in some deployments, it may be advantageous to depart from this recommendation; this is discussed further in Section 3.

3. Encoding BFR Identifiers in BitStrings

To encode a BFR-id in a BIER data packet, one must convert the BFR-id to an SI and a BitString. This conversion depends upon the parameter we are calling "BitStringLength". The conversion is done as follows. If the BFR-id is N, then

- SI is the integer part of the quotient (N-1)/BitStringLength

- The BitString has one bit position set. If the low-order bit is bit 1, and the high-order bit is bit BitStringLength, the bit position that represents BFR-id N is ((N-1) modulo BitStringLength)+1.

If several different BFR-ids all resolve to the same SI, then all

those BFR-ids can be represented in a single BitString. The BitStrings for all of those BFR-ids are combined using a bitwise logical OR operation.

Different BIER domains may use different values of BitStringLength. Within a BIER domain, all BFRs MUST use the same BitStringLength value; this value is known by provisioning. A BFR MUST support a BitStringLength value of 256. Particular encapsulation types MAY allow other BitStringLengths to be optionally supported. For example, when using the encapsulation specified in [MPLS_BIER_ENCAPS], a BFR may support any or all of the following BitStringLengths: 64, 128, 256, 512, 1024, 2048, and 4096.

A BFR MUST support SI values in the range [0,15], and MAY support SI values in the range [0,255].

It is possible to design procedures that allow BFRs within a given BIER domain to use different BitStringLength values, and to enable each BFR to discover the BitStringLength values used by all the other BFRs in the domain. However, this document presupposes that they all use the same value; procedures for the use of different values may be specified in future documents.

When a BFIR determines that a multicast data packet needs to be forwarded to a particular set of destination BFERs, it partitions that set of BFERs into subsets, where each subset contains the target BFERs whose BFR-ids all resolve to the same SI. Call these the "SI-subsets" for the packet. Each SI-subset can be represented by a single BitString. The BFIR creates a copy of the packet for each SI-subset. The BIER encapsulation is then applied to each packet. The encapsulation specifies a single SI for each packet, and contains the BitString that represents all the BFR-ids in the corresponding SI-subset.

Suppose, for example, that a BFIR determines that a given packet needs to be forwarded to three BFERs, whose BFR-ids are 27, 235, and 497. The BFIR will have to forward two copies of the packet. One copy, associated with SI=0, will have a BitString with bits 27 and 235 set. The other copy, associated with SI=1, will have a BitString with bit 241 set.

In order to minimize the number of copies that must be made of a

given multicast packet, it is RECOMMENDED that the BFR-ids be assigned "densely" (see Section 2) from the numbering space. This will minimize the number of SIs that have to be used in the domain. However, depending upon the details of a particular deployment, other assignment methods may be more advantageous. Suppose, for example, that in a certain deployment, every multicast flow is either intended for the "east coast" or for the "west coast". In such a deployment, it would be advantageous to assign BFR-ids so that all the "west coast" BFR-ids fall into the same SI-subset, and so that all the "east coast" BFR-ids fall into the same SI-subset.

When a BFR receives a BIER data packet, it will infer the SI from the encapsulation. Then the set of BFERs to which the packet needs to be forwarded can then be inferred from the SI and the BitString.

In some of the examples given later in this document, we will use a BitStringLength of 4, and will represent a BFR-id in the form "SI:xyzw", where SI is the Set Identifier of the BFR-id (assuming a BitStringLength of 4), and xyzw is a string of 4 bits. A BitStringLength of 4 is used only in the examples; we would not expect actual deployments to have such a small BitStringLength.

It is expected that there will be several different forms of BIER encapsulation. The particular encapsulation that is used in a given deployment will depend on the type of network infrastructure that is used to realize the BIER domain. Details of the BIER encapsulation will be given in companion documents. An encapsulation for use in MPLS networks is described in [MPLS_BIER_ENCAPS]

4. Layering

It is helpful to think of the BIER architecture as consisting of three layers: the "routing underlay", the "BIER layer", and the "multicast flow overlay".

4.1. The Routing Underlay

The "routing underlay" establishes "adjacencies" between pairs of BFRs, and determines one or more "best paths" from a given BFR to a given set of BFRs. Each such path is a sequence of BFRs <BFR(k),

BFR(k+1), ..., BFR(k+n)> such that BFR(k+j) is "adjacent" to

BFR(k+j+1) (for 0<=j<n).

At a given BFR, say BFR-A, for every IP address that is the address of a BFR in the BIER domain, the routing underlay will map that IP address into a set of one or more "equal cost" adjacencies. If a BIER data packet has to be forwarded by BFR-A to a given BFER, say BFER-B, the packet will follow the path from BFR-A to BFER-B that is determined by the routing underlay.

It is expected that in a typical deployment, the routing underlay will be the Interior Gateway Protocol (IGP), e.g., OSPF, that is used for unicast routing. In that case, the underlay adjacencies are just the OSPF adjacencies. A BIER data packet traveling from BFR-A to BFER-B will follow the path that OSPF has selected for unicast traffic from BFR-A to BFER-B.

If one wants to have multicast traffic from BFR-A to BFER-B travel a path that is different from the path used by the unicast traffic from A to B, one can use a different underlay. For example, if multi-topology OSPF is being used, one OSPF topology could be used for unicast traffic, and the other for multicast traffic. Alternatively, one could deploy a routing underlay that creates a Steiner tree. Then BIER could be used to forward multicast data packets along the Steiner tree, while unicast packets follow the "ordinary" OSPF best path. It is even possible to have multiple routing underlays used by BIER, as long as one can infer from a data packet's BIER encapsulation which underlay to use.

By default, the routing underlay used by BIER is the unicast IGP.

Note that specification of the protocol and procedures of the routing underlay is outside the scope of this document.

4.2. The BIER Layer

The BIER layer consists of the protocol and procedures that are used in order to transmit a multicast data packet across a BIER domain, from its BFIR to its BFERs. This includes the following components:

- o Protocols and procedures that advertise each BFR's BFR-prefix and BFR-id to all other BFRs in the same BIER domain.
- o The imposition by a BFIR of a BIER header on a multicast data packet.
- o The procedures for forwarding BIER-encapsulated packets, and for modifying the BIER header during transit.

4.3. The Multicast Flow Overlay

The "multicast flow overlay" consists of the set of protocols and procedures that enable the following set of functions.

- o When a BFIR receives a multicast data packet from outside the BIER domain, the BFIR must determine the set of BFERs for that packet. This information is provided by the multicast flow overlay.

- o When a BFER receives a BIER-encapsulated packet from inside the BIER domain, the BFER must determine how to further forward the packet. This information is provided by the multicast flow overlay.

For example, suppose the BFIR and BFERs are Provider Edge (PE) routers providing Multicast Virtual Private Network (MVPN) service. The multicast flow overlay consists of the protocols and procedures described in [RFC6513] and [RFC6514]. The MVPN signaling described in those RFCs enables an ingress PE to determine the set of egress PEs for a given multicast flow (or set of flows); it also enables an egress PE to determine the "Virtual Routing and Forwarding Tables" (VRFs) to which multicast packets from the backbone network should be sent. MVPN signaling also has several components that depend on the type of "tunneling technology" used to carry multicast data though the network. Since BIER is, in effect, a new type of "tunneling technology", some extensions to the MVPN signaling are needed in order to properly interface the multicast flow overlay with the BIER layer. These will be specified in a companion document.

MVPN is just one example of a multicast flow overlay. Protocols and procedures for other overlays will be provided in companion documents. It is also possible to implement the multicast flow overlay by means of a "Software Defined Network" (SDN) controller. Specification of the protocols and procedures of the multicast flow overlay is outside the scope of this document.

5. Advertising BFR-ids and BFR-Prefixes

As stated in Section 2, each BFER is assigned a BFR-id (for a given BIER domain). Each BFER must advertise this assignment to all the other BFRs in the domain. Similarly, each BFR is assigned a BFR-prefix (for a given BIER domain), and must advertise this assignment to all the other BFRs in the domain. Finally, it is useful for each BFR to advertise its value of BitStringLength (for a given BIER domain)

If the BIER domain is also a link state routing IGP domain (i.e., an OSPF or IS-IS domain), the advertisement of the BFR-id, BFR-prefix,

and BitStringLength can be done using the advertisement capabilities of the IGP. For example, if a BIER domain is also an OSPF domain, these advertisements can be done using the OSPF "Opaque Link State Advertisement" (Opaque LSA) mechanism. Details of the necessary extensions to OSPF and IS-IS will be provided in companion documents. (See [OSPF_BIER_EXTENSIONS] for the extensions to OSPF.)

These advertisements enable each BFR to associate a given BFR-id with a given BFR-prefix. As will be seen in subsequent sections of this document, knowledge of this association is an important part of the forwarding process.

Since each BFR needs to have a unique (in the domain) BFR-id, two different BFRs can advertise ownership of the same BFR-id only if there has been a provisioning error. If BFR-A determines that BFR-B and BFR-C have both advertised the same BFR-id, BFR-A MUST log an error. Suppose that the duplicate BFR-id is "N". When BFR-A is functioning as a BFIR, it MUST NOT encode the BFR-id value N in the BIER encapsulation of any packet, even if it has determined that the packet needs to be received by BFR-B and/or BFR-C. This will mean that BFR-B and BFR-C cannot receive multicast traffic at all until the provisioning error is fixed. However, that is preferable to having them receive each other's traffic.

6. BIER Intra-Domain Forwarding Procedures

This section specifies the rules for forwarding a BIER-encapsulated data packet within a BIER domain.

6.1. Overview

This section provides a brief overview of the BIER forwarding procedures. Subsequent sub-sections specify the procedures in more detail.

To forward a BIER-encapsulated packet:

1. Determine the packet's SI.

2. From the SI and the BitString, determine the set of destination BFERs for the packet.

3. Using information provided by the routing underlay, determine the next hop adjacency for each of the destination BFERs.

4. Partition the set of destination BFERs such that all the BFERs in a single partition have the same next hop. We will say that each partition is associated with a next hop.

5. For each partition:

   a. Make a copy of the packet.

   b. Clear any bit in the packet's BitString that identifies a BFER that is not in the partition.

   c. Transmit the packet to the associated next hop.

If a BFR receives a BIER-encapsulated packet whose SI and BitString identify that BFR itself, then the BFR is also a BFER for that packet. As a BFER, it must pass the payload to the multicast flow overlay. If the BitString has more than one bit set, the packet also needs to be forwarded further within the BIER domain. If the BF(E)R also forwards one or more copies of the packet within the BIER domain, the bit representing the BFR's own BFR-id will be cleared in all the copies.

When BIER on a BFER passes a packet to the multicast flow overlay, it may need to provide contextual information obtained from the BIER encapsulation. The information that needs to pass between the BIER layer and the multicast flow layer is specific to the multicast flow layer. Specification of the interaction between the BIER layer and the multicast flow layer is outside the scope of this specification.

When BIER on a BFER passes a packet to the multicast flow overlay, the overlay will determine how to further dispatch the packet. If the packet needs to be forwarded into another BIER domain, then the BFR will act as a BFER in one BIER domain and as a BFIR in another. A BIER-encapsulated packet cannot pass directly from one BIER domain to another; at the boundary between BIER domains, the packet must be decapsulated and passed to the multicast flow layer.

Note that when a BFR transmits multiple copies of a packet within a BIER domain, only one copy will be destined to any given BFER. Therefore it is not possible for any BIER-encapsulated packet to be delivered more than once to any BFER.

6.2. BFR Neighbors

The "BFR Neighbors" (BFR-NBRs) of a given BFR, say BFR-A, are those BFRs that, according to the routing underlay, are adjacencies of BFR-A. Each BFR-NBR will have a BFR-prefix.

Suppose a BIER-encapsulated packet arrives at BFR-A. From the packet's encapsulation, BFR-A learns the BFR-ids of the BFERs to which the packet is destined. Then using the information advertised per Section 5, BFR-A can find the BFR-prefix of each destination

BFER. Given the BFR-prefix of a particular destination BFER, say BFER-D, BFR-A learns from the routing underlay an IP address of the BFR that is the next hop on the path from BFR-A to BFER-D. Let's call this next hop BFR-NH. BFR-A must then determine the BFR-prefix of BFR-NH. (This determination can be made from the information advertised per Section 5.) This BFR-prefix is the BFR-NBR of BFR-A on the path from BFR-A to BFER-D.

Note that if the routing underlay provides multiple equal cost paths from BFR-A to BFER-D, BFR-A may have multiple BFR-NBRs for BFER-D.

6.3. The Bit Index Routing Table

The Bit Index Routing Table (BIRT) is a table that maps from the BFR-id of a BFER to the BFR-prefix of that BFER, and to the BFR-NBR on the path to that BFER.

```
  ( A ) ------------ (  B  ) ------------ ( C ) ------------ ( D )
 4 (0:1000)             \                  \           1 (0:0001)
                         \                  \
                        ( E )              ( F )
                      3 (0:0100)         2 (0:0010)
```

Figure 1: BIER Topology 1

As an example, consider the topology shown in Figure 1. In this diagram, we represent the BFR-id of each BFR in the SI:xyzw form discussed in Section 3. This topology will result in the BIRT of Figure 2 at BFR-B. The first column shows the BFR-id as a number and also (in parentheses) in the SI:BitString format that corresponds to a BitStringLength of 4. (The actual minimum BitStringLength is 64, but we use 4 in the examples.)

BFR-id (SI:BitString)	BFR-Prefix of Dest BFER	BFR-NBR
4 (0:1000)	A	A
1 (0:0001)	D	C
3 (0:0100)	E	E
2 (0:0010)	F	C

Figure 2: Bit Index Routing Table at BFR-B

6.4. The Bit Index Forwarding Table

The "Bit Index Forwarding Table" (BIFT) is derived from the BIRT as follows.

Suppose that several rows in the BIRT have the same SI and the same BFR-NBR. By taking the logical OR of the BitStrings of those rows, we obtain a bit mask that corresponds to that combination of SI and BFR-NBR. We will refer to this bit mask as the "Forwarding Bit Mask" (F-BM) for that <SI,BFR-NBR> combination.

For example, in Figure 2, we see that two of the rows have the same SI (0) and same BFR-NBR (C). The Bit Mask that corresponds to <SI=0, BFR-NBR-C> is 0011 ("0001" OR'd with "0010").

The BIFT is used to map from the BFR-id of a BFER to the corresponding F-BM and BFR-NBR. For example, Figure 3 shows the BIFT that is derived from the BIRT of Figure 2. Note that BFR-ids 1 and 2 have the same SI and the same BFR-NBR, hence they have the same F-BM.

BFR-id (SI:Bitstring)	F-BM	BFR-NBR
1 (0:0001)	0011	C
2 (0:0010)	0011	C
3 (0:0100)	0100	E
4 (0:1000)	1000	A

Figure 3: Bit Index Forwarding Table

This Bit Index Forwarding Table (BIFT) is programmed into the data-plane and used to forward packets, applying the rules specified below in Section 6.5.

6.5. The BIER Forwarding Procedure

Below is the procedure for forwarding a BIER-encapsulated packet.

1. Determine the packet's SI.

2. Find the position of least significant (rightmost) bit in the packet's BitString that is set. (Remember, bits are numbered from 1, starting with the least significant bit.) Use that bit position, together with the SI, as the 'index' into the BIFT.

3. Extract from the BIFT the F-BM and the BFR-NBR.

4. Copy the packet. Update the copy's BitString by AND'ing it with the F-BM (i.e., PacketCopy->BitString &= F-BM). Then forward the copy to the BFR-NBR. Note that when a packet is forwarded to a particular BFR-NBR, its BitString identifies only those BFERs that are to be reached via that BFR-NBR.

5. Now update the original packet's BitString by AND'ing it with the INVERSE of the F-BM (i.e., Packet->Bitstring &= ~F-BM). (This clears the bits that identify the BFERs to which a copy of the packet has just been forwarded.) Go to step 2.

Note that this procedure causes the packet to be forwarded to a particular BFR-NBR only once. The number of lookups in the BIFT is the same as the number of BFR-NBRs to which the packet must be forwarded; it is not necessary to do a separate lookup for each destination BFER.

Suppose it has been decided (by the above rules) to send a packet to a particular BFR-NBR. If that BFR-NBR is connected via multiple parallel interfaces, it may be desirable to apply some form of load balancing. Load balancing algorithms are outside the scope of this

document. However, if the packet's encapsulation contains an "entropy" field, the entropy field SHOULD be respected; two packets with the same value of the entropy field SHOULD be sent on the same interface (if possible).

In some cases, the routing underlay may provide multiple equal cost paths (through different BFR-NBRs) to a given BFER. This is known as "Equal Cost Multiple Paths" (ECMP). The procedures described in this section must be augmented in order to support load balancing over ECMP. The necessary augmentations can be found in Section 6.7.

In the event that unicast traffic to the BFR-NBR is being sent via a "bypass tunnel" of some sort, the BIER-encapsulated multicast traffic send to the BFR-NBR SHOULD also be sent via that tunnel. This allows any existing "Fast Reroute" schemes to be applied to multicast traffic as well as to unicast traffic.

Some examples of these forwarding procedures can be found in Section 6.6.

The rules given in this section can be represented by the following pseudocode:

```
void ForwardBitMaskPacket (Packet)
{
    SI=GetPacketSI(Packet);
    Offset=SI*BitStringLength;
    for (Index = GetFirstBitPosition(Packet->BitString); Index ;
         Index = GetNextBitPosition(Packet->BitString, Index)) {
        F-BM = BIFT[Index+Offset]->F-BM;
        if (!F-BM) continue;
        BFR-NBR = BIFT[Index+Offset]->BFR-NBR;
        PacketCopy = Copy(Packet);
        PacketCopy->BitString &= F-BM;
        PacketSend(PacketCopy, BFR-NBR);
        Packet->BitString &= ~F-BM;
    }
}
```

Figure 4: Pseudocode

Note that at a given BFER, the BFR-NBR entry corresponding to the BFER's own BFR-id will be the BFER's own BFR-prefix. In this case,

the "PacketSend" function sends the packet to the multicast flow layer.

6.6. Examples of BIER Forwarding

In this section, we give two examples of BIER forwarding, based on the topology in Figure 1. In these examples, all packets have SI=0, and the BitStringLength is 4. Figure 5 shows the BIFT entries for SI=0 only. For compactness, we show the first column of the BIFT, the BFR-id, only as an integer.

BFR-A BIFT

Id	F-BM	NBR
1	0111	B
2	0111	B
3	0111	B
4	1000	A

BFR-B BIFT

Id	F-BM	NBR
1	0011	C
2	0011	C
3	0100	E
4	1000	A

BFR-C BIFT

Id	F-BM	NBR
1	0001	D
2	0010	F
3	1100	B
4	1100	B

Figure 5: BIFTs for Forwarding Examples

6.6.1. Example 1

BFR-D, BFR-E and BFR-F are BFER's. BFR-A is the BFIR. Suppose that BFIR-A has learned from the multicast flow layer that BFER-D is interested in a given multicast flow. If BFIR-A receives a packet of that flow from outside the BIER domain, BFIR-A applies the BIER encapsulation to the packet. The encapsulation must be such that the SI is zero. The encapsulation also includes a BitString, with just bit 1 set and with all other bits clear (i.e., 0001). This indicates that BFER-D is the only BFER that needs to receive the packet. Then BFIR-A follows the procedures of Section 6.5:

- o Since the packet's BitString is 0001, BFIR-A finds that the first bit in the string is bit 1. Looking at entry 1 in its BIFT, BFR-A determines that the bit mask F-BM is 0111 and the BFR-NBR is BFR-B.

- o BFR-A then makes a copy of the packet, and applies F-BM to the copy: Copy->BitString &= 0111. The copy's Bitstring is now 0001 (0001 & 0111).

o The copy is now sent to BFR-B.

o BFR-A then updates the packet's BitString by applying the inverse of the F-BM: Packet->Bitstring &= ~F-BM. As a result, the packet's BitString is now 0000 (0001 & 1000).

o As the packet's BitString is now zero, the forwarding procedure is complete.

When BFR-B receives the multicast packet from BFR-A, it follows the same procedure. The result is that a copy of the packet, with a BitString of 0001, is sent to BFR-C. BFR-C applies the same procedures, and as a result sends a copy of the packet, with a BitString of 0001, to BFR-D.

At BFER-D, the BIFT entry (not pictured) for BFR-id 1 will specify an F-BM of 0000 and a BFR-NBR of BFR-D itself. This will cause a copy of the packet to be delivered to the multicast flow layer at BFR-D. The packet's BitString will be set to 0000, and the packet will not be forwarded any further.

6.6.2. Example 2

This example is similar to Example 1, except that BFIR-A has learned from the multicast flow layer that both BFER-D and BFER-E are interested in a given multicast flow. If BFIR-A receives a packet of that flow from outside the BIER domain, BFIR-A applies the BIER encapsulation to the packet. The encapsulation must be such that the SI is zero. The encapsulation also includes a BitString with two bits set: bit 1 is set (as in example 1) to indicate that BFR-D is a BFER for this packet, and bit 3 is set to indicate that BFR-E is a BFER for this packet. I.e., the BitString (assuming again a BitStringLength of 4) is 0101. To forward the packet, BFIR-A follows the procedures of Section 6.5:

o Since the packet's BitString is 0101, BFIR-A finds that the first bit in the string is bit 1. Looking at entry 1 in its BIFT, BFR-A determines that the bit mask F-BM is 0111 and the BFR-NBR is BFR-B.

o BFR-A then makes a copy of the packet, and applies the F-BM to the copy: Copy->BitString &= 0111. The copy's Bitstring is now 0101

(0101 & 0111).

o The copy is now sent to BFR-B.

o BFR-A then updates the packet's BitString by applying the inverse of the F-BM: Packet->Bitstring &= ~F-BM. As a result, the packet's BitString is now 0000 (0101 & 1000).

o As the packet's BitString is now zero, the forwarding procedure is complete.

When BFR-B receives the multicast packet from BFR-A, it follows the procedure of Section 6.5, as follows:

o Since the packet's BitString is 0101, BFR-B finds that the first bit in the string is bit 1. Looking at entry 1 in its BIFT, BFR-B determines that the bit mask F-BM is 0011 and the BFR-NBR is BFR-C.

o BFR-B then makes a copy of the packet, and applies the F-BM to the copy: Copy->BitString &= 0011. The copy's Bitstring is now 0001 (0101 & 0011).

o The copy is now sent to BFR-C.

o BFR-B then updates the packet's BitString by applying the inverse of the F-BM: Packet->Bitstring &= F-BM. As a result, the packet's BitString is now 0100 (0101 & 1100).

o Now BFR-B finds the next bit in the packet's (modified) BitString. This is bit 3. Looking at entry 3 in its BIFT, BFR-B determines that the F-BM is 0100 and the BFR-NBR is BFR-E.

o BFR-B then makes a copy of the packet, and applies the F-BM to the copy: Copy->BitString &= 0100. The copy's Bitstring is now 0100 (0100 & 0100).

o The copy is now sent to BFR-E.

o BFR-B then updates the packet's BitString by applying the inverse of the F-BM: Packet->Bitstring &= ~F-BM. As a result, the

packet's BitString is now 0000 (0100 & 1011).

- As the packet's BitString is now zero, the forwarding procedure is complete.

Thus BFR-B forwards two copies of the packet. One copy of the packet, with BitString 0001, has now been sent from BFR-B to BFR-C. Following the same procedures, BFR-C will forward the packet to BFER-D.

At BFER-D, the BIFT entry (not pictured) for BFR-id 1 will specify an F-BM of 0000 and a BFR-NBR of BFR-D itself. This will cause a copy of the packet to be delivered to the multicast flow layer at BFR-D. The packet's BitString will be set to 0000, and the packet will not be forwarded any further.

The other copy of the packet has been sent from BFR-B to BFER-E, with BitString 0100.

At BFER-E, the BIFT entry (not pictured) for BFR-id 3 will specify an F-BM of 0000 and a BFR-NBR of BFR-E itself. This will cause a copy of the packet to be delivered to the multicast flow layer at BFR-E. The packet's BitString will be set to 0000, and the packet will not be forwarded any further.

6.7. Equal Cost Multi-path Forwarding

In many networks, the routing underlay will provide multiple equal cost paths from a given BFR to a given BFER. When forwarding multicast packets through the network, it can be beneficial to take advantage of this by load balancing among those paths. This feature is known as "equal cost multiple path forwarding", or "ECMP".

BIER supports ECMP, but the procedures of Section 6.5 must be modified slightly. This is shown in Figure 6, which is based on the topology of Figure 1.

```
     BFR-A BIFT             BFR-B BIFT             BFR-C BIFT
 --------------------   --------------------   --------------------
 | Id | F-BM | NBR |    | Id | F-BM | NBR |    | Id | F-BM | NBR |
 ====================   ====================   ====================
 | 1  | 0111 |  B  |    | 1  | 0011 |  C  |    | 1  | 0001 |  D  |
```

```
   --------------------   --------------------   --------------------
   | 2  | 0111 |  B  |   | 2  | 0011 |  C  |   | 2  | 0010 |  F  |
   --------------------   |    | 0110 |  E  |   --------------------
   | 3  | 0111 |  B  |   --------------------   | 3  | 1100 |  B  |
   --------------------   | 3  | 0110 |  E  |   --------------------
   | 4  | 1000 |  A  |   -------------------|   | 4  | 1100 |  B  |
   --------------------   | 4  | 1000 |  A  |   --------------------
                          --------------------
```

Figure 6: Example of ECMP

In this example, BFR-B has two equal cost paths to reach BFER-F, one via BFR-C and one via BFR-E. Since the BFR-id of BFER-F is 2, this is reflected in entry 2 of BFR-B's BIFT. Entry 2 shows that BFR-B has a choice of two BFR-NBRs for BFER-B, and that a different F-BM is associated with each choice. When BFR-B looks up entry 2 in the BIFT, it can choose either BFR-NBR. However, when following the procedures of Section 6.5, it MUST use the F-BM corresponding to the BFR-NBR that it chooses.

How the choice is made is an implementation matter. However, the usual rules for ECMP apply: packets of a given flow SHOULD NOT be

split among two paths, and any "entropy" field in the packet's encapsulation SHOULD be respected.

Note however that by the rules of Section 6.5, any packet destined for both BFER-D and BFER-F will be sent via BFR-C.

6.8. Prevention of Loops and Duplicates

The BitString in a BIER-encapsulated packet specifies the set of BFERs to which that packet is to be forwarded. When a BIER-encapsulated packet is replicated, no two copies of the packet will ever have a BFER in common. If one of the packet's BFERs forwards the packet further, that will first clear the bit that identifies itself. As a result, duplicate delivery of packets is not possible with BIER.

As long as the routing underlay provides a loop free path between each pair of BFRs, BIER-encapsulated packets will not loop. Since the BIER layer does not create any paths of its own, there is no need

for any BIER-specific loop prevention techniques beyond the forwarding procedures specified in Section 6.5.

If, at some time, the routing underlay is not providing a loop free path between BFIR-A and BFER-B, then BIER encapsulated packets may loop while traveling from BFIR-A to BFER-B. However, such loops will never result in delivery of duplicate packets to BFER-B.

These properties of BIER eliminate the need for the "reverse path forwarding" (RPF) check that is used in conventional IP multicast forwarding.

7. IANA Considerations

This document contains no actions for IANA.

8. Security Considerations

When BIER is paired with a particular multicast flow layer, it inherits the security considerations of that layer. Similarly, when BIER is paired with a particular routing underlay, it inherits the security considerations of that layer.

If the BIER encapsulation of a particular packet specifies an SI or a BitString other than the one intended by the BFIR, the packet is likely to be misdelivered. If the BIER encapsulation of a packet is modified (through error or malfeasance) in a way other than that specified in this document, the packet may be misdelivered.

If the procedures used for advertising BFR-ids and BFR-prefixes are not secure, an attack on those procedures may result in incorrect delivery of BIER-encapsulated packets.

Every BFR must be provisioned to know which of its interfaces lead to a BIER domain and which do not. If two interfaces lead to different BIER domains, the BFR must be provisioned to know that those two interfaces lead to different BIER domains. If the provisioning is not correct, BIER-encapsulated packets from one BIER domain may "leak" into another; this is likely to result in misdelivery of packets.

9. Acknowledgements

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

10. Contributor Addresses

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

Wim Henderickx
Alcatel-Lucent
Copernicuslaan 50
Antwerp 2018
Belgium

Email: wim.henderickx@alcatel-lucent.com

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

Email: Martin.Horneffer@telekom.de

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

Email: Uwe.Joorde@telekom.de

Jeff Tantsura
Ericsson
300 Holger Way
San Jose, CA 95134
US

Email: jeff.tantsura@ericsson.com

11. References

11.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

11.2. Informative References

[Boivie_Feldman]
Boivie, R. and N. Feldman, "Small Group Multicast", (expired) draft-boivie-sgm-02.txt, February 2001.

[MPLS_BIER_ENCAPS]
Wijnands, IJ., "BIER Encapsulation for MPLS Networks", internet-draft draft-wijnands-mpls-bier-encaps-00.txt, September 2014.

[OSPF_BIER_EXTENSIONS]
Kumar, N. and P. Psenak, "OSPF Extension for Bit Index Explicit Replication", internet-draft draft-kumar-ospf-bier-extension-00.txt, September 2014.

[RFC6513] Rosen, E. and R. Aggarwal, "Multicast in MPLS/BGP IP VPNs", RFC 6513, February 2012.

[RFC6514] Aggarwal, R., Rosen, E., Morin, T., and Y. Rekhter, "BGP Encodings and Procedures for Multicast in MPLS/BGP IP VPNs", RFC 6514, February 2012.

Authors' Addresses

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

Email: ice@cisco.com

Eric C. Rosen (editor)
Cisco Systems, Inc.
1414 Massachusetts Avenue
Boxborough, Massachusetts 01781
USA

Email: erosen@cisco.com

Andrew Dolganow
Alcatel-Lucent
600 March Rd.
Ottawa, Ontario K2K 2E6
Canada

Email: andrew.dolganow@alcatel-lucent.com

Tony Przygienda
Ericsson

Email: antoni.przygienda@ericsson.com