Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: September 6, 2015

T. Eckert
Cisco

March 5, 2015

Traffic Enginering for Bit Index Explicit Replication BIER-TE
draft-eckert-bier-te-arch-00

Abstract

This document proposes an architecture for BIER-TE: Traffic Engineering for Bit Index Explicit Replication (BIER).

BIER-TE shares part of its architecture with BIER as described in [I-D.wijnands-bier-architecture]. It also proposes to share the packet format with BIER.

BIER-TE forwards and replicates packets like BIER based on a BitString in the packet header but it does not require an IGP. It does support traffic engineering by explicit hop-by-hop forwarding and loose hop forwarding of packets. It does support Fast ReRoute (FRR) for link and node protection and incremental deployment. Because BIER-TE like BIER operates without explicit in-network tree-building but also supports traffic engineering, it is more similar to SR than RSVP-TE.

Table of Contents

1. Introduction

1.1. Overview

This document specifies the architecture for BIER-TE: traffic engineering for Bit Index Explicit Replication BIER.

BIER-TE shares architecture and packet formats with BIER as described in [I-D.wijnands-bier-architecture].

BIER-TE forwards and replicates packets like BIER based on a BitString in the packet header but it does not require an IGP. It does support traffic engineering by explicit hop-by-hop forwarding and loose hop forwarding of packets. It does support Fast ReRoute (FRR) for link and node protection and incremental deployment. Because BIER-TE like BIER operates without explicit in-network tree-building but also supports traffic engineering, it is more similar to SR than RSVP-TE.

The key differences over BIER are:

- BIER-TE replaces in-network autonomous path calculation by explicit paths calculated offpath by the BIER-TE controller host.

- In BIER-TE every BitPosition of the BitString of a BIER-TE packet indicates one or more adjacencies - instead of a BFER as in BIER.

- BIER-TE in each BFR has no routing table but only a BIER-TE Forwarding Table (BIFT) indexed by BitPosition and populated with only those adjacencies to which the BFR should replicate packets to.

Currently, BIER-TE does not support BIER-sub-domains and it does not not use BFR-id or "Set Identifiers" (SI) in BIER-TE headers that share the same format as BIER headers.

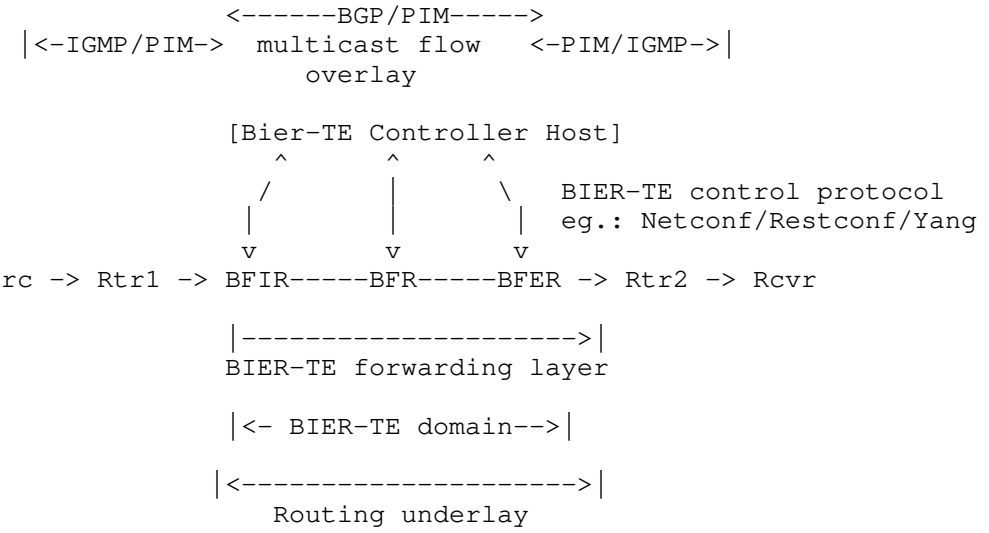
1.2. Requirements Language

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 RFC 2119 [RFC2119].

2. Layering

End to end BIER-TE operations consists of four components: The "Multicast Flow Overlay", the "BIER-TE Controller Host", the "Routing Underlay" and the "BIER-TE forwarding layer".

Picture 2: Layers of BIER-TE



2.1. The Multicast Flow Overlay

The Multicast Flow Overlay operates as in BIER. See [I-D.wijnands-bier-architecture]. Instead of interacting with the BIER layer, it interacts with the BIER-TE Controller Host

2.2. The BIER-TE Controller Host

The BIER-TE controller host is an offpath central host. It communicates via protocols such as Netconf/Restconf/Yang with BFRs. The protocols used between BFRs and the controller are outside the

scope of this document. This document is only concerned about the logic how a controller can assign BitPositions to the topology and BitStrings to BIER-TE packets:

During bring-up or modifications of the network topology, the controller needs to talk to all BFRs to assign BitPositions to adjacencies of the network topology. During day-to-day operations of the network it only needs to talks to BFIRs to install BitStrings for multicast flows.

These two tasks have the following steps:

2.2.1. Assignment of BitPositions to adjacencies of the network topology

The BIER-TE controller host tracks the BFR topology of the BIER-TE domain. It determines what adjacencies require BitPositions so that BIER-TE explicit paths can be built through them as desired by operator policy.

The controller then pushes the BitPositions/adjacencies to the BIFT of the BFRs, populating only those BitPositions to the BIFT of each BFR to which that BFR should be able to send packets to - adjacencies connecting to this BFR.

2.2.2. Changes in the network topology

If the network topology changes (not failure based) so that adjacencies that are assigned to BitPositions are no longer needed, the controller can re-use those BitPositions for new adjacencies. First, these BitPositions need to be removed from any BFIR flow state and BFR BIFT state (and BTAFT if FRR is supported, see below), then they can be repopulated, first into BIFT (and if FRR is supported BTAFT), then into BFIR.

2.2.3. Set up per-multicast flow BIER-TE state

The BIER-TE controller host tracks the multicast flow overlay to determine what multicast flow needs to be sent by a BFIR to which set of BFER. It calculates the desired distribution tree across the BIER-TE domain based on algorithms outside the scope of this document (eg.: CSFP, Steiner Tree,...). It then pushes the calculated BitString into the BFIR.

2.2.4. Link/Node Failures and Recovery

When link or nodes fail or recover in the topology, BIER-TE can quickly respond with the optional FRR procedures described below. It can also more slowly react by recalculating the BitStrings of affected multicast flows. This reaction is slower than the FR procedure because the controller needs to receive link/node up/down indications, recalculate the desired BitStrings and push them down into the BFIRs. with FRR, this is all performed locally on a BFR receiving the adjacency up/down notification.

2.3. The BIER-TE Forwarding Layer

When the BIER-TE Forwarding Layer receives a packet, it simply looks up the BitPositions that are set in the BitString of the packet in the Bit Index Forwarding Table (BIFT) that was populated by the BIER-TE controller host. For every BP that is set in the BitString, and that has one or more adjacencies in the BIFT, a copy is made according to the type of adjacencies for that BP in the BIFT. Before sending any copy, the BFR resets all BitPositions in the BitString of the packet to which it can create a copy. This is done to inhibit that packets can loop.

If the BFR support BIER-TE FRR operations, then the BIER-TE forwarding layer will receive fast adjacency up/down notification uses the BIER-TE FRR Adjacency Table to modify the BitString of the packet before it performs BIER-TE forwarding. This is detailed in the FRR section.

2.4. The Routing Underlay

BIER-TE is sending BIER packets to directly connected BIER-TE neighbors as L2 (unicasted) BIER packets without requiring a routing underlay. BIER-TE forwarding uses the Routing underlay for forward_routed adjacencies which copy BIER-TE packets to not-directly-connected BFRs (see below for adjacency definitions).

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

3. BIER-TE Forwarding

3.1. The Bit Index Forwarding Table (BIFT)

The Bit Index Forwarding Table (BIFT) exists in every BFR. It is a table indexed by BitPosition and is populated by the BIER-TE control plane. Each index can be empty or contain a list of one or more adjacencies.

Index	Adjacencies
1	forward_connected(interface,neighbor,DNR)
2	forward_connected(interface,neighbor,DNR) forward_connected(interface,neighbor,DNR)
3	local_decap([VRF])
4	forward_routed([VRF,]l3-neighbor)
5	<empty>
6	ECMP({adjacency1,...adjacencyN}, seed)
...	
BitStringLength	...

Bit Index Forwarding Table

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

Adjacencies for the same BP when populated in more than one BFR by the controller do not have to have the same adjacencies. This is up to the controller. BPs for p2p links are one case (see below).

3.2. Adjacency Types

3.2.1. Forward Connected

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

Packets sent to an adjacency with "DoNotReset" (DNR) set in the BIFT will not have the BitPosition for that adjacency reset when the BFR

creates a copy for it. The BitPosition will still be reset for copies of the packet made towards other adjacencies. The can be used for example in ring topologies as explained below.

3.2.2. Forward Routed

A "forward_routed" adjacency is an adjacency towards a BFR that is not a forward_connected adjacency: towards a loopback address of a BFR or towards an interface address that is non-directly connected. Forward_routed packets are forwarded via the Routing Underlay.

If the Routing Underlay has multiple paths for a forward_routed adjacency, it will perform ECMP independent of BIER-TE for packets forwarded across a forward_routed adjacency.

If the Routing Underlay has FRR, it will perform FRR independent of BIER-TE for packets forwarded across a forward_routed adjacency.

3.2.3. ECMP

An "Equal Cost Multipath" (ECMP) adjacency has a list of two or more adjacencies included in it. It copies the BIER-TE to one of those adjacencies based on the ECMP hash calculation. The BIER-TE ECMP hash algorithm must select the same adjacency from that list for all packets with the same "entropy" value in the BIER-TE header if the same number of adjacencies and same seed are given as parameters. Further use of the seed parameter is explained below.

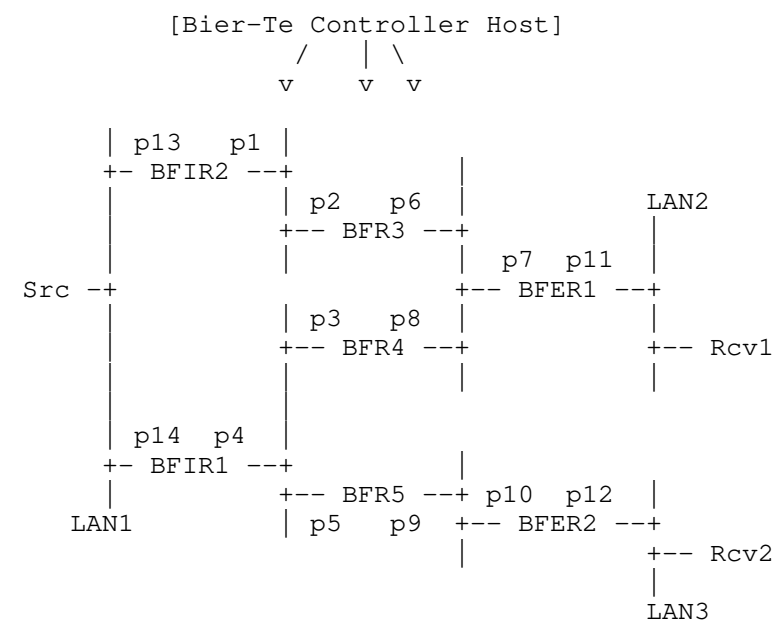
3.2.4. Local Decap

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

3.3. Basic BIER-TE Forwarding Example

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

Picture 1: Forwarding Example



pXX indicate the BitPositions number assigned by the BIER-TE controller host to adjacencies in the BIER-TE topology. For example, p9 is the adjacency towards BFR9 on the LAN connecting to BFER2.

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

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

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

...and so on.

Traffic needs to flow from BFIR2 towards Rcv1, Rcv2. The controller determines it wants it to pass across the following paths:

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

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

This BitString is set up in BFIR2. Multicast packets arriving at BFIR2 from Src are assigned this BitString.

BFIR2 forwards based on that BitString. It has p2 and p13 populated. Only p13 is in BitString which has an adjacency towards BFR3. BFIR2 resets p2 in BitString and sends a copy towards BFR2.

BFR3 sees a BitString of p5,p7,p8,p10,p11,p12. It is only interested in p1,p7,p8. It creates a copy of the packet to BFER1 (due to p7) and one to BFR4 (due to p8). It resets p7, p8 before sending.

BFER1 sees a BitString of p5,p10,p11,p12. It is only interested in p6,p7,p8,p11 and therefore considers only p11. p11 is a "local_decap" adjacency installed by the BIER-TE controller host because BFER1 should pass packets to IP multicast. The local_decap adjacency instructs BFER1 to create a copy, decapsulate it from the BIER header and pass it on to the NextProtocol, in this example IP multicast. IP multicast will then forward the packet out to LAN2 because it did receive PIM or IGMP joins on LAN2 for the traffic.

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

4. BIER-TE Controller Host BitPosition Assignments

This section describes how the BIER-TE controller host can use the different BIER-TE adjacency types to define the BitPositions of a BIER-TE domain.

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

4.1. P2P Links

Each P2p link in the BIER-TE domain is assigned one unique BitPosition with a forward_connected adjacency pointing to the neighbor on the p2p link.

4.2. BFER

Every BFER is given a unique BitPosition with a local_decap adjacency.

4.3. Leaf BFIRs

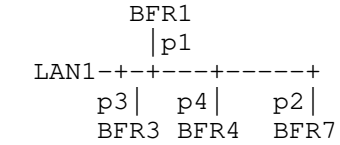
Leaf BFIRs are BFIRs where incoming BIER-TE packets never need to be forwarded to another BFR but are only sent to the BFIR to exit the BIER-TE domain. For example, in networks where PEs are spokes connected to P routers, those PEs are Leaf BFIRs unless there is a U-turn between two PEs.

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

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

4.4. LANs

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



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

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

4.5. Hub and Spoke

In a setup with a hub and multiple spokes connected via separate p2p links to the hub, all p2p links can share the same BitPosition. The BitPosition on the hubs BIFT is set up with a list of forward_connected adjacencies, one for each Spoke.

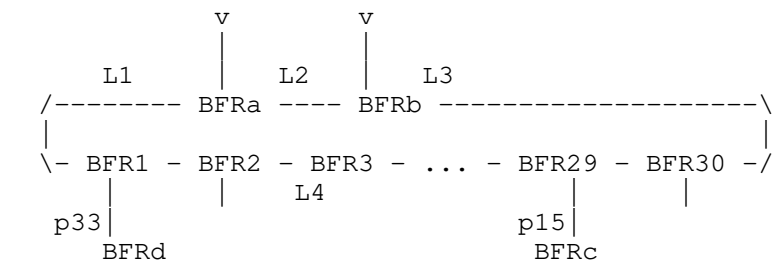
This option is similar to the BitPosition optimization in LANs: Redundantly connected spokes need their own BitPositions.

4.6. Rings

In L3 rings, instead of assigning a single BitPosition for every p2p link in the ring, it is possible to save BitPositions by setting the "Do Not Reset" (DNR) flag on forward_connected adjacencies.

For the rings shown in the following picture, a single BitPosition will suffice to forward traffic entering the ring at BFRa or BFRb all the way up to BFR1:

On BFRa, BFRb, BFR30,... BFR3, the BitPosition is populated with a forward_connected adjacency pointing to the clockwise neighbor on the ring and with DNR set. On BFR2, the adjacency also points to the clockwise neighbor BFR1, but without DNR set. Handling DNR this way ensures that copies forwarded from any BFR in the ring to a BFR outside the ring will not have this BitPosition, therefore minimizing the chance to create loops.



4.7. Equal Cost MultiPath (ECMP)

The ECMP adjacency allows to use just one BP per link bundle between two BFRs instead of one BP for each p2p member link of that link bundle. In the following picture, one BP is used across L1,L2,L3 and BFR1/BFR2 have for the BP

```
          --L1-----
    BFR1 --L2----- BFR2
          --L3-----
```

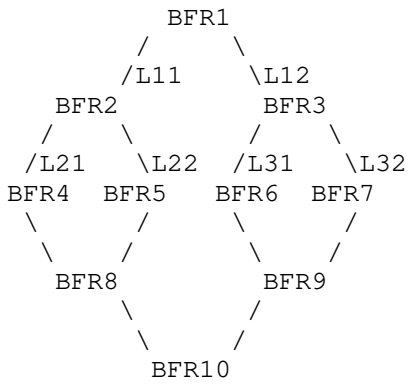
BIFT entry in BFR1:

Index	Adjacencies
6	ECMP({L1-to-BFR2,L2-to-BFR2,L3-to-BFR2}, seed)

BIFT entry in BFR2:

Index	Adjacencies
6	ECMP({L1-to-BFR1,L2-to-BFR1,L3-to-BFR1}, seed)

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



BIFT entry in BFR1:

6	ECMP({L11-to-BFR2,L12-to-BFR3}, seed)

BIFT entry in BFR2:

6	ECMP({L21-to-BFR4,L22-to-BFR5}, seed)

BIFT entry in BFR3:

6	ECMP({L31-to-BFR6,L32-to-BFR7}, seed)

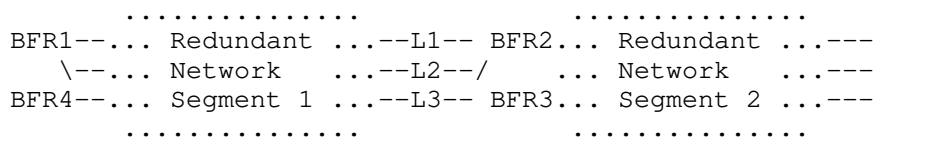
With the setup of ECMP in above topology, traffic would not be equally load-split. Instead, links L22 and L31 would see no traffic at all: BFR2 will only see traffic from BFR1 for which the ECMP hash in BFR1 selected the first adjacency in a list of 2 adjacencies: link L11-to-BFR2. When forwarding in BFR2 performs again an ECMP with two adjacencies on that subset of traffic, then it will again select the first of its two adjacencies to it: L21-to-BFR4. And therefore L22 and BFR5 sees no traffic.

To resolve this issue, the ECMP adjaceny on BFR1 simply needs to be set up with a different seed than the ECMP adjacncies on BFR2/BFR3

This issue is called polarization. It depends on the ECMP hash. It is possible to build ECMP that does not have polarization, for example by taking entropy from the actual adjacency members into account, but that can make it harder to achieve evenly balanced load-splitting on all BFR without making the ECMP hash algorithm potentially too complex for fast forwarding in the BFRs.

4.8. Routed adjacencies

Routed adjacencies can reduce the number of BitPositions required when the traffic engineering requirement is not hop-by-hop explicit path selection, but loose-hop selection.



Assume he requirement in above network is to explicitly engineer paths such that specific traffic flows are passed from segment 1 to segment 2 via link L1 (or via L2 or via L3).

To achieve this, BFR1 and BFR4 are set up with a forward_routed adjacency BitPosition towards an address of BFR2 on link L1 (or link L2 BFR3 via L3).

For paths to be engineered through a specific node BFR2 (or BFR3), BFR1 and BFR4 are set up up with a forward_routed adjacency BitPosition towards a loopback address of BFR2 (or BFR3).

4.8.1. Supporting nodes without BIER-TE

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

5. Avoiding loops and duplicates

5.1. Loops

Whenever BIER-TE creates a copy of a packet, the BitString of that copy will have all BitPositions cleared that are associated with adjacencies in the BFR. This inhibits looping of packets. The only exception are adjacencies with DNR set.

With DNR set, looping can happen. Consider in the ring picture that link L4 from BFR3 is plugged into the L1 interface of BFRa. This creates a loop where the rings clockwise BitPosition is never reset for copies of the packets traveling clockwise around the ring.

To inhibit looping in the face of such physical misconfiguration, only forward_connected adjacencies are permitted to have DNR set, and

the link layer destination address of the adjacency (eg.: MAC address) protects against closing the loop. Link layers without port unique link layer addresses should not used with the DNR flag set.

5.2. Duplicates

Duplicates happen when the topology of the BitString is not a tree but redundantly connecting BFRs with each other. The controller must therefore ensure to only create BitStrings that are trees in the topology.

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

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

6. FRR

FRR is an optional procedure. To leverage it, the BIER-TE controller host and BFRs need to support it. It does not have to be supported on all BFRs, but only those that are attached to a link/adjacency for which FRR support is required.

If BIER-TE FRR is supported by the BIER-TE controller host, then it needs to calculate the desired backup paths for link and/or node failures in the BIER-TE domain and download this information into the BIER-TE Adjacency FRR Table (BTAFT) of the BFRs. The BTAFT then drives FRR operations in the BIER-TE forwarding plane of that BFR.

6.1. The BIER-TE Adjacency FRR Table (BTAFT)

The BIER-TE IF FRR Table exists in every BFR that is supporting BIER-TE FRR procedures. It is indexed by FRR Adjacency Index. Associated with each FRR Adjacency Index is a ResetBitmask, AddBitmask and BitPosition.

FRR Adjacency Index	BitPosition	ResetBitmask	AddBitmask
1	5	..0010000	..11000000

...

An FRR Adjacency is an adjacency that is used in the BIFT of the BFR. The BFR has to be able to determine whether the adjacency is up or down in less than 50msec. An FRR adjacency can be a forward_connected adjacency with fast L2 link state Up/Down state notifications or a forward_connected or forward_routed adjacency with a fast aliveness mechanism such as BFD. Details of those mechanism are outside the scope of this architecture.

The FRR Adjacency Index is the index that would be indicated on the fast Up/Down notifications to the BIER-TE forwarding plane

The BitPosition is the BP in the BIFT in which the FRR Adjacency is used

6.2. FRR in BIER-TE forwarding

The BIER-TE forwarding plane receives fast Up/Down notifications with the FRR Adjacency Index. From the BitPosition in the BTAFT entry, it remembers which BPs are currently affected (have a down adjacency).

When a packet is received, BIER-TE forwarding checks if it has affected BPs to which it would forward. If it does, it will remove the ResetBitmask bits from the packets BitString and add the AddBitmask bits to the packets BitString.

Afterwards, normal BIER-TE forwarding occurs, taking the modified BitString into account.

6.3. FRR in the BIER-TE Controller Host

The basic rules how the BIER-TE controller host would calculate ResetBitMask and AddBitmask are as follows:

1. The BIER-TE controller host has to determine whether a failure of the adjacency should be taken to indicate link or node failure. This is a policy decision.

2. The ResetBitmask has the BitPosition of the failed adjacency.

3. In the case of link protection, the AddBitmask are the segments forming a path from the BFR over to the BFR on the other end of the failed link.

4. In the case of node protection, the AddBitmask are the segments forming a tree from the BFR over to all necessary BFR downstream of the (assumed to be failed) BFR across the failed adjacency.

5. The ResetBitmask is extended with those segments that could lead to duplicate packets if the AddBitmask is added to possible BitStrings of packets using the failing BitPosition.

6.4. BIER-TE FRR Benefits

Compared to other FRR solutions, such as RSVP-TE/P2MP FRR, BIER-TE FRR has two key distinctions

- It maintains the goal of BIER-TE not to establish in-network per multicast traffic flow state. For that reason, the backup path/ trees are only tied to the topology but not to individual distribution trees.

- For the case of node failure, it allows to build a path engineered backup tree (4.) as opposed to only a set of p2p backup tunnels.

7. BIER-TE Forwarding Pseudocode

The following sections of Pseudocode are meant to illustrate the BIER-TE forwarding plane. This code is not meant to be normative but to serve both as a potentially easier to read and more precise representation of the forwarding functionality and to illustrate how simple BIER-TE forwarding is and that it can be efficiently be implemented.

The following procedure is executed on a BFR whenever the BIFT is changed by the BIER-TE controller host:

```
global MyBitsOfInterest

void BIFTChanged()
{

    for (Index = 0; Index++ ; Index <= BitStringLength)
        if(BIFT[Index] != <empty>)
            MyBitsOfInterest != 2<<(Index-1)
}
```

The following procedure is executed whenever an adjacency used for BIER-TE FRR changes state:

```
global ResetBitMaskByBT[BitStringLength]
global AddtBitMaskByBT[BitStringLength]
global FRRaffectedBP

void FrrUpDown(FrrAdjacencyIndex, UpDown)
{
    global FRRAdjacenciesDown
    local Idx = FrrAdjacencyIndex

    if (UpDown == Up)
        FRRAdjacenciesDown &= ~ 2<<(FrrAdjacencyIndex-1)
    else
        FRRAdjacenciesDown |=   2<<(FrrAdjacencyIndex-1)

    for (Index = GetFirstBitPosition(FRRAdjacenciesDown); Index ;
        Index = GetNextBitPosition(FRRAdjacenciesDown, Index))

        local BP = BTAFT[Index].BitPosition
        FRRaffectedBP |= 2<<(Index)
        ResetBitMaskByBT[BP] |= BTAFT[Index].ResetBitMask
        AddBitMaskByBT[BP]   |= BTAFT[Index].AddBitMask
}
```

The following procedure is executed whenever a BIER-TE packet is to be forwarded:

```
void ForwardBierTePacket (Packet)
{
    // We calculate in BitMask the subset of BPs of the BitString
    // for which we have adjacencies. This is purely an
    // optimization to avoid to replicate for every BP
    // set in BitString only to discover that for most of them,
    // the BIFT has no adjacency.

    local BitMask = Packet->BitString
    Packet->BitString &= ~MyBitsOfInterest
    BitMask &= MyBitsOfInterest

    // FRR Operations
    // Note: this algorithm is not optimal yet for ECMP cases
    // it performs FRR replacement for all candidate ECMP paths

    local MyFRRBP = BitMask & FRRaffectedBP
    for (BP = GetFirstBitPosition(MyFRRNP); BP ;
         BP = GetNextBitPosition(MyFRRNP, BP))
        BitMask &= ~ResetBitMaskByBT[BP]
        BitMask |=  ResetBitMaskByBT[BT]

    // Replication
    for (Index = GetFirstBitPosition(BitMask); Index ;
         Index = GetNextBitPosition(BitMask, Index))
        foreach adjacency BIFT[Index]

            if(adjacency == ECMP(ListOfAdjacencies, seed) )
                I = ECMP_hash(sizeof(ListOfAdjacencies),
                              Packet->Entropy, seed)
                adjacency = ListOfAdjacencies[I]

            PacketCopy = Copy(Packet)

            switch(adjacency)
                case forward_connected(interface,neighbor,DNR):
                    if(DNR)
                        PacketCopy->BitString |= 2<<(Index-1)
                    SendToL2Unicast(PacketCopy,interface,neighbor)

                case forward_routed([VRF],neighbor):
                    SendToL3(PacketCopy,[VRF,]l3-neighbor)

                case local_decap([VRF],neighbor):
                    DecapBierHeader(PacketCopy)
                    PassTo(PacketCopy,[VRF,]Packet->NextProto)
}
```

8. Security Considerations

The security considerations are the same as for BIER with the following differences:

BFR-ids and BFR-prefixes are not used in BIER-TE, nor are procedures for their distribution, so these are not attack vectors against BIER-TE.

9. IANA Considerations

This document requests no action by IANA.

10. Acknowledgements

The author would like to thank Ijsbrand Wijnands and Neale Ranns for their extensive review and suggestions.

11. Change log [RFC Editor: Please remove]

00: Initial version.

12. References

[I-D.wijnands-bier-architecture]
Wijnands, I., Rosen, E., Dolganow, A., Przygienda, T., and S. Aldrin, "Multicast using Bit Index Explicit Replication", draft-wijnands-bier-architecture-04 (work in progress), February 2015.

[I-D.wijnands-mpls-bier-encapsulation]
Wijnands, I., Rosen, E., Dolganow, A., Tantsura, J., and S. Aldrin, "Encapsulation for Bit Index Explicit Replication in MPLS Networks", draft-wijnands-mpls-bier-encapsulation-02 (work in progress), December 2014.

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

Author's Address

Toerless Eckert
Cisco

Email: eckert@cisco.com

Network Working Group
Internet-Draft
Intended status: Informational
Expires: August 13, 2015

N. Kumar
R. Asati
Cisco
M. Chen
X. Xu
Huawei
A. Dolganow
Alcatel-Lucent
T. Przygienda
Ericsson
A. Gulko
Thomson Reuters
D. Robinson
id3as-company Ltd

February 9, 2015

BIER Use Cases
draft-kumar-bier-use-cases-02.txt

Abstract

Bit Index Explicit Replication (BIER) is an architecture that provides optimal multicast forwarding through a "BIER domain" without requiring intermediate routers to maintain any multicast related per-flow state. BIER also does not require any explicit tree-building protocol for its operation. 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). The BFIR router adds a BIER header to the packet. The BIER header contains a bit-string in which each bit represents exactly one BFER to forward the packet to. The set of BFERs to which the multicast packet needs to be forwarded is expressed by setting the bits that correspond to those routers in the BIER header.

This document describes some of the use-cases for BIER.

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 http://datatracker.ietf.org/drafts/current/.

Table of Contents

1. Introduction

Bit Index Explicit Replication (BIER) [I-D.wijnands-bier-architecture] is an architecture that provides optimal multicast forwarding through a "BIER domain" without requiring intermediate routers to maintain any multicast related per-

flow state. BIER also does not require any explicit tree-building protocol for its operation. 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). The BFIR router adds a BIER header to the packet. The BIER header contains a bit-string in which each bit represents exactly one BFER to forward the packet to. The set of BFERs to which the multicast packet needs to be forwarded is expressed by setting the bits that correspond to those routers in the BIER header.

The obvious advantage of BIER is that there is no per flow multicast state in the core of the network and there is no tree building protocol that sets up tree on demand based on users joining a multicast flow. In that sense, BIER is potentially applicable to many services where Multicast is used and not limited to the examples described in this draft. In this document we are describing a few use-cases where BIER could provide benefit over using existing mechanisms.

2. Specification of Requirements

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

3. BIER Use Cases

3.1. Multicast in L3VPN Networks

The Multicast L3VPN architecture [RFC6513] describes many different profiles in order to transport L3 Multicast across a providers network. Each profile has its own different tradeoffs (see section 2.1 [RFC6513]). When using "Multidirectional Inclusive" "Provider Multicast Service Interface" (MI-PMSI) an efficient tree is build per VPN, but causes flooding of egress PE's that are part of the VPN, but have not joined a particular C-multicast flow. This problem can be solved with the "Selective" PMSI to build a special tree for only those PE's that have joined the C-multicast flow for that specific VPN. The more S-PMSI's, the less bandwidth is wasted due to flooding, but causes more state to be created in the providers network. This is a typical problem network operators are faced with by finding the right balance between the amount of state carried in the network and how much flooding (waste of bandwidth) is acceptable. Some of the complexity with L3VPN's comes due to providing different profiles to accommodate these trade-offs.

With BIER there is no trade-off between State and Flooding. Since the receiver information is explicitly carried within the packet,

there is no need to build S-PMSI's to deliver multicast to a sub-set of the VPN egress PE's. Due to that behaviour, there is no need for S-PMSI's.

Mi-PMSI's and S-PMSI's are also used to provide the VPN context to the Egress PE router that receives the multicast packet. Also, in some MVPN profiles it is also required to know which Ingress PE forwarded the packet. Based on the PMSI the packet is received from, the target VPN is determined. This also means there is a requirement to have a least a PMSI per VPN or per VPN/Ingress PE. This means the amount of state created in the network is proportional to the VPN and ingress PE's. Creating PMSI state per VPN can be prevented by applying the procedures as documented in [RFC5331]. This however has not been very much adopted/implemented due to the excessive flooding it would cause to Egress PE's since *all* VPN multicast packets are forwarded to *all* PE's that have one or more VPN's attached to it.

With BIER, the destination PE's are identified in the multicast packet, so there is no flooding concern when implementing [RFC5331]. For that reason there is no need to create multiple BIER domain's per VPN, the VPN context can be carry in the multicast packet using the procedures as defined in [RFC5331]. Also see [I-D.rosen-l3vpn-mvpn-bier] for more information.

With BIER only a few MVPN profiles will remain relevant, simplifying the operational cost and making it easier to be interoperable among different vendors.

3.2. BUM in EVPN

The current widespread adoption of L2VPN services [RFC4664], especially the upcoming EVPN solution [I-D.ietf-l2vpn-evpn] which transgresses many limitations of VPLS, introduces the need for an efficient mechanism to replicate broadcast, unknown and multicast (BUM) traffic towards the PEs that participate in the same EVPN instances (EVIs). As simplest deployable mechanism, ingress replication is used but poses accordingly a high burden on the ingress node as well as saturating the underlying links with many copies of the same frame headed to different PEs. Fortunately enough, EVPN signals internally P-Multicast Service Interface (PMSI) [RFC6513] attribute to establish transport for BUM frames and with that allows to deploy a plethora of multicast replication services that the underlying network layer can provide. It is therefore relatively simple to deploy BIER P-Tunnels for EVPN and with that distribute BUM traffic without building of P-router state in the core required by PIM, mLDP or comparable solutions.

Specifically, the same I-PMSI attribute suggested for mVPN can be used easily in EVPN and given EVPN can multiplex and disassociate BUM frames on p2mp and mp2mp trees using upstream assigned labels, BIER P-Tunnel will support BUM flooding for any number of EVIs over a single sub-domain for maximum scalability but allow at the other extreme of the spectrum to use a single BIER sub-domain per EVI if such a deployment is necessary.

Multiplexing EVIs onto the same PMSI forces the PMSI to span more than the necessary number of PEs normally, i.e. the union of all PEs participating in the EVIs multiplexed on the PMSI. Given the properties of BIER it is however possible to encode in the receiver bitmask only the PEs that participate in the EVI the BUM frame targets. In a sense BIER is an inclusive as well as a selective tree and can allow to deliver the frame to only the set of receivers interested in a frame even though many others participate in the same PMSI.

As another significant advantage, it is imaginable that the same BIER tunnel needed for BUM frames can optimize the delivery of the multicast frames though the signaling of group memberships for the PEs involved has not been specified as of date.

3.3. IPTV and OTT Services

IPTV is a service, well known for its characteristics of allowing both live and on-demand delivery of media traffic over end-to-end Managed IP network.

Over The Top (OTT) is a similar service, well known for its characteristics of allowing live and on-demand delivery of media traffic between IP domains, where the source is often on an external network relative to the receivers.

Content Delivery Networks (CDN) operators provide layer 4 applications, and often some degree of managed layer 3 IP network, that enable media to be securely and reliably delivered to many receivers. In some models they may place applications within third party networks, or they may place those applications at the edges of their own managed network peerings and similar inter-domain connections. CDNs provide capabilities to help publishers scale to meet large audience demand. Their applications are not limited to audio and video delivery, but may include static and dynamic web content, or optimized delivery for Massive Multiplayer Gaming and similar. Most publishers will use a CDN for public Internet delivery, and some publishers will use a CDN internally within their IPTV networks to resolve layer 4 complexity.

In a typical IPTV environment the egress routers connecting to the receivers will build the tree towards the ingress router connecting to the IPTV servers. The egress routers would rely on IGMP/MLD (static or dynamic) to learn about the receiver's interest in one or more multicast group/channels. Interestingly, BIER could allows provisioning any new multicast group/channel by only modifying the channel mapping on ingress routers. This is deemed beneficial for the linear IPTV video broadcasting in which every receivers behind every egress PE routers would receive the IPTV video traffic.

With BIER in IPTV environment, there is no need of tree building from egress to ingress. Further, any addition of new channel or new egress routers can be directly controlled from ingress router. When a new channel is included, the multicast group is mapped to Bit string that includes all egress routers. Ingress router would start sending the new channel and deliver it to all egress routers. As it can be observed, there is no need for static IGMP provisioning in each egress routers whenever a new channel/stream is added. Instead, it can be controlled from ingress router itself by configuring the new group to Bit Mask mapping on ingress router.

With BIER in OTT environment, these edge routers in CDN domain terminating the OTT user session connect to the Ingress BIER routers connecting content provider domains or a local cache server and leverage the scalability benefit that BIER could provide. This may rely on MBGP interoperation (or similar) between the egress of one domain and the ingress of the next domain, or some other SDN control plane may prove a more effective and simpler way to deploy BIER. For a single CDN operator this could be well managed in the Layer 4 applications that they provide and it may be that the initial receiver in a remote domain is actually an application operated by the CDN which in turn acts as a source for the Ingress BIER router in that remote domain, and by doing so keeps the BIER more descrete on a domain by domain basis.

3.4. Multi-service, converged L3VPN network

Increasingly operators deploy single networks for multiple-services. For example a single Metro Core network could be deployed to provide Residential IPTV retail service, residential IPTV wholesale service, and business L3VPN service with multicast. It may often be desired by an operator to use a single architecture to deliver multicast for all of those services. In some cases, governing regulations may additionally require same service capabilities for both wholesale and retail multicast services. To meet those requirements, some operators use multicast architecture as defined in [RFC5331]. However, the need to support many L3VPNs, with some of those L3VPNs scaling to hundreds of egress PE's and thousands of C-multicast

flows, make scaling/efficiency issues defined in earlier sections of this document even more prevalent. Additionally support for ten's of millions of BGP multicast A-D and join routes alone could be required in such networks with all consequences such a scale brings.

With BIER, again there is no need of tree building from egress to ingress for each L3VPN or individual or group of c-multicast flows. As described earlier on, any addition of a new IPTV channel or new egress router can be directly controlled from ingress router and there is no flooding concern when implementing [RFC5331].

3.5. Control-plane simplification and SDN-controlled networks

With the advent of Software Defined Networking, some operators are looking at various ways to reduce the overall cost of providing networking services including multicast delivery. Some of the alternatives being consider include minimizing capex cost through deployment of network-elements with simplified control plane function, minimizing operational cost by reducing control protocols required to achieve a particular service, etc. Segment routing as described in [I-D.ietf-spring-segment-routing] provides a solution that could be used to provide simplified control-plane architecture for unicast traffic. With Segment routing deployed for unicast, a solution that simplifies control-plane for multicast would thus also be required, or operational and capex cost reductions will not be achieved to their full potential.

With BIER, there is no longer a need to run control protocols required to build a distribution tree. If L3VPN with multicast, for example, is deployed using [RFC5331] with MPLS in P-instance, the MPLS control plane would no longer be required. BIER also allows migration of C-multicast flows from non-BIER to BIER-based architecture, which makes transition to control-plane simplified network simpler to operationalize. Finally, for operators, who would desire centralized, offloaded control plane, multicast overlay as well as BIER forwarding could migrate to controller-based programming.

3.6. Data center Virtualization/Overlay

Virtual eXtensible Local Area Network (VXLAN) [RFC7348] is a kind of network virtualization overlay technology which is intended for multi-tenancy data center networks. To emulate a layer2 flooding domain across the layer3 underlay, it requires to have a mapping between the VXLAN Virtual Network Instance (VNI) and the IP multicast group in a ratio of 1:1 or n:1. In other words, it requires to enable the multicast capability in the underlay. For instance, it requires to enable PIM-SM [RFC4601] or PIM-BIDIR [RFC5015] multicast

routing protocol in the underlay. VXLAN is designed to support 16M VNIs at maximum. In the mapping ratio of 1:1, it would require 16M multicast groups in the underlay which would become a significant challenge to both the control plane and the data plane of the data center switches. In the mapping ratio of n:1, it would result in inefficiency bandwidth utilization which is not optimal in data center networks. More importantly, it is recognized by many data center operators as a unaffordable burden to run multicast in data center networks from network operation and maintenance perspectives. As a result, many VXLAN implementations are claimed to support the ingress replication capability since ingress replication eliminates the burden of running multicast in the underlay. Ingress replication is an acceptable choice in small-sized networks where the average number of receivers per multicast flow is not too large. However, in multi-tenant data center networks, especially those in which the NVE functionality is enabled on a high amount of physical servers, the average number of NVEs per VN instance would be very large. As a result, the ingress replication scheme would result in a serious bandwidth waste in the underlay and a significant replication burden on ingress NVEs.

With BIER, there is no need for maintaining that huge amount of multicast states in the underlay anymore while the delivery efficiency of overlay BUM traffic is the same as if any kind of stateful multicast protocols such as PIM-SM or PIM-BIDIR is enabled in the underlay.

3.7. Financial Services

Financial services extensively rely on IP Multicast to deliver stock market data and its derivatives, and critically require optimal latency path (from publisher to subscribers), deterministic convergence (so as to deliver market data derivatives fairly to each client) and secured delivery.

Current multicast solutions e.g. PIM, mLDP etc., however, don’t sufficiently address the above requirements. The reason is that the current solutions are primarily subscriber driven i.e. multicast tree is setup using reverse path forwarding techniques, and as a result, the chosen path for market data may not be latency optimal from publisher to the (market data) subscribers.

As the number of multicast flows grows, the convergence time might increase and make it somewhat nondeterministic from the first to the last flow depending on platforms/implementations. Also, by having more protocols in the network, the variability to ensure secured delivery of multicast data increases, thereby undermining the overall security aspect.

BIER enables setting up the most optimal path from publisher to subscribers by leveraging unicast routing relevant for the subscribers. With BIER, the multicast convergence is as fast as unicast, uniform and deterministic regardless of number of multicast flows. This makes BIER a perfect multicast technology to achieve fairness for market derivatives per each subscriber.

4. Security Considerations

There are no security issues introduced by this draft.

5. IANA Considerations

There are no IANA consideration introduced by this draft.

6. Acknowledgments

The authors would like to thank IJsbrand Wijnands, Greg Shepherd and Christian Martin for their contribution.

7. References

7.1. Normative References

[I-D.rosen-l3vpn-mvpn-bier]
Rosen, E., Sivakumar, M., Wijnands, I., Aldrin, S., Dolganow, A., and T. Przygienda, "Multicast VPN Using BIER", draft-rosen-l3vpn-mvpn-bier-02 (work in progress), December 2014.

[I-D.wijnands-bier-architecture]
Wijnands, I., Rosen, E., Dolganow, A., Przygienda, T., and S. Aldrin, "Multicast using Bit Index Explicit Replication", draft-wijnands-bier-architecture-04 (work in progress), February 2015.

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

7.2. Informative References

[I-D.ietf-l2vpn-evpn]
Sajassi, A., Aggarwal, R., Bitar, N., Isaac, A., and J. Uttaro, "BGP MPLS Based Ethernet VPN", draft-ietf-l2vpn-evpn-11 (work in progress), October 2014.

[I-D.ietf-spring-segment-routing]
Filsfils, C., Previdi, S., Bashandy, A., Decraene, B., Litkowski, S., Horneffer, M., Shakir, R., Tantsura, J., and E. Crabbe, "Segment Routing Architecture", draft-ietf-spring-segment-routing-01 (work in progress), February 2015.

[RFC4601] Fenner, B., Handley, M., Holbrook, H., and I. Kouvelas, "Protocol Independent Multicast - Sparse Mode (PIM-SM): Protocol Specification (Revised)", RFC 4601, August 2006.

[RFC4664] Andersson, L. and E. Rosen, "Framework for Layer 2 Virtual Private Networks (L2VPNs)", RFC 4664, September 2006.

[RFC5015] Handley, M., Kouvelas, I., Speakman, T., and L. Vicisano, "Bidirectional Protocol Independent Multicast (BIDIR-PIM)", RFC 5015, October 2007.

[RFC5331] Aggarwal, R., Rekhter, Y., and E. Rosen, "MPLS Upstream Label Assignment and Context-Specific Label Space", RFC 5331, August 2008.

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

[RFC7348] Mahalingam, M., Dutt, D., Duda, K., Agarwal, P., Kreeger, L., Sridhar, T., Bursell, M., and C. Wright, "Virtual eXtensible Local Area Network (VXLAN): A Framework for Overlaying Virtualized Layer 2 Networks over Layer 3 Networks", RFC 7348, August 2014.

Authors' Addresses

Nagendra Kumar
Cisco
7200 Kit Creek Road
Research Triangle Park, NC 27709
US

Email: naikumar@cisco.com

Rajiv Asati
Cisco
7200 Kit Creek Road
Research Triangle Park, NC 27709
US

Email: rajiva@cisco.com

Mach(Guoyi) Chen
Huawei

Email: mach.chen@huawei.com

Xiaohu Xu
Huawei

Email: xuxiaohu@huawei.com

Andrew Dolganow
Alcatel-Lucent
600 March Road
Ottawa, ON K2K2E6
Canada

Email: andrew.dolganow@alcatel-lucent.com

Tony Przygienda
Ericsson
300 Holger Way
San Jose, CA 95134
USA

Email: antoni.przygienda@ericsson.com

Arkadiy Gulko
Thomson Reuters
195 Broadway
New York NY 10007
USA

Email: arkadiy.gulko@thomsonreuters.com

Dom Robinson
id3as-company Ltd
UK

Email: Dom@id3as.co.uk

Network Work group
Internet-Draft
Intended status: Standards Track
Expires: September 6, 2015

N. Kumar
C. Pignataro
N. Akiya
Cisco Systems, Inc.
L. Zheng
M. Chen
Huawei Technologies
G. Mirsky
Ericsson

March 5, 2015

BIER Ping and Trace
draft-kumarzheng-bier-ping-00

Abstract

Bit Index Explicit Replication (BIER) is an architecture that provides optimal multicast forwarding through a "BIER domain" without requiring intermediate routers to maintain any multicast related per-flow state. BIER also does not require any explicit tree-building protocol for its operation. 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). The BFIR router adds a BIER header to the packet. The BIER header contains a bit-string in which each bit represents exactly one BFER to forward the packet to. The set of BFERs to which the multicast packet needs to be forwarded is expressed by setting the bits that correspond to those routers in the BIER header.

This document describes the mechanism and basic BIER OAM packet format that can be used to perform failure detection and isolation on BIER data plane.

This Internet-Draft will expire on September 6, 2015.

Table of Contents

1. Introduction

[I-D.wijnands-bier-architecture] introduces and explains BIER architecture that provides optimal multicast forwarding through a "BIER domain" without requiring intermediate routers to maintain any multicast related per-flow state. BIER also does not require any explicit tree-building protocol for its operation. 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). The BFIR router adds a BIER header to the packet. The BIER header contains a bit-string in which each bit represents exactly one BFER to forward the packet to. The set of BFERs to which the multicast packet needs to be forwarded is expressed by setting the bits that correspond to those routers in the BIER header.

This document describes the mechanism and basic BIER OAM packet format that can be used to perform failure detection and isolation on BIER data plane without any dependency on other layers like IP layer.

2. Conventions used in this document

2.1. Terminology

BFER - Bit Forwarding Egress Router

BFIR - Bit Forwarding Ingress Router

BIER - Bit Index Explicit Replication

ECMP - Equal Cost Multi-Path

OAM - Operation, Administration and Maintenance

SI - Set Identifier

2.2. Requirements notation

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

3. BIER OAM

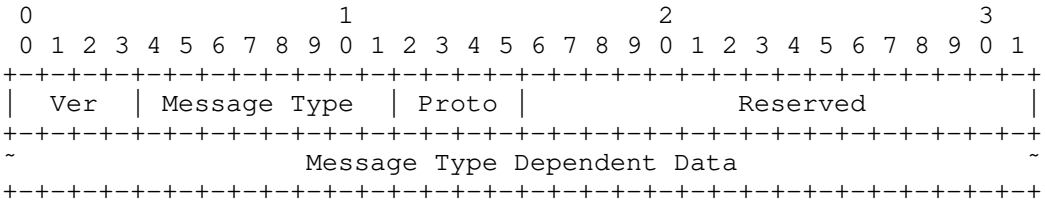
BIER OAM is defined in a way that it stays within BIER layer by following directly the BIER header without mandating the need for IP header. [I-D.wijnands-mpls-bier-encapsulation] defines a 4-bit field as "Proto" to identify the payload following BIER header. In order to differentiate the BIER data packet from BIER OAM packet, this document introduces a new value for the Proto field as:

Proto:

PROTO-TBD1: BIER OAM

3.1. BIER OAM message format

The BIER OAM packet header format that follows BIER header is as follows:

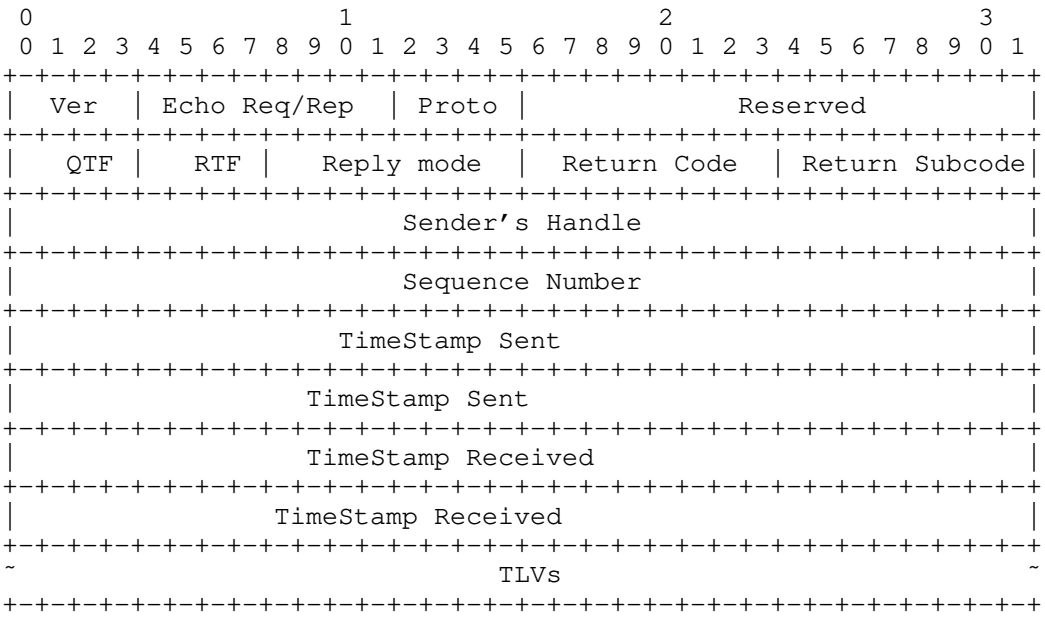


Type

The Message Type is one of the following:

Type	Value Field
TBD1	BIER Echo Request
TBD2	BIER Echo Reply

The Echo Request/Reply header format is as follows:



Proto

Set to 0 for Echo Request/Reply header.

QTF

Querier Timestamp Format. When set to 2, the Timestamp Sent field is (in seconds and microseconds, according to the Initiator's clock) in NTP format [RFC5905]. When set to 3, the timestamp format is in IEEE 1588-2008 (1588v2) Precision Time Protocol format.

RTF

Responder Timestamp Format. When set to 2, the Timestamp Received field is (in seconds and microseconds, according to the Initiator's clock) in NTP format [RFC5905]. When set to 3, the timestamp format is in IEEE 1588-2008 (1588v2) Precision Time Protocol format.

Reply mode

The Reply mode is set to one of the below:

Value	Meaning
1	Do not Reply
2	Reply via IPv4/IPv6 UDP packet.
3	Reply via BIER packet

Return Code

Set to zero if Type is TBD1. Set as defined in section 3.2 if Type is TBD2.

Return subcode

To Be updated.

Sender's Handle, Sequence number and Timestamp

The Sender's Handle is filled by the Initiator, and returned unchanged by responder BFR. This is used for matching the replies to the request.

The Sequence number is assigned by the Initiator and can be used to detect any missed replies.

The Timestamp Sent is the time when the echo request is sent. The TimeStamp Received in echo reply is the time (accordingly to responding BFR clock) that the corresponding echo request was received. The format depends on the QTF/RTF value.

TLVs

Carries the TLVs as defined in Section 3.3.

3.2. Return Code

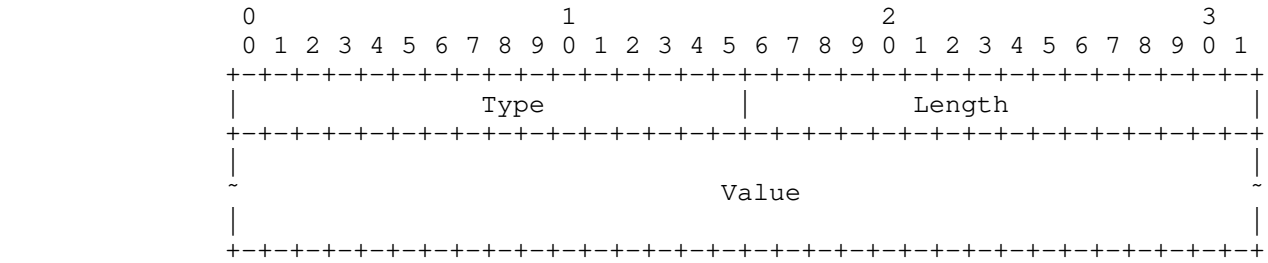
Responder uses Return Code field to reply with validity check or other error message to Initiator. It does not carry any meaning in Echo Request and MUST be set to zero.

The Return Code can be one of the following:

Value	Value Meaning
0	No return code (Set by Initiator in Echo Request)
1	Malformed echo request received
2	One or more of the TLVs was not understood
3	Replying BFR is the only BFER in header Bitstring
4	Set-Identifier Mismatch
5	Packet-Forward-Success
6	Invalid Multipath Info Request
7	No control plane entry for Multicast Overlay Data
8	No matching entry in forwarding table.
9	Replying BFR is one of the BFER in header Bitstring

3.3. BIER OAM TLV

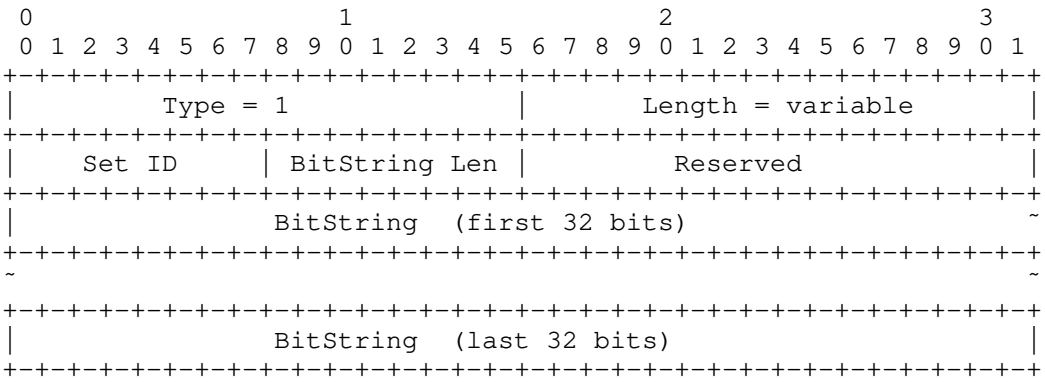
This section defines various TLVs that can be used in BIER OAM packet. The TLVs (Type-Length-Value tuples) have the following format:



TLV Types are defined below; Length is the length of the Value field in octets. The Value field depends on the TLV Type.

3.3.1. Original SI-BitString TLV

The Original SI-BitString TLV carries the set of BFER and carries the same BitString that Initiator includes in BIER header.This TLV has the following format:



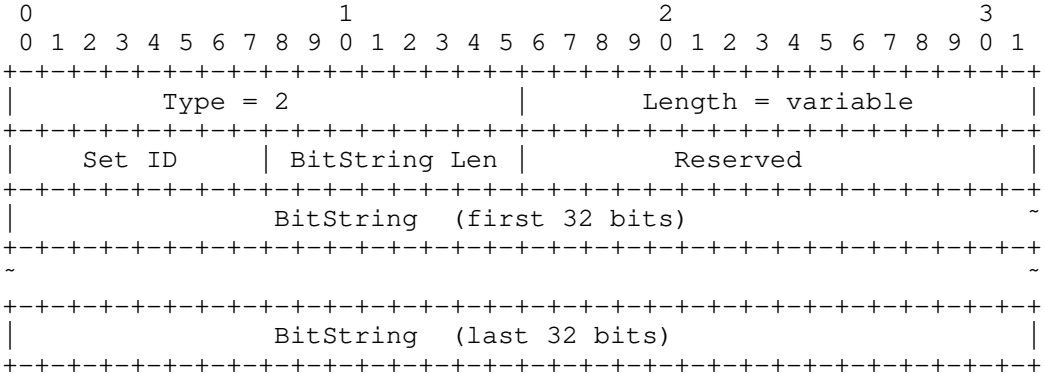
The Length field is set as defined in section 3 of [I-D.wijnands-mpls-bier-encapsulation].

Set ID field is set to the Set Identifier to which the BitString belongs to. This value is derived as defined in section 3 of [I-D.wijnands-bier-architecture]

The BitString field carries the set of BFR-IDs that Initiator will include in the BIER header. This TLV MUST be included by Initiator in Echo Request packet

3.3.2. Target SI-BitString TLV

The Target SI-BitString TLV carries the set of BFER from which the Initiator expects the reply from.This TLV has the following format:



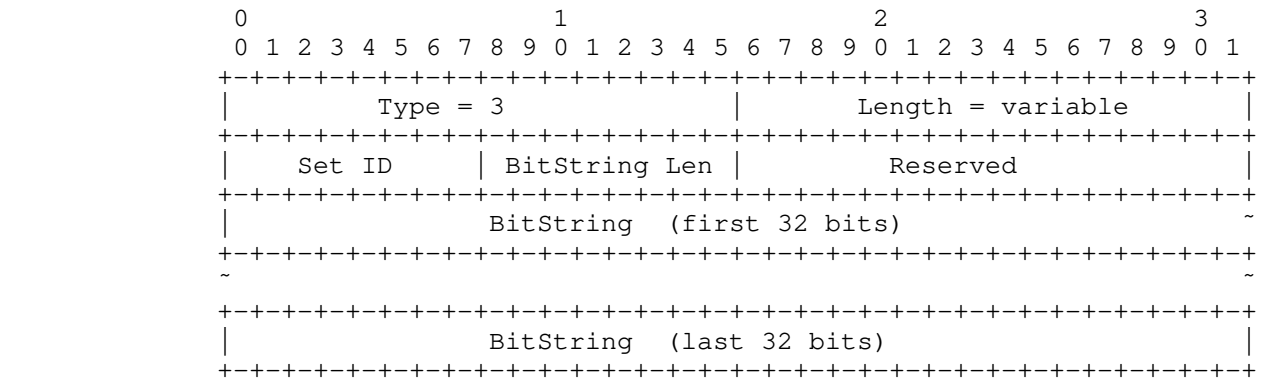
The Length field is set as defined in section 3 of [I-D.wijnands-mpls-bier-encapsulation].

Set ID field is set to the Set Identifier to which the BitString belongs to. This value is derived as defined in section 3 of [I-D.wijnands-bier-architecture]

The BitString field carries the set of BFR-IDs of BFER(s) that Initiator expects the response from. The BitString in this TLV may be different from the BitString in BIER header and allows to control the BFER responding to the Echo Request. This TLV MUST be included by Initiator in BIER OAM packet if the Downstream Mapping TLV (section 3.3.4) is included.

3.3.3. Incoming SI-BitString TLV

The Incoming SI-BitString TLV will be included by Responder BFR in Reply message and copies the BitString from BIER header of incoming Echo Request message. This TLV has the following format:



The Length field is set as defined in section 3 of [I-D.wijnands-mpls-bier-encapsulation].

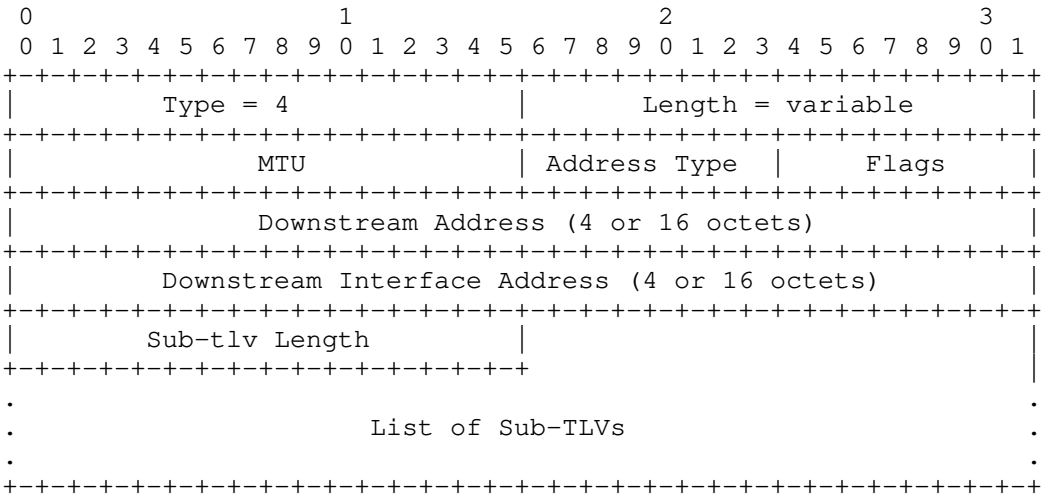
Set ID field is set to the Set Identifier of the incoming BIER-MPLS label. This value is derived as defined in section 2.2 of [I-D.psenak-ospf-bier-extensions]

The BitString field copies the BitString from BIER header of the incoming Echo Request. A Responder BFR SHOULD include this TLV in Echo Reply if the Echo Request is received with I flag set in Downstream Mapping TLV.

An Initiator MUST NOT include this TLV in Echo Request.

3.3.4. Downstream Mapping TLV

This TLV has the following format:



MTU

Set to MTU value of outgoing interface.

Address Type

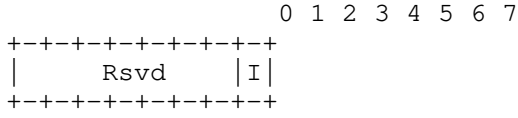
The Address Type indicates the address type and length of IP address for downstream interface. The Address type is set to one of the below:

Type	Addr. Type	DA Length	DIA Length
1	IPv4 Numbered	4	4
2	IPv4 Unnumbered	4	4
3	IPv6 Numbered	16	16
4	IPv6 Unnumbered	16	4

DA Length - Downstream Address field Length
DIA Length - Downstream Interface Address field Length

Flags

The Flags field has the following format:



When I flag is set, the Responding BFR SHOULD include the Incoming SI-BitString TLV in echo reply message.

Downstream Address and Downstream Interface Address

If the Address Type is 1, the Downstream Address MUST be set to IPV4 BFR-Prefix of downstream BFR and Downstream Interface Address is set to downstream interface address.

If the Address Type is 2, the Downstream Address MUST be set to IPV4 BFR-Prefix of downstream BFR and Downstream Interface Address is set to the index assigned by upstream BFR to the interface.

If the Address Type is 3, the Downstream Address MUST be set to IPV6 BFR-Prefix of downstream BFR and Downstream Interface Address is set to downstream interface address.

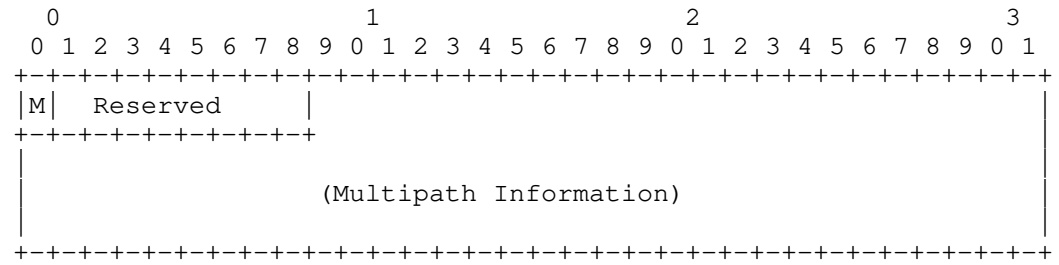
If the Address Type is 4, the Downstream Address MUST be set to IPv6 BFR-Prefix of downstream BFR and Downstream Interface Address is set to index assigned by upstream BFR to the interface.

3.3.4.1. Downstream Mapping Sub-TLVs

This section defines the optional Sub-TLVs that can be included in Downstream Mapping TLV.

Sub-TLV Type	Value
1	Multipath Entropy Data
2	Egress BitString

3.3.4.1.1. Multipath Entropy Data



M Flag

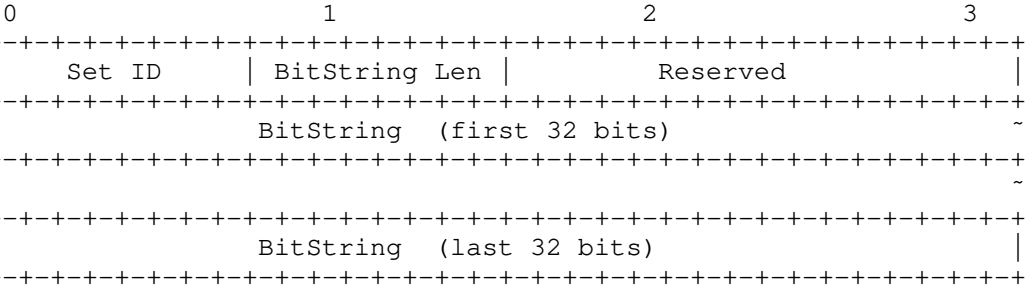
This flag is set to 0 if all packets will be forwarded out through interface defined in Downstream Mapping TLV. When set to 1, Multipath Information will be defined with Bit masked Entropy data.

Multipath Information

Entropy Data encoded as defined in section x3.

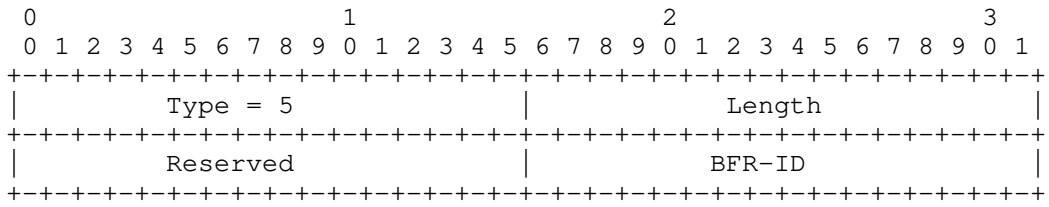
3.3.4.1.2. Egress BitString

This Sub-TLV MAY be included by Responder BFR with the rewritten BitString in outgoing interface as defined in section 6.1 of [I-D.wijnands-bier-architecture]



3.3.5. Responder BFER TLV

The Responder BFER TLV will be included by the BFER replying to the request. This is used to identify the originator of BIER Echo Reply. This TLV have the following format:

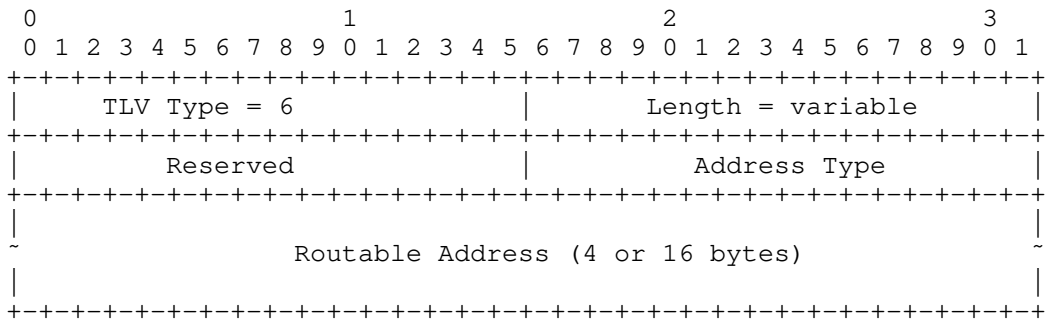


BFR-ID

The BFR-ID field carries the BFR-ID of replying BFER. This TLV MAY be included by Responding BFER in BIER Echo Reply packet.

3.3.6. Responder BFR TLV

The Responder BFR TLV will be included by the transit BFR replying to the request. This is used to identify the replying BFR without BFR-ID. This TLV have the following format:



Length

The Length field varies depending on the Address Type.

Address Type

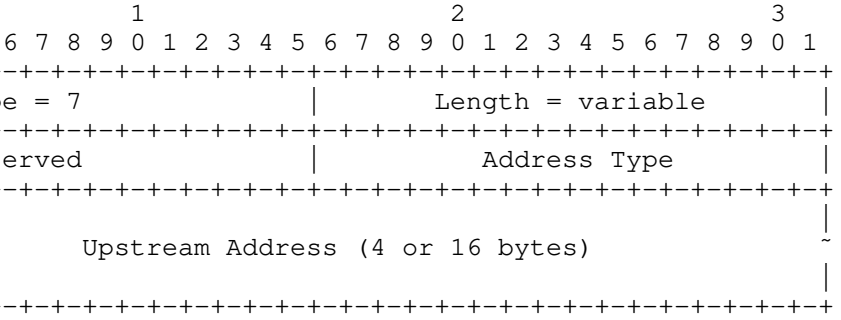
Set to 1 for IPv4 or 2 for IPv6.

Routable Address

Carries any locally routable address of replying BFR. This TLV MAY be included by Responding BFR in BIER Echo Reply packet.

3.3.7. Upstream Interface TLV

The Upstream Interface TLV will be included by the replying BFR in Echo Reply. This is used to identify the incoming interface and the BIER-MPLS label in the incoming Echo Request. This TLV have the following format:



Length

The Length field varies depending on the Address Type.

Address Type

Set to 1 for IPv4 or 2 for IPv6.

Upstream Address

As defined in Section 3.3.4

4. Procedures

This section describes aspects of Ping and traceroute operations. While this document explains the behavior when Reply mode is "Reply via BIER packet", the future version will be updated with details about the format when the reply mode is "Reply via IP/UDP packet".

4.1. BIER OAM processing

A BIER OAM packet MUST be sent to BIER control plane for OAM processing if one of the following conditions is true:

- The receiving BFR is a BFER.
- TTL of BIER-MPLS Label expired.

- Presence of Router Alert label in the label stack.

4.2. Per BFER ECMP Discovery

As defined in [I-D.wijnands-bier-architecture], BIER follows unicast forwarding path and allows load balancing over ECMP paths between BFIR and BFER. BIER OAM MUST support ECMP path discovery between a BFIR and a given BFER and MUST support path validation and failure detection of any particular ECMP path between BFIR and BFER.

[I-D.wijnands-mpls-bier-encapsulation] proposes the BIER header with Entropy field that can be leveraged to exercise all ECMP paths. Initiator/BFIR will use traceroute message to query each hop about the Entropy information for each downstream paths. To avoid complexity, it is suggested that the ECMP query is performed per BFER by carrying required information in BIER OAM message.

Initiator MUST include Multipath Entropy Data Sub-TLV in Downstream Mapping TLV. It MUST also include the BFER in BitString TLV to which the Multipath query is performed.

Any transit BFR will reply back with Bit-masked Entropy for each downstream path as defined in [RFC4379]

4.3. Sending BIER Echo Request

Initiator MUST set the Message Type as TBD1 and Return Code as 0. Initiator infer the Set Identifier value from the respective BitString that will be appended in BIER header and include in "SI" field.

The Proto field in OAM packet MUST be set to 0, if there is no data packet following immediately after OAM packet. In all other cases, the Proto field MUST be set to value as defined in [I-D.wijnands-mpls-bier-encapsulation], same as of the data packet that follows after OAM packet.

Initiator MUST include Original SI-BitString TLV. Initiator MUST NOT include more than one Original SI-BitString TLV. In Ping mode, Initiator MAY include Target SI-BitString TLV listing all the BFER from which the Initiator expects a response. In traceroute mode, Initiator SHOULD include Target SI-Bitstring TLV. Initiator on receiving a reply with Return code as "Replying router is one of the BFER in BIER header Bitstring" , SHOULD unset the respective BFR-id from Target SI-Bitstring on any subsequent request.

Initiator MAY also include Downstream Mapping TLV (section 3.3.4). In presence of Multipath Entropy Data Sub-TLV, the Target SI-BitString TLV MUST carry only one BFER id.

Initiator then encapsulates with BIER header and set the Proto as TBD1 and further encapsulates with BIER-MPLS label. In ping mode, the BIER-MPLS Label TTL MUST be set to 255. In traceroute mode, the BIER-MPLS Label TTL is set successively starting from 1 and MUST stop sending the Echo Request if it receives a reply with Return code as "Replying router is the only BFER in BIER header Bitstring" from all BFER listed in BitString TLV.

4.4. Receiving BIER Echo Request

Sending a BIER OAM Echo Request to control plane for payload processing is triggered as mentioned in section 4.1.

Any BFR on receiving Echo Request MUST send Echo Reply with Return Code set to 1, if the packet fails sanity check. If the packet sanity check is fine, it initiates a variable as "Best-return-code" and further processes it as below:

1. Set the Best-return-code to "SI Mismatch", if the received BIER-MPLS Label is not assigned to the Set ID value in Original SI-BitString TLV. Go to section 4.5.

2. Set the Best-return-code to "One or more of the TLVs was not understood", if any of the TLVs in echo request message is not understood. Go to section 4.5.

3. Set the Best-return-code to "Invalid Multipath Info Request", if the Echo Request is received with more than 1 BFER id in Target SI-BitString TLV AND Multipath Entropy Data Sub-TLV. Go to section 4.5.

4. Set the Best-return-code to "Replying router is the only BFER in BIER header Bitstring", and go to section 4.5 if the responder is BFER and there is no more bits in BIER header Bitstring left for forwarding.

5. Set the Best-return-code to "Replying router is one of the BFER in BIER header Bitstring", and include Downstream Mapping TLV, if the responder is BFER and there is more bits in BitString left for forwarding. In addition, include the Multipath information as defined in Section 4.2 if the received Echo Request carries Multipath Entropy Data Sub-TLV. Go to section 4.5.

6. Set the Best-return-code to "No matching entry in forwarding table", if the forwarding lookup defined in section 6.5 of [I-D.wijnands-bier-architecture] does not match any entry for the received BitString in BIER header.

7. Set the Best-return-code to "Packet-Forward-OK", and include Downstream Mapping TLV. Go to section 4.5

4.5. Sending Echo Reply

Responder MUST include DDMAP in Echo Reply if the incoming Echo Request carries DDMAP. Responder MUST set the Message Type as TBD2 and Return Code as Best-return-code. The SI field MUST be set to 0 and Proto field MUST be set to 0.

Responder appends BIER header listing the BitString with BFIR ID (from received Echo Request), set the Proto to PROTO-TBD1 and set the BFIR as zero.

4.6. Receiving Echo Reply

Initiator on receiving echo reply will use the Sender's Handle to match with echo request sent. If no match is found, Initiator MUST ignore the Echo Reply.

If receiving echo reply have Downstream Mapping, Initiator SHOULD copy the same to subsequent Echo Request(s).

5. IANA Considerations

This document request the IANA the creation and management of below registries:

5.1. Message Types, Reply Modes, Return Codes

This document request to assign the Message Types and Reply mode mentioned in section 3.1, , Return code mentioned in Section 3.2.

5.2. TLVs

The TLVs and Sub-TLVs requested by this document for IANA consideration are the following:

Type	Sub-Type	Value Field
1		Original SI-BitString
2		Target SI-BitString
3		Incoming SI-BitString
4		Downstream Mapping
4	1	Multipath Entropy Data
4	2	Egress BitString
5		Responder BFER
6		Responder BFR
7		Upstream Interface

6. Security Considerations

The security consideration for BIER Ping is similar to ICMP or LSP Ping. AS like ICMP or LSP ping, BFR may be exposed to Denial-of-service attacks and it is RECOMMENDED to regulate the BIER Ping packet flow to control plane. A rate limiter SHOULD be applied to avoid any attack.

As like ICMP or LSP Ping, a traceroute can be used to obtain network information. It is RECOMMENDED that the implementation check the integrity of BFIR of the Echo messages against any local secured list before processing the message further

7. Acknowledgement

TBD

8. Contributing Authors

TBD

9. References

9.1. Normative References

[I-D.psenak-ospf-bier-extensions]
Psenak, P., Kumar, N., Wijnands, I., Dolganow, A., Przygienda, T., and J. Zhang, "OSPF Extensions For BIER", draft-psenak-ospf-bier-extensions-02 (work in progress), February 2015.

[I-D.wijnands-bier-architecture]
Wijnands, I., Rosen, E., Dolganow, A., Przygienda, T., and S. Aldrin, "Multicast using Bit Index Explicit Replication", draft-wijnands-bier-architecture-04 (work in progress), February 2015.

[I-D.wijnands-mpls-bier-encapsulation]
Wijnands, I., Rosen, E., Dolganow, A., Tantsura, J., and S. Aldrin, "Encapsulation for Bit Index Explicit Replication in MPLS Networks", draft-wijnands-mpls-bier-encapsulation-02 (work in progress), December 2014.

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

[RFC4379] Kompella, K. and G. Swallow, "Detecting Multi-Protocol Label Switched (MPLS) Data Plane Failures", RFC 4379, February 2006.

[RFC5905] Mills, D., Martin, J., Burbank, J., and W. Kasch, "Network Time Protocol Version 4: Protocol and Algorithms Specification", RFC 5905, June 2010.

9.2. Informative References

[RFC0792] Postel, J., "Internet Control Message Protocol", STD 5, RFC 792, September 1981.

[RFC6291] Andersson, L., van Helvoort, H., Bonica, R., Romascanu, D., and S. Mansfield, "Guidelines for the Use of the "OAM" Acronym in the IETF", BCP 161, RFC 6291, June 2011.

[RFC6424] Bahadur, N., Kompella, K., and G. Swallow, "Mechanism for Performing Label Switched Path Ping (LSP Ping) over MPLS Tunnels", RFC 6424, November 2011.

[RFC6425] Saxena, S., Swallow, G., Ali, Z., Farrel, A., Yasukawa, S., and T. Nadeau, "Detecting Data-Plane Failures in Point-to-Multipoint MPLS - Extensions to LSP Ping", RFC 6425, November 2011.

Authors' Addresses

Nagendra Kumar
Cisco Systems, Inc.
7200 Kit Creek Road
Research Triangle Park, NC 27709
US

Email: naikumar@cisco.com

Carlos Pignataro
Cisco Systems, Inc.
7200 Kit Creek Road
Research Triangle Park, NC 27709-4987
US

Email: cpignata@cisco.com

Nobo Akiya
Cisco Systems, Inc.
2000 Innovation Drive
Kanata, ON K2K 3E8
Canada

Email: nobo@cisco.com

Lianshu Zheng
Huawei Technologies
China

Email: vero.zheng@huawei.com

Mach Chen
Huawei Technologies

Email: mach.chen@huawei.com

Greg Mirsky
Ericsson

Email: gregory.mirsky@ericsson.com

Internet Engineering Task Force
Internet-Draft
Intended status: Standards Track
Expires: September 10, 2015

A. Przygienda
J. Tantsura
Ericsson

March 09, 2015

Automatic Assignment of BIER BFR-ids in ISIS
draft-prz-bier-bfrid-assignment-00

Abstract

Specification of an ISIS extension to support auto-election of BFR IDs in BIER using ISIS.

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 RFC 2119 [RFC2119] .

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 http://datatracker.ietf.org/drafts/current/.

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

This Internet-Draft will expire on September 10, 2015.

Table of Contents

1. Introduction

Bit Index Explicit Replication (BIER) [I-D.draft-wijnands-bier-architecture-04] defines an architecture where all intended multicast receivers are encoded as bitmask in the Multicast packet header within different encapsulations such as [I-D.draft-wijnands-mpls-bier-encapsulation-02]. A router that receives such a packet will forward the packet based on the Bit Position in the packet header towards the receiver(s), following a precomputed tree for each of the bits in the packet. Each receiver is represented by a unique bit in the bitmask corresponding to its BFR-id. BFR-ids are sub-domain specific.

Once the number of receivers becomes large (i.e. many sets are present) or receivers choose to participate in many independent sub-domains, assignment of a unique BIER bit to a node is a non-trivial problem that can benefit highly from an automated solution. The

usual trade-offs are either a centralized (server) approach or a distributed approach which (from experience with other protocols such as DHCP or OSPF), provide at the cost of additional protocol complexity higher availability.

This document presents necessary, optional extensions to the currently deployed ISIS for IP [RFC1195] protocol to support automatic election of BFR-ids by means of a distributed protocol. This document defines new TLVs to be advertised by every router participating in BIER signaling and supporting such an election. In case some nodes are statically configured with a BFR-id, the protocol can detect misconfiguration, i.e. overlapping bit assignments or otherwise respects statically assigned BFR ids.

This extension operates seamlessly in a backwards compatible fashion with BIER procedures for ISIS as defined in [I-D.draft-przygienda-bier-isis-ranges-02]. Only BFRs implementing this extensions benefit from automatic assignment.

2. Terminology

Some of the terminology specified in [I-D.draft-wijnands-bier-architecture-04] is replicated here and extended by necessary definitions:

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

BIER-OL: BIER Overlay Signaling. (The method for the BFIR to learn about BFER's).

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

BFIR: Bit Forwarding Ingress Router (The ingress border router that inserts the BM into the packet).

BFER: Bit Forwarding Egress Router. A router that participates in Bit Index Forwarding as leaf. Each BFER must be a BFR. Each BFER must have a valid BFR-id assigned.

BFT: Bit Forwarding Tree used to reach all BFERs in a domain.

BIFT: Bit Index Forwarding Table.

BMS: Bit Mask Set. Set containing bit positions of all BFER participating in a set.

BMP: Bit Mask Position, a given bit in a BMS.

Invalid BMP: Unassigned Bit Mask Position, consisting of all 0s.

IGP signalled BIER domain: A BIER underlay where the BIER synchronization information is carried in IGP. Observe that a multi-topology is NOT a separate BIER domain in IGP.

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

BFR-id: An optional, unique identifier for a BFR within a BIER sub-domain.

Invalid BFR-id: Unassigned BFR-id, consisting of all 0s.

3. IANA Considerations

This document adds the following new sub-sub-TLVs to the registry of sub-TLVs for BIER Info sub-TLV.

BIER Protocol Election sub-sub-TLV Value: TBD (suggested - to be assigned by IANA)

This document adds the following new TLV to the registery of ISIS TLVs.

BIER PE BMP Assignments TLV Value: TBD (suggested - to be assigned by IANA)

4. Procedures

The following sections present BIER IGP protocol procedures for the auto-election and maintainance of unique BIER BFR-ids across subdomains. Compared to purely administrative assignment of the bitmask use of those procedures largely facilitates deployment of BIER in large setups. The election and bit assignment procedures described in the according sections indicate how the BFRs participate in an election mechanism that allows them to

- use a dynamically chosen Designated and Backup Designated router for coordination and distribution of necessary state across all participants in the set across the network in a robust fashion
- allocate the necessary BMP in a sub-domain for each BFER

- automatically or administratively partition the elections for different sub-domains across the set of BFRs for maximum reliability

- discover administrative misconfiguration of BFERs

4.1. Election Algorithm

After a sub-domain <MT,SD,MLs> [I-D.draft-przygienda-bier-isis-ranges-02] is enabled, the according election procedures for D-BFR and Backup D-BFR are performed based upon the set of available BIER-PE sub-sub-TLVs. Given the fact that SD is uniquely tied to its MT per today's architecture and MLs are of no further importance to the introduced procedures, a sub-domain will be abbreviated without loss of generality as <SD>.

The election is indebted to and largely modeled (to the point of quoting parts of it verbatim) after the DR OSPF Election procedure in OSPF [RFC2328] which has proven to work exceedingly well over many years in the field.

This section describes the algorithm used for calculating a network's Designated BFR and Backup Designated BFR and procedures that allow those to allocate bit mask bits to a participating BFER in a sub-domain SD which we designate as BFER<SD>. The election runs per SD the router is participating in. The initial time a router runs the election algorithm, the D-BFR<SD> and BD-BFR<SD> are initialized to 0.0.0.0 or equivalent empty router ID. This indicates the lack of both a Designated BFR<SD> and a Backup Designated BFR<SD>.

The D-BFR<SD> election algorithm proceeds as follows:

- Call the router doing the calculation Router X<SD>. A router can participate in multiple elections for other BMS and multi-topologies at varying priorities.

- The list of BFRs participating in <SD> whose according BIER-PEs<SD> have been received by Router X<SD> and are connected (i.e. reachable via SPF computation) in standard topology MUST be examined.

- Router X<SD> itself MUST be also considered to be on the list.

- Discard all routers from the list that are ineligible to become D-BFR<SD>. (Routers having Router Priority of 0 for <SD> MUST NOT be eligible to become D-BFR<SD>.)

The following steps MUST then be executed, considering only those routers that remain on the list:

1. Note the current values for D-BFR<SD> and Backup D-BFR<SD>. This is used later for comparison purposes.

2. Calculate the new Backup D-BFR<SD> as follows.

   * Only those routers on the list that have not declared themselves to be D-BFR<SD> MUST be eligible to become Backup D-BFR<SD>.

   * If one or more of these routers have declared themselves Backup D-BFR<SD> (i.e., they are currently listing themselves as Backup D-BFR<SD>, but not as D-BFR<SD>, in their according BIER-PE packets) the one having highest Router Priority for <SD> MUST be declared to be Backup D-BFR<SD>.

   * In case of a tie, the one having the highest Router ID XOR'ed with SD (assuming big endian order, both values right-aligned and all bits of the shorter value filled up with zeroes to the length of the longer value) MUST be chosen.

   * If no routers have declared themselves Backup D-BFR<SD>, the router having highest Router Priority for <SD> MUST be chosen, (again excluding those routers who have declared themselves D-BFR<SD>), and again use the Router ID XOR'ed with SD to break ties.

3. Calculate the new D-BFR<SD> for the network as follows. If one or more of the routers have declared themselves D-BFR<SD> (i.e., they are currently listing themselves as D-BFR<SD> in their BIER-PE advertisements) the one having highest Router Priority for <SD> is declared to be D-BFR<SD>. In case of a tie, the one having the highest Router ID XOR'ed with SD is chosen. If no routers have declared themselves D-BFR<SD>, assign the D-BFR<SD> to be the same as the newly elected BD-BFR<SD>.

4. If Router X<SD> is now newly the D-BFR<SD> or newly the BD-BFR<SD>, or is now no longer the D-BFR<SD> or no longer the BD-BFR<SD>, repeat steps 2 and 3, and then proceed to step 5. For example, if Router X<SD> is now the D-BFR<SD>, when step 2 is repeated X<SD> will no longer be eligible for BD-BFR<SD> election. Among other things, this will ensure that no router will declare itself both BD-BFR<SD> and D-BFR<SD>.

5. As a result of these calculations, the router itself may now be D-BFR<SD> or BD-BFR<SD>. See Section 4.2 and Section 4.3 for the additional duties this would entail.

6. If the above calculations have caused the identity of either the D-BFR<SD> or BD-BFR<SD> to change, all routers must re-evaluate whether they have been elected D-BFR<SD> or BD-BFR<SD> and initiate according procedures. In case the new D-BFR<SD> or BD-BFR<SD> is not advertising according bitmask assignment and they are needed, they initiate according procedures in Section 4.2.1.

The reason behind the election algorithm's complexity is the desire for an orderly transition from BD-BFR<SD> to D-BFR<SD>, when the current D-BFR<SD> fails. This orderly transition is ensured through the introduction of hysteresis: no new BD-BFR<SD> can be chosen until the old Backup accepts its new D-BFR<SD> responsibilities.

The above procedure may elect the same router to be both D-BFR<SD> and BD-BFR<SD>, although that router will never be the calculating router (Router X<SD>) itself. The elected D-BFR<SD> may not be the router having the highest Router Priority for <SD>, nor will the BD-BFR<SD> necessarily have the second highest Router Priority. If Router X<SD> is not itself eligible to become D-BFR<SD>, it is possible that neither a BD-BFR<SD> nor a D-BFR<SD> will be selected in the above procedure. Note also that if Router X<SD> is the only router that is eligible to become D-BFR<SD>, it will select itself as D-BFR<SD> and there will be no BD-BFR<SD> for the network.

4.2. D-BFR Procedures

A router that assumes D-BFR role for a given <SD> combination invokes additional set of procedures as synchronization and election point for all the BFRs in <SD>.

4.2.1. Assignment of BMPs to BFERs in <SD>

Each BFER includes a strongly abbreviated DHCP-like FSM to obtain from the D-BFR<SD> its BMP or to advertise an administrative preference of its BMP.

The procedure is initiated by a BFER<SD> announcing in BIER Info sub-TLV for <SD> its assigned bit (or request for BMP assignment). The D-BFR<SD> initiates then a set of procedures to assign BMPs to such BFER in the <SD> or announces collisions.

Observe that BFERs can request (or announce) the bits even before a BDR<SD> has been chosen so the election and assignment are largely orthogonal sets of procedures.

4.3. BD-BFR Procedures

A router that is elected BD-BFR<SD> MUST mirror in its advertisements the exact state of the D-BFR<SD> and on each received advertisement maintains its internal states to use as starting point in all D-BFR<SD> procedures in case it looses connectivity (i.e. it cannot compute SPF reachability to the D-BFR in standard topology) to the D-BFR<SD>.

4.4. BFER Procedures

A BFER in <SD> controls its BMP in the set by providing values in its BIER Info sub-TLV for <SD> and signalling towards B-DR using A and R bits per Section 9.2. If it advertises the BFR-id without A or R bit set it indicates a fixed value it has chosen administratively.

It may request the assignment of a BMP by setting the R bit. The prefered BFR-id is signalled by providing a BFR-id value. The D-BFR MUST try to keep the preferred setting value when choosing BMP for the BFER. All other BFRs MUST NOT use the BFR-id value when the R bit is set. In case of routers not understanding this extensions, the behavior is enforced by the means of the C bit.

Once the BFER has been assigned a value from D-BFR and is willing to accept it, it MUST copy the value into the BFR-id field in the BIER-PE-BMPs it receives and set the A bit while clearing the R bit.

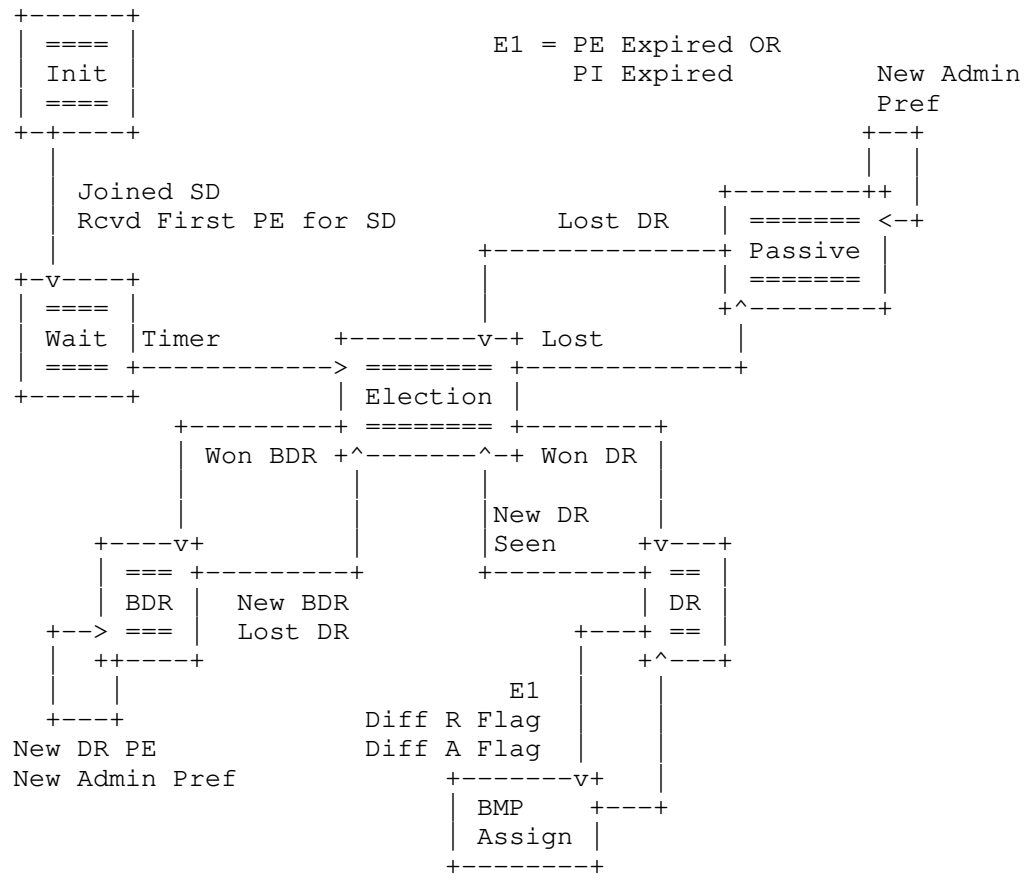
On the other side, the D-BFR for <SD> advertises the BMP assignments by the means of advertising BIER-PE-BMP for <SD>.

5. Special Considerations

5.1. BD-BFER to D-BFER Transition

In the normal case a router will assume its role as D-BFR<SD> promoting itself from BD-BFR<SD> with its own set of procedures. Based on those it will hold the state of the abdicating D-BFR<SD> and it MUST use this state as initial state for the D-BFR procedures it initiates per Section 4.2 . This should warranty a seamless fall-over without changes in the assignments of bits for BFERs for the according <SD> which SHOULD take preference over all other considerations. Observe that the implication is that a configured administrative preference MUST NOT be used unless changed or set explicitly again. The FSMs visualize this behavior more explicitly.

6. Election FSM for BFR<SD>



The full set of procedures can be described as a finite state machine per <SD> run within each participating BFR with the following events and transitions

6.1. States

Init Initial State of the Machine

Wait State waiting for routers to update their PEs for <SD> on startup

Election State that runs the election procedures and generates according events that progress it into another state immediately

Passive State entered when lost both DR and BDR in election.

Elected DR

Elected BDR

BMP Assign State in which the assignment of bits happens upons requests from BFERs.

6.2. Events

Timer Initial timer waiting for s of other routers before election is triggered.

Signalling/Rcvd First PE First PE for <SD> has been received or signalling enabled for the set S on BFR.

Lost DR Current D-BFR<SD> cannot be reached anymore via SPF computation in standard topology.

Lost Lost election for D-BFR and BD-BFR.

Won BDR Won election for BD-BFR.

Won DR Won election for D-BFR.

New BDR A new BD-BFR has been elected by the D-BFR.

New DR PE New BIER-PE Instance from D-BFR.

New Admin Pref Changed Administrative preference.

Diff R Flag R flag has been announced by a BFR which was not present before. In case of a new R flag, an assignment should be attempted. In case of R flag being deleted

if the A flag is set, the validity of the copied BFR-id with the assignment is checked

if the A flag is clear, the value is assumed non-negotiable and re-assignments may be necessary

Diff A Flag A flag has been withdrawn or announced. If A flag was present before and

R flag is clear, the value is assumed non-negotiable and re-assignments may be necessary.

R flag is set, a new assignment is requested.

If A flag was not present before and

R flag is clear, the validity of the copied BFR-id with the assignment is checked

R flag is set, the client MUST be declared faulty and disregarded.

To Be Completed TBD

7. FSM Figure/Events for BFER: TBD

8. Backwards Compatiblity

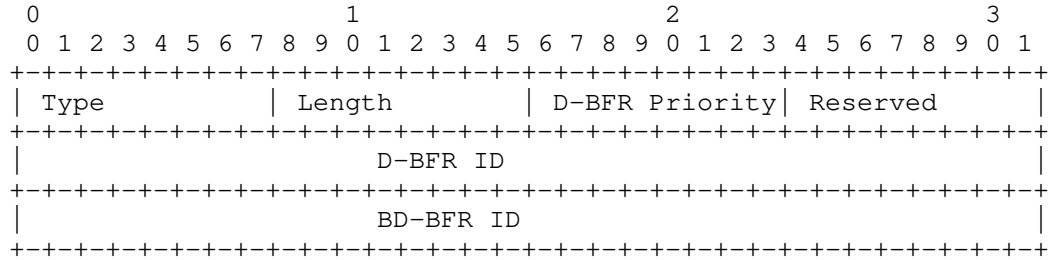
The procedures prescribed guarantee a complete backwards compatiblity to [I-D.draft-przygienda-bier-isis-ranges-02]. During the assignment procedure the according values are hidden from BFRs lacking this extension by the means of the C bit. Once assigned, they become visible. On the other hand, BFR-id values chosen by the BFRs without election extensions are respected in assignment.

9. Packet Formats

Some of the new information is carried within the the existing BIER Info sub-TLV per [I-D.draft-przygienda-bier-isis-ranges-02] and some presents a new ISIS TLV.

9.1. BIER-PE: BIER Protocol Election sub-sub-TLV

This sub-sub-TLV is included in the BIER Info sub-TLV of the according sub-domain as specified by [I-D.draft-przygienda-bier-isis-ranges-02]. It MUST be included in the BIER Info sub-TLV only once, otherwise the first instance is used.



Type: TBD1.

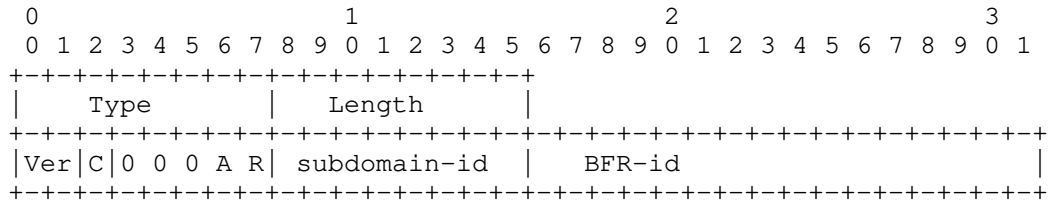
Length: 1 octet.

Priority Priority at which this router is set to become D-BFR for the <SD>.

D-BFR ID ID of the router chosen as D-BFR. If the router elected itself as D-BFR it MUST set it to its own ID.

BD-BFR ID ID of the router chosen as BD-BFR. If the router elected itself as BD-BFR it MUST set it to its own ID.

9.2. Reuse of the Reserved Bits in BIER Info sub-TLV



Version Version of the protocol. It remains at 0.

C The compatiblity bit. It is set according to following rules:

If the R bit is set, C is set to 0, i.e. the TLV is not compatible with version 0 of the BIER information. This will prevent routers not implementing this specification from looking at this advertisement.

If the R bit is clear, C is set to 1. In case the BFR-id has been obtained without an error by requesting it from a D-BFR, the value is copied into BFR-id of this sub-TLV, otherwise it is set to invalid BFR-id.

R Request Bit. When set, this bit advertises that the BFER is willing to accept another BMP than the one administratively

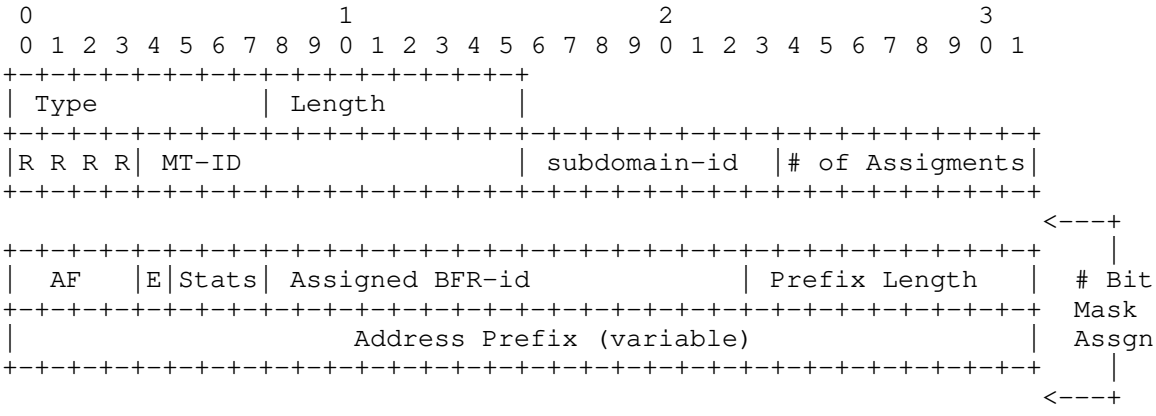
desired from D-BFR<SD>. The value of BMP is then determined by the according element in BIER-PE-BMP of the D-BFR<SD>.

A When this bit is set, the BFER advertises that the value indicated in the BFR-id has been copied from the assignment provided by D-BFR. If clear and BFR-id is set, the value is administratively assigned and is non-negotiable.

BFR-id If set and R bit is clear, it indicates the BFR-id the BFR is occupying to the D-BFR. If the R bit is set, it indicates the desired BFR-id to be assigned or no preference.

9.3. BIER-PE-BMP: BIER PE BMP Assignments TLV

This TLV is advertised only for the <SD> for which the router has been elected to be D-BDR<SD> or BD-BDR<SD>. It can repeat multiple times.



Type TBD

MT-ID Multi topology for which the assignments are provided

subdomain-id subdomain-id for which the assignments are provided

AF identifies address family of the prefix for which the assignment is provided. Values TBD

Prefix Length Prefix length of the prefix for which the assignment is provided.

Prefix Prefix containing the identifying prefix from TLVs 235, 237, 135 or 236 for which the assignment is provided.

Assigned BFR-id Bit Mask Position assigned by D-BFR, set to invalid BMP on an error status. 2 octets.

E Bit indicating assignment error, i.e. the BFER does NOT have a valid assignment.

Status Status of the assignment, 3 bits.

0 Assignment is OK and can be used (based on either administratively requested BMP or chosen by D-BFR for the requesting BFER). E-bit MUST be clear.

1 error: Unresolvable collision with other administratively set values, Bit Mask Position cannot be used. E-bit MUST be set.

2 error: Out of Bit Mask Positions for the Topology and Set, Bit Mask Position cannot be used. E-bit MUST be set.

all other values reserved, MUST NOT be used.

The assignments SHOULD be sorted on BFER-ID. Assignments MUST NOT repeat when the TLV is advertised multiple times and a router discovering such condition MUST issue an adequate warning. When multiple assignments for the same BFR are found, the first one in first TLV MUST be used and all others disregarded.

The assignments MUST NOT repeat any BIER Info sub-TLVs that have the R and A bit cleared, e.g. purely administrative assignments. A router discovering such condition MUST issue an adequate warning and disregard such assignments.

The assignments MUST repeat all assigned BIER Info sub-TLVs (that have A bit set). When such assignment is not advertised anymore, the according BFER MUST interpret that as loss as assignment, i.e. start with R bit again or set the BFR-id to invalid BFR-id.

10. Security Considerations

Implementations must assure that malformed TLV and sub-TLV permutations do not result in errors which cause hard protocol failures.

11. Acknowledgements

TBD.

12. Normative References

[I-D.draft-przygienda-bier-isis-ranges-02]
Przygienda et al., A., "BIER support via ISIS", internet-draft draft-przygienda-bier-isis-ranges-02.txt, Jan 2015.

[I-D.draft-wijnands-bier-architecture-04]
Wijnands, IJ., "Stateless Multicast using Bit Index Explicit Replication Architecture", internet-draft draft-wijnands-bier-architecture-04.txt, February 2015.

[I-D.draft-wijnands-mpls-bier-encapsulation-02]
Wijnands et al., IJ., "Bit Index Explicit Replication using MPLS encapsulation", internet-draft draft-wijnands-mpls-bier-encapsulation-02.txt, December 2014.

[RFC1195] Callon, R., "Use of OSI IS-IS for routing in TCP/IP and dual environments", RFC 1195, December 1990.

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

[RFC2328] Moy, J., "OSPF Version 2", STD 54, RFC 2328, April 1998.

[RFC4971] Vasseur, JP., Shen, N., and R. Aggarwal, "Intermediate System to Intermediate System (IS-IS) Extensions for Advertising Router Information", RFC 4971, July 2007.

[RFC5120] Przygienda, T., Shen, N., and N. Sheth, "M-ISIS: Multi Topology (MT) Routing in Intermediate System to Intermediate Systems (IS-ISs)", RFC 5120, February 2008.

[RFC5305] Li, T. and H. Smit, "IS-IS Extensions for Traffic Engineering", RFC 5305, October 2008.

[RFC5308] Hopps, C., "Routing IPv6 with IS-IS", RFC 5308, October 2008.

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

Authors' Addresses

Tony Przygienda
Ericsson
300 Holger Way
San Jose, CA 95134
USA

Email: antoni.przygienda@ericsson.com

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

Email: jeff.tantsura@ericsson.com

Internet Engineering Task Force
Internet-Draft
Intended status: Standards Track
Expires: August 3, 2015

A. Przygienda
Ericsson
L. Ginsberg
Cisco Systems
S. Aldrin
Huawei
J. Zhang
Juniper Networks, Inc.

January 30, 2015

BIER support via ISIS
draft-przygienda-bier-isis-ranges-02

Abstract

Specification of an ISIS extension to support BIER domains and sub-domains.

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 RFC 2119 [RFC2119] .

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 http://datatracker.ietf.org/drafts/current/.

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

This Internet-Draft will expire on August 3, 2015.

Copyright Notice

Table of Contents

1. Introduction

Bit Index Explicit Replication (BIER) [I-D.draft-wijnands-bier-architecture-02] defines an architecture where all intended multicast receivers are encoded as bitmask in the Multicast packet header within different encapsulations such as [I-D.draft-wijnands-mpls-bier-encapsulation-02]. A router that receives such a packet will forward the packet based on the Bit Position in the packet header towards the receiver(s), following a precomputed tree for each of the bits in the packet. Each receiver is represented by a unique bit in the bitmask.

This document presents necessary extensions to the currently deployed ISIS for IP [RFC1195] protocol to support distribution of information necessary for operation of BIER domains and sub-domains. This document defines a new TLV to be advertised by every router participating in BIER signaling.

2. Terminology

Some of the terminology specified in [I-D.draft-wijnands-bier-architecture-02] is replicated here and extended by necessary definitions:

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

BIER-OL: BIER Overlay Signaling. (The method for the BFIR to learn about BFER's).

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

BFIR: Bit Forwarding Ingress Router (The ingress border router that inserts the BM into the packet).

BFER: Bit Forwarding Egress Router. A router that participates in Bit Index Forwarding as leaf. Each BFER must be a BFR. Each BFER must have a valid BFR-id assigned.

BFT: Bit Forwarding Tree used to reach all BFERs in a domain.

BIFT: Bit Index Forwarding Table.

BMS: Bit Mask Set. Set containing bit positions of all BFER participating in a set.

BMP: Bit Mask Position, a given bit in a BMS.

Invalid BMP: Unassigned Bit Mask Position, consisting of all 0s.

IGP signalled BIER domain: A BIER underlay where the BIER synchronization information is carried in IGP. Observe that a multi-topology is NOT a separate BIER domain in IGP.

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

BFR-id: An optional, unique identifier for a BFR within a BIER sub-domain.

Invalid BFR-id: Unassigned BFR-id, consisting of all 0s.

3. IANA Considerations

This document adds the following new sub-TLVs to the registry of sub-TLVs for TLVs 235, 237 [RFC5120] and TLVs 135,236 [RFC5305],[RFC5308].

Value: 32 (suggested - to be assigned by IANA)

Name: BIER Info

4. Concepts

4.1. BIER Domains and Sub-Domains

An ISIS signalled BIER domain is aligned with the scope of distribution of BFR-prefixes that identify the BFRs within ISIS. ISIS acts in such a case as the according BIER underlay.

Within such a domain, ISIS extensions are capable of carrying BIER information for multiple BIER sub-domains. Each sub-domain is uniquely identified by its subdomain-id and each subdomain can reside in any of the ISIS topologies [RFC5120]. The mapping of sub-domains to topologies is a local decision of each BFR currently but is advertised throughout the domain to ensure routing consistency.

Each BIER sub-domain has as its unique attributes the encapsulation used and the type of tree it is using to forward BIER frames (currently always SPF). Additionally, per supported bitstring length in the sub-domain, each router will advertise the necessary label ranges to support it.

This RFC introduces a sub-TLV in the extended reachability TLVs to distribute such information about BIER sub-domains. To satisfy the requirements for BIER prefixes per [I-D.draft-wijnands-bier-architecture-02] additional information will be carried in [I-D.draft-ginsberg-isis-prefix-attributes].

5. Procedures

5.1. Enabling a BIER Sub-Domain

A given sub-domain with identifier BS with supported bitstring lengths MLs in a multi-topology MT [RFC5120] is denoted further as <MT,SD,MLs> and is normally not advertised to preserve the scaling of the protocol (i.e. ISIS carries no TLVs containing any of the elements related to <MT,SD>) and is enabled by a first BIER sub-TLV (Section 6.1) containing <MT,SD> being advertised into the area. The trigger itself is outside the scope of this RFC but can be for example a VPN desiring to initiate a BIER sub-domain as MI-PMSI [RFC6513] tree. It is outside the scope of this document to describe what trigger for a router capable of participating in <MT,SD> is used to start the origination of the necessary information to join into it.

5.2. Multi Topology and Sub-Domain

All routers in the flooding scope of the BIER TLVs MUST advertise a sub-domain within the same multi-topology. A router discovering a sub-domain advertised within a topology that is different from its own MUST report a misconfiguration of a specific sub-domain. Each router MUST compute BFTs for a sub-domain using only routers advertising it in the same topology.

5.3. Encapsulation

All routers in the flooding scope of the BIER TLVs MUST advertise the same encapsulation for a given <MT,SD>. A router discovering encapsulation advertised that is different from its own MUST report a misconfiguration of a specific <MT,SD>. Each router MUST compute BFTs for <MT,SD> using only routers having the same encapsulation as its own advertised encapsulation in BIER sub-TLV for <MT,SD>.

5.4. Tree Type

All routers in the flooding scope of the BIER TLVs MUST advertise the same tree type for a given <MT,SD>. In case of mismatch the behavior is analogous to Section 5.3.

5.5. Label Advertisements for MPLS encapsulated BIER sub-domains

Each router MAY advertise within the BIER MPLS Encapsulation sub-sub-TLV (Section 6.2) of a BIER Info sub-TLV (Section 6.1, denoted as TLV<MT,SD>) for <MT,SD> for every supported bitstring length a valid starting label value and a non-zero range length. It MUST advertise at least one valid label value and a non-zero range length for the required bitstring lengths per [I-D.draft-wijnands-bier-architecture-02] in case it has computed

itself as being on the BFT rooted at any of the BFRs with valid BFR-ids (except itself if it does NOT have a valid BFR-id) participating in <MT,SD>.

A router MAY decide to not advertise the BIER Info sub-TLV (Section 6.1) for <MT,SD> if it does not want to participate in the sub-domain due to resource constraints, label space optimization, administrative configuration or any other reasons.

5.5.1. Special Consideration

A router MUST advertise for each bitstring length it supports in <MT,SD> a label range size that guarantees to cover the maximum BFR-id injected into <MT,SD> (which implies a certain maximum set id per bitstring length as described in [I-D.draft-wijnands-bier-architecture-02]). Any router that violates this condition MUST be excluded from BIER BFTs for <MT,SD>.

5.6. BFR-id Advertisements

Each BFER MAY advertise with its TLV<MT,SD> the BFR-id that it has administratively chosen.

If a router discovers that two BFRs it can reach advertise the same value for BFR-id for <MT,SD>, it MUST report a misconfiguration and disregard those routers for all BIER calculations and procedures for <MT,SD> to align with [I-D.draft-wijnands-bier-architecture-02]. It is worth observing that based on this procedure routers with colliding BFR-id assignments in <MT,SD> MAY still act as BFIRs in <MT,SD> but will be never able to receive traffic from other BFRs in <MT,SD>.

5.7. Flooding

BIER domain information SHOULD change and force flooding infrequently. Especially, the router SHOULD make every possible attempt to bundle all the changes necessary to sub-domains and ranges advertised with those into least possible updates.

5.8. Version

This RFC specifies Version 0 of the BIER extension encodings. Packet encoding supports introduction of future, higher versions with e.g. new sub-sub-TLVs or redefining reserved bits that can maintain the compatiblity to Version 0 or choose to indicate that the compatibility cannot be maintained anymore (changes that cannot work with the provided encoding would necessitate obviously introduction of completely new sub-TLV for BIER).

This kind of 'versioning' allows to introduce e.g. backwards-compatible automatic assignment of unique BFR-ids within sub-domains or addition of optional sub-sub-TLVs that can be ignored by version 0 BIER routers without the danger of incompatiblity.

This is a quite common technique in software development today to maintain and extend backwards compatible APIs.

6. Packet Formats

All ISIS BIER information is carried within the TLVs 235, 237 [RFC5120] and TLVs 135,236 [RFC5305], [RFC5308].

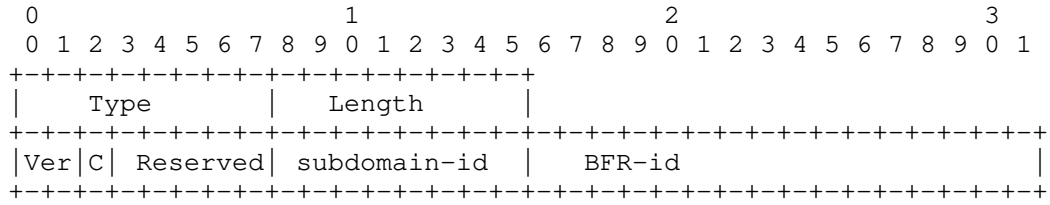
6.1. BIER Info sub-TLV

This sub-TLV carries the information for the BIER sub-domains that the router participates in as BFR. It can repeat multiple times for different sub-domain <MT,SD> combinations.

The sub-TLV carries a single <MT,SD> combination followed by optional sub-sub-TLVs specified within its context such as e.g. BIER MPLS Encapsulation per Section 6.2.

On violation of any of the following conditions, the receiving router SHOULD signal a misconfiguration condition. Further results are unspecified unless described in the according section of this RFC:

- The subdomain-id MUST be included only within a single topology.



Type: as indicated in IANA section.

Length: 1 octet.

Version: Version of the BIER TLV advertised, must be 0 on transmission by router implementing this RFC. Behavior on reception depends on the 'C' bit. 2 bits

C-BIT: Compatibility bit indicating that the TLV can be interpreted by routers implementing lower than the advertised version. Router implementing this version of the RFC MUST set it to 1. On reception, IF the version of the protocol is higher than 0 AND the bit is set (i.e. its value is 1), the TLV MUST be processed normally, IF the bit is clear (i.e. its value is 0), the TLV MUST be ignored for further processing completely independent of the advertised version. When processing this sub-TLV with compatibility bit set, all sub-sub-TLV of unknown type MUST and CAN be safely ignored. 1 bit

Reserved: reserved, must be 0 on transmission, ignored on reception. May be used in future versions. 5 bits

subdomain-id: Unique value identifying the BIER sub-domain. 1 octet

BFR-id: A 2 octet field encoding the BFR-id, as documented in [I-D.draft-wijnands-bier-architecture-02]. If set to the invalid BFR-id advertising router is not owning a BFR-id in the sub-domain.

6.2. BIER MPLS Encapsulation sub-sub-TLV

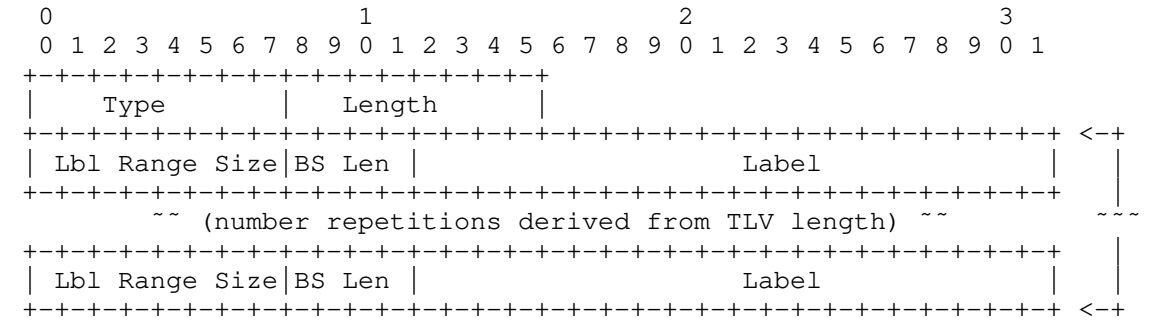
This sub-sub-TLV carries the information for the BIER MPLS encapsulation and the necessary label ranges per bitstring length for a certain <MT,SD> and is carried within the BIER Info sub-TLV (Section 6.1) that the router participates in as BFR.

On violation of any of the following conditions, the receiving router SHOULD signal a misconfiguration condition. Further results are unspecified:

- The sub-sub-TLV MUST be included once AND ONLY once within the sub-TLV.
- Label ranges within the sub-sub-TLV MUST NOT overlap. A receiving BFR MAY additionally check whether any of the ranges in all the sub-sub-TLVs advertised by another BFR overlap and apply the same treatement on violations.
- Bitstring lengths within the sub-sub-TLV MUST NOT repeat.
- The sub-sub-TLV MUST include the required bitstring lengths per [I-D.draft-wijnands-bier-architecture-02].

- All label range sizes MUST be greater than 0.

- All labels MUST represent valid label values.



Type: value of 0 indicating MPLS encapsulation.

Length: 1 octet.

Local BitString Length (BS Len): Bitstring length for the label range that this router is advertising per [I-D.draft-wijnands-mpls-bier-encapsulation-02]. 4 bits.

Label Range Size: Number of labels in the range used on encapsulation for this BIER sub-domain for this bitstring length, 1 octet. This MUST never be advertised as 0 (zero) and otherwise, this sub-sub-TLV must be treated as if not present for BFT calculations and a misconfiguration SHOULD be reported by the receiving router.

Label: First label of the range used on encapsulation for this BIER sub-domain for this bitstring length, 20 bits. The label is used for example by [I-D.draft-wijnands-mpls-bier-encapsulation-02] to forward traffic to sets of BFERs.

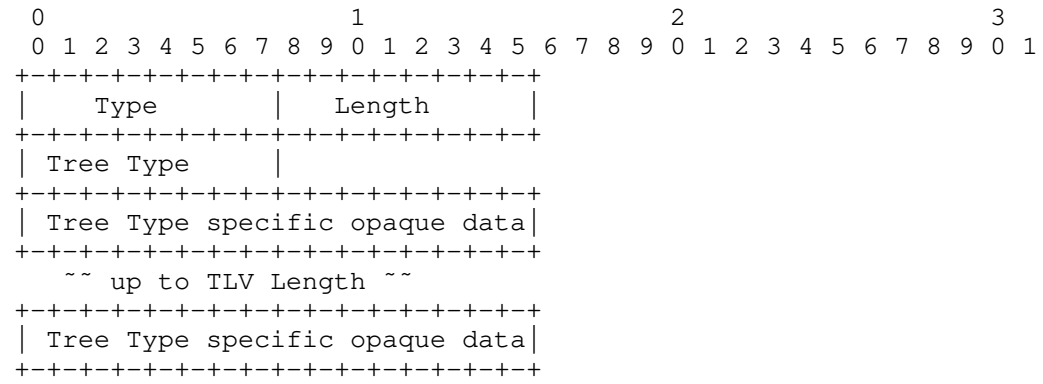
6.3. Optional BIER sub-domain Tree Type sub-sub-TLV

This sub-sub-TLV carries the information of the BIER tree type for a certain <MT,SD>. It is carried within the BIER Info sub-TLV (Section 6.1) that the router participates in as BFR. This sub-sub-TLV is optional and its absence indicates the same as its presence

with Tree Type value 0 (SPF). BIER implementation following this version of the RFC SHOULD NOT advertise this TLV.

On violation of any of the following conditions, the receiving router implementing this RFC SHOULD signal a misconfiguration condition. Further results are unspecified unless described further:

- The sub-sub-TLV MUST be included once AND ONLY once.
- The advertised BIER TLV version is 0 and the value of Tree Type MUST be 0 (SPF).



Type: value of 1 indicating BIER Tree Type.

Length: 1 octet.

Tree Type: The only supported value today is 0 and indicates that BIER uses normal SPF computed reachability to construct BIFT. BIER implementation following this RFC MUST ignore the node for purposes of the sub-domain <MT,SD> if this field has any value except 0.

Tree type specific opaque data: Opaque data up to the length of the TLV carrying tree type specific parameters. For Tree Type 0 (SPF) no such data is included and therefore TLV Length is 1.

7. Security Considerations

Implementations must assure that malformed TLV and Sub-TLV permutations do not result in errors which cause hard protocol failures.

8. Acknowledgements

The RFC is aligned with the [I-D.draft-psenak-ospf-bier-extension-01] draft as far as the protocol mechanisms overlap.

Many thanks for comments from (in no particular order) Hannes Gredler, Ijsbrand Wijnands and Peter Psenak.

9. Normative References

[I-D.draft-ginsberg-isis-prefix-attributes]
Ginsberg et al., U., "IS-IS Prefix Attributes for Extended IP and IPv6 Reachability", internet-draft draft-ginsberg-isis-prefix-attributes-00.txt, October 2014.

[I-D.draft-psenak-ospf-bier-extension-01]
Psenak, P. and IJ. Wijnands, "OSPF Extension for Bit Index Explicit Replication", internet-draft draft-ietf-ospf-prefix-link-attr-01.txt, October 2014.

[I-D.draft-wijnands-bier-architecture-02]
Wijnands, IJ., "Stateless Multicast using Bit Index Explicit Replication Architecture", internet-draft draft-wijnands-bier-architecture-02.txt, February 2014.

[I-D.draft-wijnands-mpls-bier-encapsulation-02]
Wijnands et al., IJ., "Bit Index Explicit Replication using MPLS encapsulation", internet-draft draft-wijnands-mpls-bier-encapsulation-02.txt, February 2014.

[RFC1195] Callon, R., "Use of OSI IS-IS for routing in TCP/IP and dual environments", RFC 1195, December 1990.

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

[RFC4971] Vasseur, JP., Shen, N., and R. Aggarwal, "Intermediate System to Intermediate System (IS-IS) Extensions for Advertising Router Information", RFC 4971, July 2007.

[RFC5120] Przygienda, T., Shen, N., and N. Sheth, "M-ISIS: Multi Topology (MT) Routing in Intermediate System to Intermediate Systems (IS-ISs)", RFC 5120, February 2008.

[RFC5305] Li, T. and H. Smit, "IS-IS Extensions for Traffic Engineering", RFC 5305, October 2008.

[RFC5308] Hopps, C., "Routing IPv6 with IS-IS", RFC 5308, October 2008.

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

Authors' Addresses

Tony Przygienda
Ericsson
300 Holger Way
San Jose, CA 95134
USA

Email: antoni.przygienda@ericsson.com

Les Ginsberg
Cisco Systems
510 McCarthy Blvd.
Milpitas, CA 95035
USA

Email: ginsberg@cisco.com

Sam Aldrin
Huawei
2330 Central Expressway
Santa Clara, CA 95051
USA

Email: aldrin.ietf@gmail.com

Jeffrey (Zhaohui) Zhang
Juniper Networks, Inc.
10 Technology Park Drive
Westford, MA 01886
USA

Email: zzhang@juniper.net

OSPF
Internet-Draft
Intended status: Standards Track
Expires: August 29, 2015

P. Psenak, Ed.
N. Kumar
IJ. Wijnands
Cisco
A. Dolganow
Alcatel-Lucent
T. Przygienda
Ericsson
J. Zhang
Juniper Networks, Inc.
S. Aldrin
Huawei Technologies
February 25, 2015

OSPF Extensions For BIER
draft-psenak-ospf-bier-extensions-02.txt

Abstract

Bit Index Explicit Replication (BIER) is an architecture that provides optimal multicast forwarding through a "BIER domain" without requiring intermediate routers to maintain any multicast related per-flow state. BIER also does not require any explicit tree-building protocol for its operation. 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). The BFIR router adds a BIER header to the packet. The BIER header contains a bit-string in which each bit represents exactly one BFER to forward the packet to. The set of BFERs to which the multicast packet needs to be forwarded is expressed by setting the bits that correspond to those routers in the BIER header.

This document describes the OSPF protocol extension required for BIER with MPLS encapsulation.

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 http://datatracker.ietf.org/drafts/current/.

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

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

This Internet-Draft will expire on August 29, 2015.

Table of Contents

1. Introduction

Bit Index Explicit Replication (BIER) is an architecture that provides optimal multicast forwarding through a "BIER domain" without requiring intermediate routers to maintain any multicast related per-flow state. Neither does BIER explicitly require a tree-building protocol for its operation. 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). The BFIR router adds a BIER header to the packet. The BIER header contains a bit-string in which each bit represents exactly one BFER to forward the packet to. The set of BFERs to which the multicast packet needs to be forwarded is expressed by setting the bits that correspond to those routers in the BIER header.

BIER architecture requires routers participating in BIER within a given BIER domain to exchange some BIER specific information among themselves. BIER architecture allows link-state routing protocols to perform the distribution of these information. In this document we describe extensions to OSPF to distribute BIER specific information for the case where BIER uses MPLS encapsulation as described in [I-D.wijnands-mpls-bier-encapsulation].

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].

2. Flooding of the BIER Information in OSPF

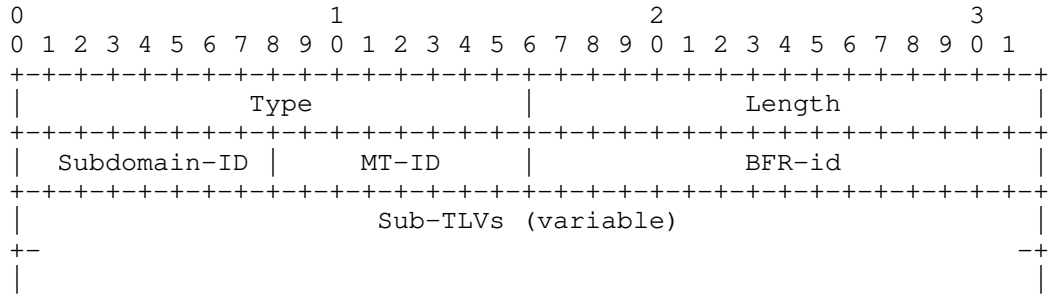
All the BIER specific information that a BIER router needs to advertise to other BIER routers are associated with the BFR-Prefix, a unique (within a given BIER domain), routable IP address that is assign to each BIER router as described in section 2 of [I-D.wijnands-bier-architecture].

Given that the BIER information is associated with the prefix, the OSPF Extended Prefix Opaque LSA [I-D.ietf-ospf-prefix-link-attr] is used to flood BIER related information.

2.1. The BIER Sub-TLV

A new Sub-TLV of the Extended Prefix TLV (defined in [I-D.ietf-ospf-prefix-link-attr]) is defined for distributing BIER information. The new Sub-TLV is called BIER Sub-TLV. Multiple BIER Sub-TLVs may be included in the Extended Prefix TLV.

BIER Sub-TLV has the following format:



Type: TBD

Length: 4 bytes

Subdomain-ID: Unique value identifying the BIER subdomain within the BIER domain, as described in section 1 of [I-D.wijnands-bier-architecture].

MT-ID: Multi-Topology ID (as defined in [RFC4915]) that identifies the topology that is associated with the BIER sub-domain.

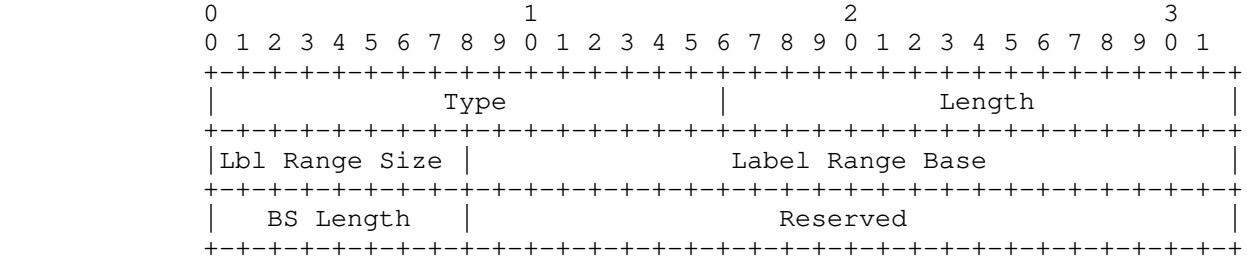
BFR-id: A 2 octet field encoding the BFR-id, as documented in section 2 [I-D.wijnands-bier-architecture]. If the BFR-id is zero, it means, the advertising router is not advertising any BIER-id.

Each BFR sub-domain MUST be associate with a single OSPF topology that is identified by the MT-ID. If the association between BEIR sub-domain and OSPF topology advertised in the BIER sub-TLV is in conflict with the association locally configured on the receiving router, BIER sub-TLV SHOULD be ignored.

2.2. The BIER MPLS Encapsulation Sub-TLV

BIER MPLS Encapsulation Sub-TLV is a sub-TLV of the BIER Sub-TLV. BIER MPLS Encapsulation Sub-TLVt is used in order to advertise MPLS specific information used for BIER. It MAY appear multiple times in the BIER Sub-TLV.

BIER MPLS Encapsulation Sub-TLV has the following format:



Type: TBD

Length: 4 bytes

Label Range Size: A 1 octet field encoding the label range size of the label range. It MUST be greater then 0, otherwise the TLV MUST be ignored.

Label Range Base: A 3 octet field, where the 20 rightmost bits represent the first label in the label range.

BS Length: A 1 octet field encoding the supported BitString length associated with this BFR-prefix. The values allowed in this field are specified in section 3 of [I-D.wijnands-mpls-bier-encapsulation].

The "label range" is the set of labels beginning with the label range base and ending with (label range base)+(label range size)-1. A unique label range is allocated for each BitStream length and Multi-Topology ID. These labels are used for BIER forwarding as described in [I-D.wijnands-bier-architecture] and [I-D.wijnands-mpls-bier-encapsulation].

The size of the label range is determined by the number of Set Identifiers (SI) (section 2 of [I-D.wijnands-bier-architecture]) that are used in the network. Each SI maps to a single label in the label range. The first label is for SI=0, the second label is for SI=1, etc.

If same BS length is repeated in multiple BIER MPLS Encapsulation Sub-TLV inside the same BIER Sub-TLV, the first BIER MPLS Encapsulation Sub-TLV with such BS length MUST be used and any subsequent BIER MPLS Encapsulation Sub-TLVs with the same BS length MUST be ignored.

Label ranges within all BIER MPLS Encapsulation Sub-TLV inside the same BIER Sub-TLV SHOULD NOT overlap. If the overlap is detected, overlapping BIER MPLS Encapsulation Sub-TLV SHOULD be ignored.

2.3. Flooding scope of BIER Information

Flooding scope of the OSPF Extended Prefix Opaque LSA [I-D.ietf-ospf-prefix-link-attr] that is used for advertising BIER Sub TLV is set to area. If (and only if) a single BIER domain contains multiple OSPF areas, OSPF must propagate BIER information between areas. The following procedure is used in order to propagate BIER related information between areas:

When an OSPF ABR advertises a Type-3 Summary LSA from an intra-area or inter-area prefix to all its connected areas, it will also originate an Extended Prefix Opaque LSA, as described in [I-D.ietf-ospf-prefix-link-attr]. The flooding scope of the Extended Prefix Opaque LSA type will be set to area-scope. The route-type in the OSPF Extended Prefix TLV is set to inter-area. When determining whether a BIER Sub-TLV should be included in this LSA ABR will:

- look at its best path to the prefix in the source area and find the advertising router associated with the best path to that prefix.

- determine if such advertising router advertised a BIER Sub-TLV for the prefix. If yes, ABR will copy the information from such BIER MPLS Sub-TLV when advertising BIER MPLS Sub-TLV to each connected area.

3. Security Considerations

Implementations must assure that malformed TLV and Sub-TLV permutations do not result in errors which cause hard OSPF failures.

4. IANA Considerations

The document requests two new allocations from the OSPF Extended Prefix sub-TLV registry as defined in [I-D.ietf-ospf-prefix-link-attr].

BIER Sub-TLV: TBD

BIER MPLS Encapsulation Sub-TLV: TBD

5. Acknowledgments

The authors would like to thank Rajiv Asati, Christian Martin, Greg Shepherd and Eric Rosen for their contribution.

6. Normative References

[I-D.ietf-ospf-prefix-link-attr]
Psenak, P., Gredler, H., Shakir, R., Henderickx, W., Tantsura, J., and A. Lindem, "OSPFv2 Prefix/Link Attribute Advertisement", draft-ietf-ospf-prefix-link-attr-03 (work in progress), February 2015.

[I-D.wijnands-bier-architecture]
Wijnands, I., Rosen, E., Dolganow, A., and T. Przygienda, "Multicast using Bit Index Explicit Replication", draft-wijnands-bier-architecture-00 (work in progress), September 2014.

[I-D.wijnands-mpls-bier-encapsulation]
Wijnands, I., Rosen, E., Dolganow, A., and J. Tantsura, "Encapsulation for Bit Index Explicit Replication in MPLS Networks", draft-wijnands-mpls-bier-encapsulation-00 (work in progress), September 2014.

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

[RFC4915] Psenak, P., Mirtorabi, S., Roy, A., Nguyen, L., and P. Pillay-Esnault, "Multi-Topology (MT) Routing in OSPF", RFC 4915, June 2007.

Authors' Addresses

Peter Psenak (editor)
Cisco
Apollo Business Center
Mlynske nivy 43
Bratislava 821 09
Slovakia

Email: ppsenak@cisco.com

Nagendra Kumar
Cisco
7200 Kit Creek Road
Research Triangle Park, NC 27709
US

Email: naikumar@cisco.com

IJsbrand Wijnands
Cisco
De Kleetlaan 6a
Diegem 1831
Belgium

Email: ice@cisco.com

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

Email: andrew.dolganow@alcatel-lucent.com

Tony Przygienda
Ericsson
300 Holger Way
San Jose, CA 95134
USA

Email: antoni.przygienda@ericsson.com

Jeffrey Zhang
Juniper Networks, Inc.
10 Technology Park Drive
Westford, MA 01886
USA

Email: zzhang@juniper.net

Sam Aldrin
Huawei Technologies
2330 Central Expressway
Santa Clara, CA 95051
USA

Email: zzhang@juniper.net

Internet Engineering Task Force
Internet-Draft
Intended status: Standards Track
Expires: August 6, 2015

IJ. Wijnands, Ed.
Cisco Systems, Inc.
E. Rosen, Ed.
Juniper Networks, Inc.
A. Dolganow
Alcatel-Lucent
T. Przygienda
Ericsson
S. Aldrin
Huawei Technologies
February 2, 2015

Multicast using Bit Index Explicit Replication
draft-wijnands-bier-architecture-04

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

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 http://datatracker.ietf.org/drafts/current/.

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

This Internet-Draft will expire on August 6, 2015.

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. The BIER control plane protocols (see Section 4.2) run within a BIER domain, allowing the BFRs within that domain to exchange the necessary information.

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.

A BIER domain may contain one or more sub-domains. Each BIER domain MUST contain at least one sub-domain, the "default sub-domain" (also denoted "sub-domain zero"). If a BIER domain contains more than one sub-domain, each BFR in the domain MUST be provisioned to know the set of sub-domains to which it belongs. Each sub-domain is identified by a sub-domain-id in the range [0,255].

For each sub-domain to which a given BFR belongs, if the BFR is capable of acting as a BFIR or a BFER, it MUST be provisioned with a "BFR-id" that is unique within the sub-domain. A BFR-id is a small unstructured number. For instance, if a particular BIER sub-domain contains 1,374 BFRs, each one could be given a BFR-id in the range 1-1374.

If a given BFR belongs to more than one sub-domain, it may (though it need not) have a different BFR-id for each sub-domain.

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. The BFIR also determines the sub-domain over which the packet must be sent. (Procedures for assigning a particular packet to a particular sub-domain are outside the scope of this document.) The BFIR 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 in the sub-domain to which the packet has been assigned. 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 sub-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:

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

- 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 (in the sub-domain to which the packet has been assigned) 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 has been assigned to sub-domain 0, and needs to be delivered to three BFERs, where those BFERs have BFR-ids in sub-domain 0 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 unique and routable within the BIER 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. Note that a BFR in a given BIER domain has the same BFR-prefix in all the sub-domains of that BIER domain.

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

Note that the value 0 is not a legal BFR-id.

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 for each sub-domain 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. Each BFIR in a given BIER domain MUST be provisioned to know the

BitStringLength to use when imposing a BIER encapsulation on a particular set of packets. This value of BitStringLength SHOULD be a value that is supported by all the BFRs in the domain. (That is, the BitStringLength value used by a BFIR when imposing a BIER encapsulation on a particular packet SHOULD be a value that is supported by all the BFRs and BFERs in the domain that might have to forward or receive the packet.) However, under certain circumstances, it is possible to make exceptions to this rule. This is discussed in Section 6.10.

Every BFIR MUST be able to impose a BIER encapsulation whose BitStringLength of 256. Every BFR MUST be able to forward a BIER-encapsulated packet whose BitStringLength is 256. Every BFER MUST be able to receive and properly process a BIER-encapsulated packet whose BitStringLength is 256.

Particular BIER 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]. ("Supporting the values in a given range" means, in this context, that any value in the given range is legal, and will be properly interpreted.)

When a BFIR determines that a multicast data packet, assigned to a given sub-domain, needs to be forwarded to a particular set of destination BFERs, the BFIR partitions that set of BFERs into subsets, where each subset contains the target BFERs whose BFR-ids in the given sub-domain 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. Of course, in order to properly interpret the BitString, it must be possible to infer the sub-domain-id from the encapsulation as well.

Suppose, for example, that a BFIR determines that a given packet needs to be forwarded to three BFERs, whose BFR-ids (in the appropriate sub-domain) 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. 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 possible that several different forms of BIER encapsulation will be developed. If so, 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(s) 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 default topology that the Interior Gateway Protocol (IGP), e.g., OSPF, uses 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. (Each topology would be considered to be a different underlay.) Alternatively, one could deploy a routing underlay that creates a multicast-specific tree of some sort, perhaps a Steiner tree. Then BIER could be used to forward multicast data packets along the multicast-specific 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 is being used for that packet.

If multiple routing underlays are used in a single BIER domain, each BIER sub-domain MUST be associated with a single routing underlay. (Though multiple sub-domains may be associated with the same routing underlay.) A BFR that belongs to multiple sub-domains MUST be provisioned to know which routing underlay is used by each sub-domain. By default (i.e., in the absence of any provisioning to the contrary), each sub-domain uses the default topology of the unicast IGP as the routing underlay.

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:

- Protocols and procedures that advertise, to all other BFRs in the same BIER domain, each BFR's BFR-prefix.

- Protocols and procedures that advertise, to all other BFRs in the same BIER domain, each BFR's BFR-id for each sub-domain.

- The imposition by a BFIR of a BIER header on a multicast data packet.

- 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.

- 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.

- 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 sub-domain). Each BFER must advertise these assignments 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 supported values 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-prefix, <sub-domain-id,BFR-id> 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] and [ISIS_BIER_EXTENSIONS].)

These advertisements enable each BFR to associate a given <sub-domain-id, 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 each sub-domain) BFR-id, two different BFRs will not advertise ownership of the same <sub-domain-id, BFR-id> unless there has been a provisioning error.

- If BFR-A determines that BFR-B and BFR-C have both advertised the same BFR-id for the same sub-domain, 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 that has been assigned to the given sub-domain, 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 in the given sub-domain until the provisioning error is fixed. However, that is preferable to having them receive each other's traffic.

- If BFR-A has been provisioned with BFR-id N for a particular sub-domain, has not yet advertised its ownership of BFR-id N for that sub-domain, but has received an advertisement from a different BFR (say BFR-B) that is advertising ownership of BFR-id N for the same sub-domain, then BFR-A SHOULD log an error, and MUST NOT advertise its own ownership of BFR-id N for that sub-domain as long as the advertisement from BFR-B is extant.

  This procedure may prevent the accidental misconfiguration of a new BFR from impacting an existing BFR.

If a BFR advertises that it has a BFR-id of 0 in a particular sub-domain, other BFRs receiving the advertisement MUST interpret that advertisement as meaning that the advertising BFR does not have a BFR-id in that sub-domain.

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 sub-domain.

2. Determine the packet's SI.

3. From the sub-domain, the SI and the BitString, determine the set of destination BFERs for the packet.

4. Using information provided by the routing underlay associated with the packet's sub-domain, determine the next hop adjacency for each of the destination BFERs.

5. 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.

6. 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 sub-domain, 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 sub-domain of the packet, and the BFR-ids (in that sub-domain) 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 (associated with the packet's sub-domain) 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-B. BFR-A must then determine the BFR-prefix of BFR-B. (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.

Under certain circumstances, a BFR may have adjacencies (in a particular routing underlay) that are not BFRs. Please see Section 6.9 for a discussion of how to handle those circumstances.

6.3. The Bit Index Routing Table

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

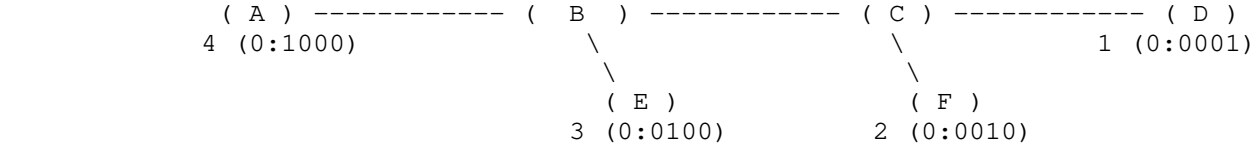


Figure 1: BIER Topology 1

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.)

Note that a BIRT is specific to a particular BIER sub-domain.

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. (Note that a BIFT is specific to a particular sub-domain.)

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 been assigned to the default sub-domain, 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:

- 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.

- 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).

- The copy is now sent to BFR-B.

- 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).

- 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:

- 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.

- 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).

- The copy is now sent to BFR-B.

- 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).

- 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:

- 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.

- 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).

- The copy is now sent to BFR-C.

- 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).

- 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.

- 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).

- The copy is now sent to BFR-E.

- 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. Two ECMP procedures are defined. In the first (described in Section 6.7.1), the choice among equal-cost paths taken by a given packet from a given BFR to a given BFER depends on (a) the packet's entropy, and (b) the other BFERs to which that packet is destined. In the second (described in Section 6.7.2), the choice depends only upon the packet's entropy.

There are tradeoffs between the two forwarding procedures described here. In the procedure of Section 6.7.1, the number of packet replications is minimized. The procedure in Section 6.7.1 also uses less memory in the BFR. In the procedure of Section 6.7.2, the path traveled by a given packet from a given BFR to a given BFER is independent of the other BFERs to which the packet is destined. While the procedures of Section 6.7.2 may cause more replications, they provide a more predictable behavior.

The two procedures described here operate on identical packet formats and will interoperate correctly. However, if deterministic behavior is desired, then all BFRs would need to use the procedure from Section 6.7.2.

6.7.1. Non-deterministic ECMP

Figure 6 shows the operation of non-deterministic ECMP in BIER.

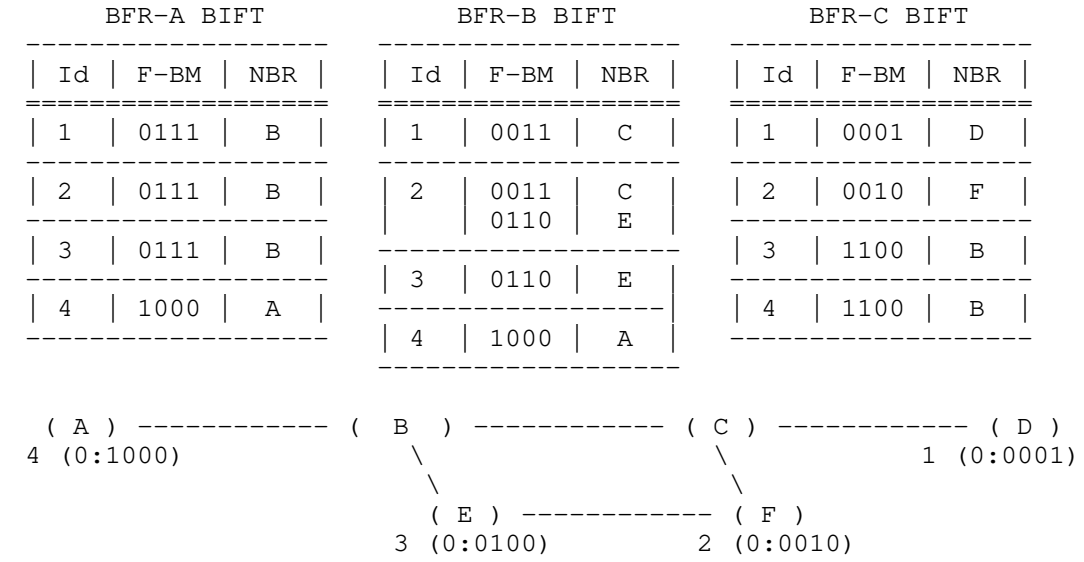


Figure 6: Example of ECMP

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.

6.7.2. Deterministic ECMP

With the procedures of Section 6.7.1, where ECMP paths exist, the path a packet takes to reach any particular BFER depends not only on routing and on the packet's entropy, but also on the set of other BFERs to which the packet is destined.

For example consider the following scenario in the network of Figure 6.

- There is a sequence of packets being transmitted by BFR-A, some of which are destined for both D and F, and some of which are destined only for F.
- All the packets in this sequence have the same entropy value, call it "Q".
- At BFR-B, when a packet with entropy value Q is forwarded via entry 2 in the BIFT, the packet is sent to E.

Using the forwarding procedure of Section 6.7.1, packets of this sequence that are destined for both D and F are forwarded according to entry 1 in the BIFT, and thus will reach F via the path A-B-C-F. However, packets of this sequence that are destined only for F are forwarded according to entry 2 in the BIFT, and thus will reach F via the path A-B-E-F.

That procedure minimizes the number of packets transmitted by BFR B. However, consider the following scenario:

- Beginning at time t0, the multicast flow in question needs to be received ONLY by BFER-F;
- Beginning at a later time, t1, the flow needs to be received by both BFER-D and BFER-F.
- Beginning at a later time, t2, the no longer needs to be received by D, but still needs to be received by F.

Then from t0 until t1, the flow will travel to F via the path A-B-E-F. From t1 until t2, the flow will travel to F via the path A-B-C-F. And from t2, the flow will again travel to F via the path A-B-E-F.

The problem is that if D repeatedly joins and leaves the flow, the flow's path from B to F will keep switching. This could cause F to receive packets out of order. It also makes troubleshooting difficult. For example, if there is some problem on the E-F link, receivers at F will get good service when the flow is also going to D (avoiding the E-F link), but bad service when the flow is not going to D. Since it is hard to know which path is being used at any given time, this may be hard to troubleshoot. Also, it is very difficult to perform a traceroute that is known to follow the path taken by the flow at any given time.

The source of this difficulty is that, in the procedures of Section 6.7.1, the path taken by a particular flow to a particular BFER depends upon whether there are lower numbered BFERs that are also receiving the flow. Thus the choice among the ECMP paths is fundamentally non-deterministic.

Deterministic forwarding can be achieved by using multiple BIFTs, such that each row in a BIFT has only one path to each destination, but the multiple ECMP paths to any particular destination are spread across the multiple tables. When a BIER-encapsulated packet arrives to be forwarded, the BFR uses a hash of the BIER Entropy field to determine which BIFT to use, and then the normal BIER forwarding algorithm (as described in Sections 6.5 and 6.6) is used with the selected BIFT.

As an example, suppose there are two paths to destination X (call them X1 and X2), and four paths to destination Y (call them Y1, Y2, Y3, and Y4). If there are, say, four BIFTs, one BIFT would have paths X1 and Y1, one would have X1 and Y2, one would have X2 and Y3, and one would have X2 and Y4. If traffic to X is split evenly among these four BIFTs, the traffic will be split evenly between the two paths to X; if traffic to Y is split evenly among these four BIFTs, the traffic will be split evenly between the four paths to Y.

Note that if there are three paths to one destination and four paths to another, 12 BIFTs would be required in order to get even splitting of the load to each of those two destinations. Of course, each BIFT uses some memory, and one might be willing to have less optimal splitting in order to have fewer BIFTs. How that tradeoff is made is an implementation or deployment decision.

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.

6.9. When Some Nodes do not Support BIER

The procedures of section Section 6.2 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, as long as the ingress and egress nodes are BFRs. In this section, we assume that the routing underlay is an SPF-based IGP that computes a shortest path tree from each node to all other nodes in the domain.

At a given BFR, say BFR B, start with a copy of the IGP-computed shortest path tree from BFR B to each router in the domain. (This tree is computed by the SPF algorithm of the IGP.) Let's call this copy the "BIER-SPF tree rooted at BFR B." BFR B then modifies this BIER-SPF tree as follows.

- BFR B looks in turn at each of B's child nodes on the BIER-SPF tree.
- If one of the child nodes does not support BIER, BFR B removes that node from the tree. The child nodes of the node that has just been removed are then re-parented on the tree, so that BFR B now becomes their parent.

- BFR B then continues to look at each of its child nodes, including any nodes that have been re-parented to B as a result of the previous step.

When all of the child nodes (the original child nodes plus any new ones) have been examined, B's children on the BIER-SPF tree will all be BFRs.

When the BIFT is constructed, B's child nodes on the BIER-SPF tree are considered to be the BFR-NBRs. The F-BMs and outgoing BIER-MPLS labels must be computed appropriately, based on the BFR-NBRs.

B may now have BFR-NBRs that are not "directly connected" to B via layer 2. To send a packet to one of these BFR-NBRs, B will have to send the packet through a unicast tunnel. This may be as simple as finding the IGP unicast next hop to the child node, and pushing on (above the BIER-MPLS label advertised by the child) the MPLS label that the IGP next hop has bound to an address of the child node.

Of course, the above is not meant as an implementation technique, just as a functional description.

While the above description assumes that the routing underlay provides an SPF tree, it may also be applicable to other types of routing underlay.

Note that the technique above can also be used to provide "node protection" (i.e., to provide fast reroute around nodes that are believed to have failed). If BFR B has a failed BFR-NBR, B can remove the failed BFR-NBR from the BIER-SPF tree, and can then re-parent the child BFR-NBRs of the failed BFR-NBR so that they appear to be B's own child nodes on the tree (i.e., so that they appear to be B's BFR-NBRs). Then the usual BIER forwarding procedures apply. However, getting the packet from B to the child nodes of the failed BFR-NBR is a bit more complicated, as it may require using a unicast bypass tunnel to get around the failed node.

When using a unicast tunnel to get a packet to a BFR-NBR, it may be advantageous to (a) set the TTL of the MPLS label entry representing the "tunnel" to a large value, rather than copying the TTL value from the BIER-MPLS label, and (b) when the tunnel labels are popped off, to avoid copying the TTL from the tunnel labels to the BIER-MPLS label. That way, the TTL of the BIER-MPLS label would only control the number of "BFR hops" that the packet may traverse.

6.10. Use of Different BitStringLengths within a Domain

When a BFIR imposes a BIER header on a particular packet, it uses the value of BitStringLength that it has been provisioned to use when imposing a BIER header. For the BIER forwarding procedures to work properly, this BitStringLength must be supported by the intermediate BFRs and by the BFERs that may receive that packet.

Suppose one wants to migrate the BitStringLength used in a particular domain from one value (X) to another value (Y). The following migration procedure can be used. First, upgrade all the BFRs in the domain so that they support both value X and value Y. Once this is done, reprovision the BFIRs so that they use BitStringLength value Y.

However, it is always possible that the following situation will occur. Suppose a packet has been BIER-encapsulated with a BitStringLength value of X, and that the packet has arrived at BFR-A. How suppose that according to the routing underlay, the next hop is BFR-B, but BFR-B does not support the BitStringLength value of X. What should BFR-A do with the packet? BFR-A has three choices. It MUST be able to do one of the three, but the choice of which procedure to follow is a local matter. The three choices are:

- BFR-A MAY discard the packet.
- BFR-A MAY re-encapsulate the packet, using a BIER header whose BitStringLength value is supported by BFR-B. (Note that if BFR-B only supports BitStringLength values that are smaller than the BitStringLength value of the packet, this may require creating an additional copy of the packet.)
- BFR-A MAY treat BFR-B as if BFR-B did not support BIER at all, and apply the rules of Section 6.9.

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, Ross Callon (who contributed much of the text on deterministic ECMP), Nagendra Kumar, Christian Martin, Neale Ranns, Greg Shepherd, Albert Tian, Ramji Vaithianathan, and Jeffrey Zhang for their ideas and contributions to this work.

10. Contributor Addresses

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

Gregory Cauchie
Bouygues Telecom

Email: gcauchie@bouyguestelecom.fr

Mach (Guoyi) Chen
Huawei

Email: mach.chen@huawei.com

Arkadiy Gulko
Thomson Reuters
195 Broadway
New York NY 10007
US

Email: arkadiy.gulko@thomsonreuters.com

Wim Henderickx

Alcatel-Lucent
Copernicuslaan 50
Antwerp 2018
BE

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

Luay Jalil
Verizon
1201 E Arapaho Rd.
Richardson, TX 75081
US

Email: luay.jalil@verizon.com

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.

[ISIS_BIER_EXTENSIONS]
Przygienda, T., Ginsberg, L., Aldrin, S., and J. Zhang, "OSPF Extensions for Bit Index Explicit Replication", internet-draft draft-przygienda-bier-isis-ranges-01.txt, October 2014.

[MPLS_BIER_ENCAPS]
Wijnands, IJ., "BIER Encapsulation for MPLS Networks", internet-draft draft-wijnands-mpls-bier-encaps-02.txt, December 2014.

[OSPF_BIER_EXTENSIONS]
Psenak, P., Kumar, N., Wijnands, IJ., Dolganow, A., Przygienda, T., and J. Zhang, "OSPF Extensions for Bit Index Explicit Replication", internet-draft draft-psenak-ospf-bier-extensions-01.txt, October 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
BE

Email: ice@cisco.com

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

Email: erosen@juniper.net

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

Email: andrew.dolganow@alcatel-lucent.com

Tony Przygienda
Ericsson
300 Holger Way
San Jose, California 95134
US

Email: antoni.przygienda@ericsson.com

Sam K Aldrin
Huawei Technologies
2330 Central Express Way
Santa Clara, California
US

Email: aldrin.ietf@gmail.com

Internet Engineering Task Force
Internet-Draft
Intended status: Standards Track
Expires: June 7, 2015

IJ. Wijnands, Ed.
Cisco Systems, Inc.
E. Rosen, Ed.
Juniper Networks, Inc.
A. Dolganow
Alcatel-Lucent
J. Tantsura
Ericsson
S. Aldrin
Huawei Technologies
December 4, 2014

Encapsulation for Bit Index Explicit Replication in MPLS Networks
draft-wijnands-mpls-bier-encapsulation-02

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 the BIER encapsulation to be used in an MPLS network.

Table of Contents

1. Introduction

[BIER_ARCH] describes a new architecture for the forwarding of multicast data packets. That architecture 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. That architecture is known as "Bit Index Explicit Replication" (BIER).

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". In the context of a given sub-domain, each BFIR and BFER must have a BFR-id that is unique within that sub-domain. A BFR-id is just a number in the range [1,65535] that, relative to a BIER sub-domain, 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 sub-domain to which the packet has been assigned, a Set-Id (SI), a BitString, and a BitStringLength. Together these values identify the set of BFERs to which the packet must be delivered.

This document is applicable when a given BIER domain is both an IGP domain and an MPLS network. In this environment, the BIER encapsulation consists of two components:

- an MPLS label (which we will call the "BIER-MPLS label"); this label appears at the bottom of a packet's MPLS label stack.
- a BIER header, as specified in Section 3.

Following the BIER header is the "payload". The payload may be an IPv4 packet, an IPv6 packet, an ethernet frame, or an MPLS packet. If it 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 BIER header contains information identifying the type of the payload.

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 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 ISIS 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 sub-domain, a given BFR needs to advertise one such label for each combination of SI and BitStringLength. If the domain

contains multiple sub-domains, a BFR needs to advertise one such label per SI per BitStringLength for each sub-domain.

The BIER-MPLS labels are locally significant (i.e., unique only to the BFR that advertises them) downstream-assigned MPLS labels. For example, suppose that there is a single sub-domain (the default sub-domain), that the network is using a BitStringLength of 256, and that all BFERs in the sub-domain 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 sub-domain and to the BitStringLength 256.

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

L1: corresponding to sub-domain 0, BitStringLength 256, SI 0.

L2: corresponding to sub-domain 0, BitStringLength 256, SI 1.

L3: corresponding to sub-domain 0, BitStringLength 256, SI 2.

L4: corresponding to sub-domain 0, BitStringLength 256, SI 3.

L5: corresponding to sub-domain 0, BitStringLength 512, SI 0.

L6: corresponding to sub-domain 0, BitStringLength 512, SI 1.

L7: corresponding to sub-domain 1, BitStringLength 256, SI 0.

L8: corresponding to sub-domain 1, BitStringLength 256, SI 1.

L9: corresponding to sub-domain 1, BitStringLength 256, SI 2.

L10: corresponding to sub-domain 1, BitStringLength 256, SI 3.

L11: corresponding to sub-domain 1, BitStringLength 512, SI 0.

L12: corresponding to sub-domain 1, BitStringLength 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 <sub-domain 1, BitStringLength 512, SI 0> and

a label for <sub-domain 1, BitStringLength 512, SI 1>, a BFR could advertise a contiguous range of labels (in this case, a range containing exactly two labels) corresponding to <sub-domain 1, BitStringLength 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].

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

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 a BIER header (see Section 3) immediately follows the stack. It will also infer the packet's sub-domain, SI, and BitStringLength from the label. The packet's "incoming TTL" (see below) is taken from the TTL field of the label stack entry that contains the BIER-MPLS label.

The BFR MUST perform the MPLS TTL processing correctly. If the 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 incoming TTL. (Of course, if the incoming TTL is 1, the packet will not be forwarded at all, but will be discarded as an MPLS packet whose TTL has been exceeded.)

3. BIER Header

The BIER header is shown in Figure 1. This header appears after the end of the MPLS label stack, immediately after the MPLS-BIER label.

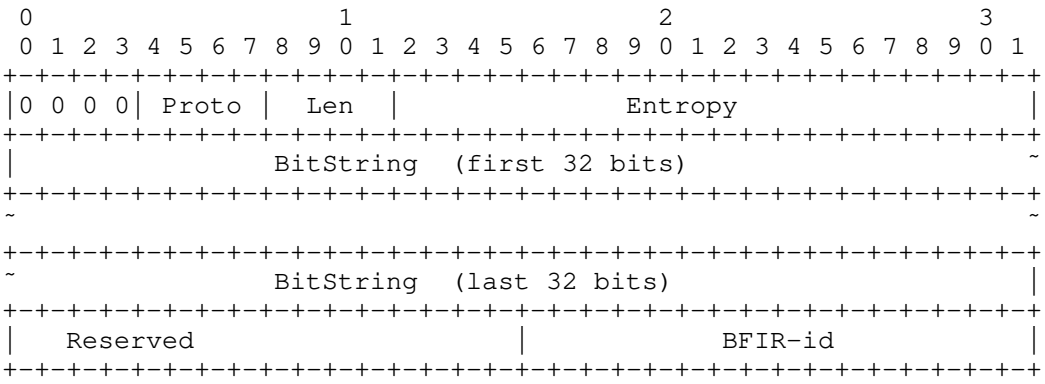


Figure 1: BIER Header

Ver:

The first 4 bits of the header are all set to zero; this ensures that the BIER header will not be confused with an IP header. This field can also be used as a version number if there are future revisions of the BIER header. However, the values 4 and 6 MUST NOT be used, as that may make the packets appear to some hardware forwarder to be IP packets.

Proto:

This 4-bit field identifies the type of the payload. (The "payload" is the packet or frame immediately following the BIER header.) The protocol field may take any of the following values:

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

2: MPLS packet with upstream-assigned label at top of stack (see [RFC5331]). If this value of the Proto field is used, the I bit MUST be set, and the BFR-id of the BFIR must be placed in the BFIR-id field. The BFIR-id provides the "context" in which the upstream-assigned label is interpreted.

3: Ethernet frame.

4: IPv4 packet.

6: IPv6 packet.

Len:

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

All other values of this field are illegal.

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, but the load balancing procedure MUST choose the same path for any two packets have the same entropy value.

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.

BitString:

The BitString that, together with the packet's SI, identifies the destination BFERs for this packet. Note that the SI for the packet is inferred from the BIER-MPLS label that precedes the BIER header.

BFIR-id

By default, this is the BFR-id of the BFIR, in the sub-domain 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.

4. Imposing and Processing the BIER Encapsulation

When a BFIR receives a multicast packet from outside the BIER domain, the BFIR carries out the following procedure:

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

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

3. If more than one sub-domain is supported, the BFIR assigns the packet to a particular sub-domain. Procedures for determining the sub-domain 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 sub-domain, of each of those 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 3). The BIER header contains the BitString that represents all the destination BFERs whose BFR-ids (in the given sub-domain) correspond to the given SI. It also contains the BFIR's BFIR-id in the sub-domain 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, but may be discussed in future revisions.

   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. Before transmitting a copy of the packet to a neighboring BFR, the BFIR finds the BIER-MPLS label that was advertised by the neighbor as corresponding to the given SI, sub-domain, and BitStringLength. An MPLS label stack is then preprended 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. The packet may then be transmitted to the neighboring BFR. (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 sub-domain, SI, and BitStringLength from 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 SI, sub-domain, and BitStringLength. 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.

Thus a BIER-encapsulated packet in an MPLS network consists of a packet that has:

- An MPLS label stack with a BIER-MPLS label at the bottom of the stack.
- A BIER header, as described in Section 3.
- The payload.

The payload may be an IPv4 packet, an IPv6 packet, an ethernet frame, or an MPLS packet. If it is an MPLS packet, the BIER header is followed by a second MPLS label stack; this stack is separate from the stack that precedes 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].

5. IANA Considerations

This document has no actions for IANA.

6. Security Considerations

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

As this document makes use of 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, 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
US

Email: arkadiy.gulko@thomsonreuters.com

Wim Henderickx
Alcatel-Lucent
Copernicuslaan 50
Antwerp 2018
BE

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

Tony Przygienda
Ericsson
300 Holger Way
San Jose, CA 95134
US

Email: antoni.przygienda@ericsson.com

9. References

9.1. Normative References

[BIER_ARCH]
Wijnands, IJ., "Multicast using Bit Index Explicit Replication Architecture", internet-draft draft-wijnands-bier-architecture-02, December 2014.

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

[RFC3032] Rosen, E., Tappan, D., Fedorkow, G., Rekhter, Y., Farinacci, D., Li, T., and A. Conta, "MPLS Label Stack Encoding", RFC 3032, January 2001.

[RFC5331] Aggarwal, R., Rekhter, Y., and E. Rosen, "MPLS Upstream Label Assignment and Context-Specific Label Space", RFC 5331, August 2008.

9.2. Informative References

[BIER_MVPN]
Rosen, E., Ed., Sivakumar, M., Wijnands, IJ., Aldrin, S., Dolganow, A., and T. Przygienda, "Multicast VPN Using Bier", internet-draft draft-rosen-l3vpn-mvpn-bier-02, December 2014.

[ISIS_BIER_EXTENSIONS]
Przygienda, T., Ginsberg, L., Aldrin, S., and J. Zhang, "OSPF Extensions for Bit Index Explicit Replication", internet-draft draft-przygienda-bier-isis-ranges-01.txt, October 2014.

[OSPF_BIER_EXTENSIONS]
Psenak, P., Kumar, N., Wijnands, IJ., Dolganow, A., Przygienda, T., and J. Zhang, "OSPF Extensions for Bit Index Explicit Replication", internet-draft draft-psenak-ospf-bier-extensions-01.txt, October 2014.

Authors' Addresses

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

Email: ice@cisco.com

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

Email: erosen@juniper.net

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

Email: andrew.dolganow@alcatel-lucent.com

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

Email: jeff.tantsura@ericsson.com

Sam K Aldrin
Huawei Technologies
2330 Central Express Way
Santa Clara, California
US

Email: aldrin.ietf@gmail.com

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: August 28, 2015

X. Xu
Huawei
S. Somasundaram
Alcatel-Lucent
C. Jacquenet
France Telecom
R. Raszuk
Mirantis Inc.

February 24, 2015

BIER Encapsulation
draft-xu-bier-encapsulation-02

Abstract

Bit Index Explicit Replication (BIER) is a new multicast forwarding paradigm which doesn't require an explicit tree-building protocol and doesn't require intermediate routers to maintain any multicast state. This document proposes a transport-independent BIER encapsulation header which is applicable in any kind of transport networks.

Status of This Memo

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

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

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

This Internet-Draft will expire on August 28, 2015.

Copyright Notice

Table of Contents

1. Introduction

Bit Index Explicit Replication (BIER) [I-D.wijnands-bier-architecture] is a new multicast forwarding paradigm which doesn't require an explicit tree-building protocol and doesn't require intermediate routers to maintain any multicast state. As described in [I-D.wijnands-bier-architecture], BIER requires that a multicast data packet (e.g., an IP packet or an MPLS packet) to be encapsulated with a BIER header that carries the information needed for supporting the BIER forwarding procedures. This information at least includes Set-Identifier (SI), Multi-Topology Identifier (MT-ID) and BitString. The SI and the BitString are used together to identify the set of egress BFRs (BFERs) to which the packet must be delivered. In addition, to indicate what type of payload is following the BIER header, a protocol type field is neccessary. This document proposes a transport-independent BIER encapsulation header which is applicable in any kind of transport networks.

1.1. Requirements Language

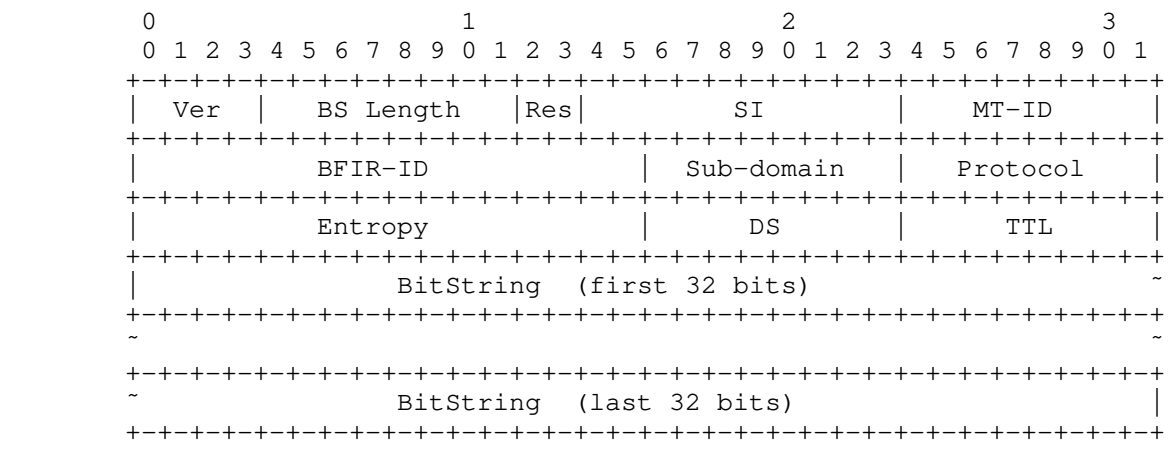
The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

2. Terminology

This memo makes use of the terms defined in [I-D.wijnands-bier-architecture].

3. BIER Header

The BIER header is shown as follows:



Ver(sion): a 4-bit field identifying the version of the BIER header. This document specifies version 0 of the BIER header.

BS Length: a one-octet field indicating the length of the BitString in 4-byte. Note that legal BS Length values are specified in [I-D.wijnands-bier-architecture].

Res: a 2-bit reserved field.

SI: a 10-bit field encoding the Set-Identifier (SI) for this packet.

MT-ID: a one-octet field indicating which routing topology [RFC4915] [RFC5120] should be applied for BIER forwarding.

BFIR-ID: a 2-octet field encoding the BFR-ID of the BFIR, in the sub-domain to which the packet has been assigned.

Sub-domain: a one-octet field encoding the sub-domain to which the packet has been assigned.

Protocol: a one-octet field indicating the protocol type of the BIER payload as per IP protocol numbers used in the Protocol field

of the IPv4 header and the Next Header field of IPv6 header. The valid BIER payload types include but not limited to IPv4, IPv6, MPLS, VXLAN [RFC7348],VXLAN-GPE [I-D.quinn-vxlan-gpe] , and etc. The corresponding IP Protocol numbers for VXLAN and VXLAN-GPE are to be allocated by IANA.

Entropy: a 2-octet field containing an "entropy" value that can be used for load balancing purposes.

BitString: a variable-length BitString field that, together with the SI field, identifies all the destination BFERs for this packet.

DS: The usage of this field is no different from that of the Differentiated Services (DS) field in the IPv4 or IPv6 headers [RFC2474].

TTL: The usage of this field is no different from that of the Time to Live (TTL) field in the IPv4 header.

4. Transport Encapsulation for BIER Header

Since the BIER encapsulation format as specified in Section 3 is transport-independent, it can be encapsulated with any type of transport encapsulation headers, such as Ethernet header, PPP header, IP header, MPLS header, GRE header, UDP header etc. It requires for each possible transport encapsulation header to be able to indicate the payload is an BIER header. For instance, In the BIER-in-MAC encapsulation case, the EtherType field in the Ethernet header is used. In the BIER-in-IP encapsulation case, the Protocol field in the IPv4 or or the Next-Header field in the IPv6 header is used. In the BIER-in-MPLS encapsulation case, either the Protocol Type field [I-D.xu-mpls-payload-protocol-identifier] within the MPLS packet or a to-be-assigned Extended Spcial Purpose label [RFC7274] is used.

5. Acknowledgements

TBD.

6. IANA Considerations

This document includes a request to IANA to allocate an EtherType code,a PPP protocol code, an IPv4 protocol code (i.e., an IPv6 Next-Header code), a UDP destination port for carring the BIER-encapsulated packet over the corresponding transport networks. Furthermore, This document includes a request to IANA to allocate IP Protocol numbers for VXLAN and VXLAN-GPE respectively.

7. Security Considerations

TBD.

8. References

8.1. Normative References

[ETYPES] The IEEE Registration Authority, "IEEE 802 Numbers", 2012.

[I-D.wijnands-bier-architecture]
Wijnands, I., Rosen, E., Dolganow, A., Przygienda, T., and S. Aldrin, "Multicast using Bit Index Explicit Replication", draft-wijnands-bier-architecture-04 (work in progress), February 2015.

[I-D.xu-mpls-payload-protocol-identifier]
Xu, X. and M. Chen, "MPLS Payload Protocol Identifier", draft-xu-mpls-payload-protocol-identifier-00 (work in progress), September 2013.

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

[RFC7274] Kompella, K., Andersson, L., and A. Farrel, "Allocating and Retiring Special-Purpose MPLS Labels", RFC 7274, June 2014.

8.2. Informative References

[I-D.quinn-vxlan-gpe]
Quinn, P., Agarwal, P., Fernando, R., Lewis, D., Kreeger, L., Smith, M., Yadav, N., Yong, L., Xu, X., Elzur, U., and P. Garg, "Generic Protocol Extension for VXLAN", draft-quinn-vxlan-gpe-03 (work in progress), July 2014.

[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, December 1998.

[RFC4915] Psenak, P., Mirtorabi, S., Roy, A., Nguyen, L., and P. Pillay-Esnault, "Multi-Topology (MT) Routing in OSPF", RFC 4915, June 2007.

[RFC5120] Przygienda, T., Shen, N., and N. Sheth, "M-ISIS: Multi Topology (MT) Routing in Intermediate System to Intermediate Systems (IS-ISs)", RFC 5120, February 2008.

[RFC7348] Mahalingam, M., Dutt, D., Duda, K., Agarwal, P., Kreeger, L., Sridhar, T., Bursell, M., and C. Wright, "Virtual eXtensible Local Area Network (VXLAN): A Framework for Overlaying Virtualized Layer 2 Networks over Layer 3 Networks", RFC 7348, August 2014.

Authors' Addresses

Xiaohu Xu
Huawei

Email: xuxiaohu@huawei.com

S Somasundaram
Alcatel-Lucent

Email: somasundaram.s@alcatel-lucent.com

Christian Jacquenet
France Telecom

Email: christian.jacquenet@orange.com

Robert Raszuk
Mirantis Inc.

Email: robert@raszuk.net