

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/>.

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 13, 2015.

Copyright Notice

Copyright (c) 2015 IETF Trust and the persons identified as the document authors. All rights reserved.

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

Table of Contents

1. Introduction	2
2. Specification of Requirements	3
3. BIER Use Cases	3
3.1. Multicast in L3VPN Networks	3
3.2. BUM in EVPN	4
3.3. IPTV and OTT Services	5
3.4. Multi-service, converged L3VPN network	6
3.5. Control-plane simplification and SDN-controlled networks	7
3.6. Data center Virtualization/Overlay	7
3.7. Financial Services	8
4. Security Considerations	9
5. IANA Considerations	9
6. Acknowledgments	9
7. References	9
7.1. Normative References	9
7.2. Informative References	9
Authors' Addresses	10

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 at 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 carried 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 allow 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 receiver behind every egress PE router 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 router 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 discrete 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 considered 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

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

Copyright (c) 2015 IETF Trust and the persons identified as the document authors. All rights reserved.

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

Table of Contents

1. Introduction	2
2. Terminology	3
3. IANA Considerations	4
4. Concepts	4
4.1. BIER Domains and Sub-Domains	4
5. Procedures	4
5.1. Enabling a BIER Sub-Domain	5
5.2. Multi Topology and Sub-Domain	5
5.3. Encapsulation	5
5.4. Tree Type	5
5.5. Label Advertisements for MPLS encapsulated BIER sub- domains	5
5.5.1. Special Consideration	6
5.6. BFR-id Advertisements	6
5.7. Flooding	6
5.8. Version	6
6. Packet Formats	7
6.1. BIER Info sub-TLV	7
6.2. BIER MPLS Encapsulation sub-sub-TLV	8
6.3. Optional BIER sub-domain Tree Type sub-sub-TLV	9
7. Security Considerations	11
8. Acknowledgements	11
9. Normative References	11
Authors' Addresses	12

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 compatibility 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).

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:

- o The sub-sub-TLV MUST be included once AND ONLY once within the sub-TLV.
- o 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 treatment on violations.
- o Bitstring lengths within the sub-sub-TLV MUST NOT repeat.
- o The sub-sub-TLV MUST include the required bitstring lengths per [I-D.draft-wijnands-bier-architecture-02].

- o All label range sizes MUST be greater than 0.
- o All labels MUST represent valid label values.

```

      0               1               2               3
      0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+-----+-----+-----+-----+
|   Type   |   Length   |                                     <--+
+-----+-----+-----+-----+-----+-----+-----+-----+
| Lbl Range Size|BS Len |                                     Label | <--+
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     ~~ (number repetitions derived from TLV length) ~~ |
+-----+-----+-----+-----+-----+-----+-----+-----+
| Lbl Range Size|BS Len |                                     Label | <--+
+-----+-----+-----+-----+-----+-----+-----+-----+

```

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 BFRs.

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:

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

```

      0               1               2               3
      0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+-----+-----+-----+-----+
|   Type           |   Length           |
+-----+-----+-----+-----+-----+-----+
| Tree Type        |
+-----+-----+-----+-----+-----+
| Tree Type specific opaque data|
+-----+-----+-----+-----+-----+
|   ~ up to TLV Length ~
+-----+-----+-----+-----+-----+
| Tree Type specific opaque data|
+-----+-----+-----+-----+-----+

```

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.

Copyright Notice

Copyright (c) 2015 IETF Trust and the persons identified as the document authors. All rights reserved.

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

Table of Contents

1. Introduction	2
2. Flooding of the BIER Information in OSPF	3
2.1. The BIER Sub-TLV	3
2.2. The BIER MPLS Encapsulation Sub-TLV	4
2.3. Flooding scope of BIER Information	5
3. Security Considerations	6
4. IANA Considerations	6
5. Acknowledgments	6
6. Normative References	6
Authors' Addresses	7

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:

0										1										2										3																			
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9										
Type										Length																																							
Subdomain-ID										MT-ID										BFR-id																													
										Sub-TLVs (variable)																																							
+-																														-+																			

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:

```

0                               1                               2                               3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|                               |                               |
|                               Type                               |
|                               |                               |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|Lbl Range Size |                               Label Range Base |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|   BS Length   |                               Reserved          |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

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: June 7, 2015

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

Multicast VPN Using BIER
draft-rosen-l3vpn-mvpn-bier-02

Abstract

The Multicast Virtual Private Network (MVPN) specifications require the use of multicast tunnels ("P-tunnels") that traverse a Service Provider's backbone network. The P-tunnels are used for carrying multicast traffic across the backbone. A variety of P-tunnel types are supported. Bit Index Explicit Replication (BIER) is a new 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. This document specifies the protocol and procedures that allow MVPN to use BIER as the method of carrying multicast traffic over an SP backbone network.

Status of This Memo

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

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <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 June 7, 2015.

Copyright Notice

Copyright (c) 2014 IETF Trust and the persons identified as the document authors. All rights reserved.

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

Table of Contents

1. Introduction	2
2. Use of the PMSI Tunnel Attribute	4
3. Explicit Tracking	6
4. Data Plane	7
5. Acknowledgments	7
6. IANA Considerations	7
7. Security Considerations	7
8. References	8
8.1. Normative References	8
8.2. Informative References	8
Authors' Addresses	9

1. Introduction

[RFC6513] and [RFC6514] specify the protocols and procedures that a Service Provider (SP) can use to provide Multicast Virtual Private Network (MVPN) service to its customers. Multicast tunnels are created through an SP's backbone network; these are known as "P-tunnels". The P-tunnels are used for carrying multicast traffic across the backbone. The MVPN specifications allow the use of several different kinds of P-tunnel technology.

Bit Index Explicit Replication (BIER) ([BIER_ARCH]) 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. The purpose of the current document is to specify the protocols and procedures needed in order to provide MVPN service using BIER to transport the multicast traffic over the backbone.

Although BIER does not explicitly build and maintain multicast tunnels, one can think of BIER as using a number of implicitly created tunnels through a "BIER domain". In particular, one can think of there as being one Point-to-Multipoint (P2MP) tunnel from each "Bit Forwarding Ingress Router" (BFIR) to all the "Bit Forwarding Egress Routers" (BFERs) in the BIER domain, where a BIER domain is generally co-extensive with an IGP network. These "tunnels" are not specific to any particular VPN. However, the MVPN architecture provides protocols and procedures that allow the traffic of multiple MVPNs to be aggregated on a single P-tunnel. In this document, we specify how to use these multi-VPN aggregation procedures to enable BIER to transport traffic from multiple MVPNs.

MVPN traffic must sometimes traverse more than one IGP domain, whereas BIER only carries multicast traffic within a single IGP domain. However, the MVPN specifications allow P-tunnels to be "segmented", where the segmentation points may either be Autonomous System Border Routers (ASBRs), as described in [RFC6514], or Area Border Routers (ABRs), as described in [SEAMLESS_MCAST]. As long as the segmentation points are capable of acting as BFIRs and BFERs, BIER can be used to provide some or all of the segments of a P-tunnel.

This revision of the document does not specify the procedures necessary to support MVPN customers that are using BIDIR-PIM. Those procedures will be added in a future revision.

This document uses the following terminology from [BIER_ARCH]:

- o BFR: Bit-Forwarding Router.
- o BFIR: Bit-Forwarding Ingress Router.
- o BFER: Bit-Forwarding Egress Router.

This document uses the following terminology from [RFC6513]:

- o MVPN: Multicast Virtual Private Network -- a VPN [RFC4364] in which multicast service is offered.
- o P-tunnel. A multicast tunnel through the network of one or more SPs. P-tunnels are used to transport MVPN multicast data
- o C-S: A multicast source address, identifying a multicast source located at a VPN customer site.
- o C-G: A multicast group address used by a VPN customer.

- o C-flow: A customer multicast flow. Each C-flow is identified by the ordered pair (source address, group address), where each address is in the customer's address space. The identifier of a particular C-flow is usually written as (C-S,C-G). Sets of C-flows can be identified by the use of the "C-*" wildcard (see [RFC6625]), e.g., (C-*,C-G).
- o I-PMSI A-D Route: Inclusive Provider Multicast Service Interface Auto-Discovery route. Carried in BGP Update messages, these routes are used to advertise the "default" P-tunnel for a particular MVPN.
- o S-PMSI A-D route: Selective Provider Multicast Service Interface Auto-Discovery route. Carried in BGP Update messages, these routes are used to advertise the fact that particular C-flows are bound to (i.e., are traveling through) particular P-tunnels.
- o PMSI Tunnel attribute (PTA). This BGP attribute carried is used to identify a particular P-tunnel. When C-flows of multiple VPNs is carried in a single P-tunnel, this attribute also carries the information needed to multiplex and demultiplex the C-flows.

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. Use of the PMSI Tunnel Attribute

As defined in [RFC6514], the PMSI Tunnel attribute is used to identify the particular P-tunnel to which one or more multicast flows are being assigned.

The PMSI Tunnel attribute (PTA) contains the following fields:

- o "Tunnel Type". IANA is requested to assign a new tunnel type codepoint for "BIER". This codepoint will be used to indicate that the PMSI is instantiated by BIER.
- o "Tunnel Identifier". When the "tunnel type" field is "BIER", this field contains two subfields:
 1. The first subfield is a single octet, containing the sub-domain-id of the sub-domain to which the BFIR will assign the packets that it transmits on the PMSI identified by the NLRI of the BGP I-PMSI or S-PMSI A-D route that contains this PTA. (How that sub-domain is chosen is outside the scope of this document.)

2. The second subfield is the BFR-Prefix (see [BIER_ARCH]) of the originator of the route that is carrying this PTA. This will either be a /32 IPv4 address or a /128 IPv6 address. Whether the address is IPv4 or IPv6 can be inferred from the total length of the PMSI Tunnel attribute.
- o "MPLS label". This field contains an upstream-assigned MPLS label. It is assigned by the router that originates the BGP route to which the PTA is attached. Constraints on the way in which the originating router selects this label are discussed below.
 - o "Leaf Info Required Bit". The setting of this bit depends upon the type of route and the NLRI of the route that carries the PTA.
 - * In an I-PMSI A-D route or a (C-*,C-*) S-PMSI A-D route, the bit SHOULD be clear.
 - * In other S-PMSI A-D routes, the bit SHOULD be set.

Note that if a PTA specifying "BIER" is attached to an I-PMSI or S-PMSI A-D route, the route MUST NOT be distributed beyond the boundaries of a BIER domain. That is, any routers that receive the route must be in the same BIER domain as the originator of the route. If the originator is in more than one BIER domain, the route must be distributed only within the BIER domain in which the BFR-Prefix in the PTA uniquely identifies the originator. As with all MVPN routes, distribution of these routes is controlled by the provisioning of Route Targets.

Suppose an ingress PE originates two x-PMSI A-D routes, where we use the term "x-PMSI" to mean "I-PMSI or S-PMSI". Suppose both routes carry a PTA, and the PTA of each route specifies "BIER".

- o If the two routes do not carry the same set of Route Targets (RTs), then their respective PTAs MUST contain different MPLS label values.
- o If the ingress PE is supporting MVPN extranet ([EXTRANET]) functionality, and if the two routes originate from different VRFs, then the respective PTAs of the two routes MUST contain different MPLS label values.
- o If the ingress PE is supporting the "Extranet Separation" feature of MVPN extranet (see Section 7.3 of [EXTRANET], section), and if one of the routes carries the "Extranet Separation" extended community and the other does not, then the respective PTAs of the two routes MUST contain different MPLS label values.

When segmented P-tunnels are being used, an ABR or ASBR may receive, from a BIER domain, an x-PMSI A-D route whose PTA specifies "BIER". This means that BIER is being used for one segment of a segmented P-tunnel. The ABR/ASBR may in turn need to originate an x-PMSI A-D route whose PTA identifies the next segment of the P-tunnel. The next segment may also be "BIER". Suppose an ASBR receives x-PMSI A-D routes R1 and R2, and as a result originates x-PMSI A-D routes R3 and R4 respectively, where the PTAs of each of the four routes specify a BIER.. Then the PTAs of R3 and R4 MUST NOT specify the same MPLS label, UNLESS both of the following conditions hold:

- o R1 and R2 have the same "originating router" in their respective NLRIs.
- o R1 and R2 specify the same MPLS label in their respective PTAs.

3. Explicit Tracking

[Editor's note: The procedures of this section are still under discussion, and significant changes may be expected in the next revision.]

When using BIER to transport an MVPN data packet through a BIER domain, an ingress PE functions as a BFIR (see [BIER_ARCH]). The BFIR must determine the set of BFERs to which the packet needs to be delivered. This is done by using the explicit tracking mechanism specified in [RFC6513] and [RFC6514].

To determine the set of BFERs to which a given MVPN data packet needs to be delivered, the BFIR originating an S-PMSI A-D route sets the LIR bit in the route's PTA. Per [RFC6514], the BFERs will respond with Leaf A-D routes. By matching the received Leaf A-D routes to the originated S-PMSI A-D routes, the originator of the S-PMSI A-D route determines the set of BFERs that need to receive the multicast data flow (or flows) that is (are) identified in the NLRI of the of the S-PMSI A-D route.

This requires that each BFIR originate an S-PMSI A-D route for each C-flow for which it serves as BFIR. The BFIR MAY include, in each such route, a PTA as described in Section 2. However, if the BFIR has originated an I-PMSI A-D route or a wildcard S-PMSI A-D route that "matches" (according to the rules of [RFC6625]) a particular C-flow, then it may do explicit tracking for that C-flow by originating an S-PMSI A-D route for that C-flow, but including a PTA that specifies "no tunnel type".

4. Data Plane

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

To transmit an MVPN data packet, an ingress PE follows the rules of [RFC6625] to find the S-PMSI A-D route or I-PMSI A-D route that is a "match for transmission" for that packet. (In applying the rules of [RFC6625], any S-PMSI A-D route with a PTA specifying "no tunnel information" is ignored.) If the matching route has a PTA specifying a "BIER", the (upstream-assigned) MPLS label from that PTA is pushed on the packet's label stack. Then the packet is forwarded according to the procedures of [BIER_ARCH] and [BIER_ENCAPS]. (See especially Section 4, "Imposing and Processing the BIER Encapsulation", of [BIER_ENCAPS].)

When a BFER receives an MVPN multicast data packet that has been BIER-encapsulated, the BIER layer passes the following information to the multicast flow layer:

- o The BFR-prefix corresponding to the sub-domain-id and BFIR-id in the BIER header.
- o The "payload", which is an MPLS packet whose top label is an upstream-assigned label. The BFR-prefix provides the "context" in which the upstream-assigned label is interpreted.

Note that per [RFC5331], the context for an upstream-assigned label is the IP address of the label assigner, which in this case is the BFR-prefix of the BFIR.

5. Acknowledgments

The authors wish to thank Jeffrey Zhang for his ideas and contributions to this work.

6. IANA Considerations

IANA is requested to assign a value for "BIER" from the "P-Multicast Service Interface Tunnel (PMSI Tunnel) Tunnel Types" registry. The reference should be this document.

7. Security Considerations

The security considerations of [BIER_ARCH], [BIER_ENCAPS], [RFC6513] and [RFC6514] are applicable.

8. References

8.1. Normative References

- [BIER_ARCH] Wijnands, IJ., "Multicast using Bit Index Explicit Replication Architecture", internet-draft draft-wijnands-bier-architecture-02, December 2014.
- [BIER_ENCAPS] Wijnands, IJ., "Multicast using Bit Index Explicit Replication Architecture", internet-draft draft-wijnands-mpls-bier-encapsulation-02, December 2014.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC4364] Rosen, E. and Y. Rekhter, "BGP/MPLS IP Virtual Private Networks (VPNs)", RFC 4364, February 2006.
- [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.
- [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.
- [RFC6625] Rosen, E., Rekhter, Y., Hendrickx, W., and R. Qiu, "Wildcards in Multicast VPN Auto-Discovery Routes", RFC 6625, May 2012.

8.2. Informative References

- [EXTRANET] Rekhter, Y. and E. Rosen, "Extranet Multicast in BGP/IP MPLS VPNs", internet-draft draft-ietf-l3vpn-mvpn-extranet-05, July 2014.
- [SEAMLESS_MCAST] Rekhter, Y., Aggarwal, R., Morin, T., Grosclaude, I., Leymann, N., and S. Saad, "Inter-Area P2MP Segmented LSPs", internet-draft draft-ietf-mpls-seamless-mcast-14, June 2014.

Authors' Addresses

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

Email: erosen@juniper.net

Mahesh Sivakumar
Cisco Systems, Inc.
510 McCarthy Blvd
Milpitas, California 95035
US

Email: masivaku@cisco.com

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

Email: ice@cisco.com

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

Email: aldrin.ietf@gmail.com

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

Internet Engineering Task Force
Internet-Draft
Intended status: Informational
Expires: August 10, 2015

G. Shepherd
Cisco
A. Dolganow
Alcatel-Lucent
A. Gulko
Thomson Reuters
February 6, 2015

Bit Indexed Explicit Replication (BIER) Problem Statement
draft-shepherd-bier-problem-statement-02

Abstract

There is a need to simplify network operations for multicast services. Current solutions require a tree-building control plane to build and maintain end-to-end tree state per flow, impacting router state capacity and network convergence times. Multi-point tree building protocols are often considered complex to deploy and debug and may include mechanics from legacy use-cases and/or assumptions which no longer apply to the current use-cases. When multicast services are transiting a provider network through an overlay, the core network has a choice to either aggregate customer state into a minimum set of core states resulting in flooding traffic to unwanted network end-points, or to map per-customer, per-flow tree state directly into the provider core state amplifying the network-wide state problem.

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 10, 2015.

Copyright Notice

Copyright (c) 2015 IETF Trust and the persons identified as the document authors. All rights reserved.

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

Table of Contents

1. Introduction	2
1.1. Requirements Language	3
2. Objectives	3
3. Deering's Multicast Model	5
4. Network Based Source Discovery	7
5. Receiver Driven State	8
6. Multicast Virtual Private Networks	9
7. Overlay	10
8. Summary	10
9. IANA Considerations	11
10. Security Considerations	11
11. References	11
11.1. Normative References	11
11.2. Informative References	11
Appendix A. Additional Stuff	12
Authors' Addresses	12

1. Introduction

There is a need to simplify network operations for multicast services. Current solutions require a tree-building control plane, to build and maintain end-to-end tree state per flow, impacting router state capacity and network convergence times. Multi-point tree building protocols are often considered complex to deploy and debug and include mechanics from legacy use-cases and/or assumptions which may no longer apply to the current use-case. When multicast services are transiting a provider network through an overlay, the core network has a choice to either aggregate customer state into a minimum set of core states resulting in flooding traffic to unwanted network end-points, or to map per-customer, per-flow tree state

directly into the provider core state amplifying the network-wide state problem.

This document attempts to discuss the uses, benefits and challenges of the current multicast solutions and to put them in an historical context to better understand why we are where we are today, and to provide a framework for discussion around new solutions that may address our current requirements and challenges.

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. Objectives

IP Multicast services have been widely adopted in networks where the benefits of efficient, concurrent delivery of content to a sufficiently large set of receivers outweighs the complexity and challenges of deploying and managing the current set of multicast protocols. These deployments are primarily dedicated multicast islands with very little cross-domain inter-networking, and fall short of the early dreams of a multicast enabled Internet.

Multicast began with a large set of requirements shoehorned into a single, complex protocol. Over time, multicast protocols essentially devolved into a set of more simple components to overcome the original complexity, and to address a growing set of use cases. Many of the early complexity can be avoided today by correctly selecting your service model and protocols. But the standard set of protocols available can still be considered overloaded for various reasons.

The current problems associated with the today's multicast solutions can be stated as follows:

- Current multicast methods all require explicit tree building protocols, thereby incurring a lot of state in the transit nodes.
- Receiver driven tree state uses Reverse Path Forwarding (RPF) to build the trees toward the root which often results in multicast forwarding following different paths than unicast forwarding between the same two endpoints.
- Multicast convergence times are negatively impacted by tree state. Any network transition requires unicast to first converge. Once unicast has converged multicast must then recalculate RPF for every tree and rebuild the trees by sending join messages toward

the new RPF neighbor per tree. Joins toward a common RPF neighbor can be aggregated but only up to the link MTU. In large multicast deployments this can result in multicast convergence times of up to a minute or more. In extreme cases the active state may time out before all the new joins are sent and received resulting in multicast to permanently fail after a network failure event even though there is a restored path. This has put an upward bound on the amount of state a multicast network can support.

- Current multicast methods, if they are to provide optimal delivery of multicast packets, require one explicitly built tree per multicast flow; there is no way to aggregate flows (having one state for multiple flows) without sacrificing optimal delivery. In the case of Multicast Virtual Private Network (MVPN) deployments, the operator is forced to choose between unwanted flooded traffic across an aggregate state entry and exposing customer state in the core.

- Some multicast solutions include data-driven events. This has required specialized capabilities to be integrated into routing equipment to protect the control plane from the multicast data plane increasing the cost of multicast support in routing equipment.

- Maintaining and troubleshooting multicast networks can be very difficult. The available solutions are so different than unicast, often revealing unique corner cases that specialized training and skills, and frequently dedicated staff are required just to operate multicast services on a network.

- Current Multicast Virtual Private Networks [RFC6513](MVPN) introduced Border Gateway Protocol[RFC4271](BGP) routes for neighbour discovery and Protocol Independent Multicast[RFC4601](PIM) Join/Prune propagation. In some deployments when many Multicast MVPNs with many Provider Edge (PE) routers exist in a network and at least some of those MVPNs have a large number of customer-multicast flows, the resulting tax on BGP may be deemed undesired as millions of BGP routes can easily result from multicast deployments. Therefore a solution that allows large MVPN scale with large number of edge PEs and c-multicast flows per MVPN is desired.

- With the introductions of Segment Routing, some networks may elect to remove the Multiprotocol Label Switching[RFC3031](MPLS) control plane and rely on Interior Gateway Protocol-only or Software Defined Networking-based Segment Routing. In such networks the alternative to existing mechanisms is needed for multicast. Removing the MPLS control plane for unicast makes

little sense unless the multicast control plane also gets simplified.

- The benefits of multi-point services are well understood, but the challenges with the current solutions often result in a failed cost/benefit analysis. Today only those networks with an overwhelming business need have successful multicast deployments, and the rest of the community have come to think of multicast as a failed technology.

How did we get here? What follows is a semi-chronological tour through the devolution of multicast protocols, solutions, and use-cases, describing why earlier complexities and challenges existed, and how they were overcome. This may help frame future work to overcome our current challenges.

3. Deering's Multicast Model

The original Multicast Extensions to the Internet Protocol [RFC0966] and Host Extensions for IP Multicasting [RFC1112] were envisioned by Stephen Deering as part of his graduate work at Stanford University. The need for a multi-point service model was motivated by the advent and deployment of layer3 network topologies breaking existing layer2 applications. The need arose to create an underlay service with the characteristics of a broadcast domain to allow these layer2 applications to continue to function without modification across a layer3 infrastructure.

Though the community quickly saw the value and envisioned many other uses for a multi-point service model, a broadcast domain remained the target model for the solution and the list of requirements focused around those of a broadcast domain. For simplicity the rules of this underlay broadcast domain can be summed up as follows: anyone can send packets into the domain; all members will receive all packets sent into the domain. In order for these layer2 applications to function across this broadcast domain overlay, all of the functions to provide this service were loaded onto network layer.

This new multi-point model was called Multicast. The first multicast solution adopted by the IETF was Distance Vector Multicast Routing Protocol [RFC1075](DVMRP). As the name implies, DVMRP uses a distance-vector routing algorithm derived from Routing Information Protocol [RFC1058](RIP) in combination with the Truncated Reverse Path Broadcasting (TRPB) algorithm to build and maintain tree state and forward multicast packets along these distribution trees. The Internet Assigned Numbers Authority (IANA) was asked to reserve a portion of the global IPv4 address space for multicast destination

addresses required by this model, and in response 224/4 was allocated as the Class D address space for IP Multicast group addresses.

DVMRP has no concept of a "join" message. All new source packets for any given group were simply flooded downstream--essentially broadcasted--following the DVMRP topology. Each leaf of the tree was responsible for sending Non Membership Reports (NMR--prunes) toward the source if there were no downstream receivers for the group. This mechanism came to be known as flood-and-prune, and is a very primitive form of network-based source discovery that all the contemporary applications came to depend on. These contemporary applications were inherently many-to-many either by the nature of the data distribution model, or at the least depended on the many-to-many nature of the network-based-source discovery mechanism.

DVMRP also incorporated the IETF's first specification of an encapsulated overlay. It was clear that this new model would not be supported by every node in the path, and an encapsulation allowed early adopters to build a global multi-point, or multicast capable topology as an overlay.

For clarity of discussion, the functions of the Deering model can be described as:

- Tree building and maintenance
- Network-based source discovery
- Source route information
- Overlay mechanism - tunneling

DVMRP was considered over-loaded in that it carries network source routing information within the protocol in parallel to any existing Interior Gateway Protocol (IGP) generated local routing table. The next generation goal was to focus on the multi-point services needed for the model but to use the local, native routing table as needed for Reverse Path Check (RPF). From this came the advent of Protocol Independent Multicast Sparse Mode [RFC4601](PIM-SM) and Protocol Independent Multicast Dense Mode [RFC3973](PIM-DM). PIM removed any embedded source routing function from the protocol, and instead relied on the exiting routing table as generated from the deployed IGP. PIM also removed any overlay functionality, but retained network-based source discovery as a fundamental part of the protocol. Oops.

4. Network Based Source Discovery

The Deering model introduced the concept of a Group address (G) representing a single broadcast domain. Any source is allowed to send to the group address and the multicast routing infrastructure will build tree state from every source to all interested receivers. All group members only need to signal their G membership to the network and the network will ensure that all source traffic sending to that same group address will arrive at all group members. The network-based source discovery operation providing these functions was intended to provide operational constancy with a layer2 broadcast domain, but comes at significant cost.

Allowing any source to send to a group is an obvious security vulnerability. Many implementations today provide various layers of access control both at the edges and core of the network just to overcome the security concerns for the basic operation of the multicast network.

Network-based source discovery methods can be grouped into two types; flood and prune (DVMRP, PIM-DM), or explicit join (PIM-SM). Both methods depend on the arrival of data to trigger complex network functions to build and maintain the per-source distribution of data for every group. Multicast is often considered complex, fragile, and difficult to troubleshoot, but it is most often the network-based source discovery functions that are the cause of this reputation.

The majority of the use-cases for multicast today are for content with well-know sources. The development of Internet Group Membership Protocol [RFC3376] (IGMPv3) provided a mechanism for group members to signal interest in a source and a group, eliminating the need for network-based source discovery, and facilitating the advent of Source Specific Multicast [RFC4607] (SSM). Many operators still ask how potential SSM group members learn about the sources. The answer is simply to use the same mechanism in which they learned about the group - out-of-band. Source (and group) discovery mechanisms are better served at the application layer for most use-cases. With SSM multicast content can be forwarded and constrained to a single source-rooted tree, or (S,G) channel which has several key benefits:

- Simplified configuration and operation
- Elimination of rouge sources 'stealing' receivers
- Elimination of rouge sources consuming network resources
- Elimination of group address resource restrictions

5. Receiver Driven State

Today's multicast solutions are primarily receiver driven. This is a logical approach in that it is the receiver that decides if and when to join or leave a group or channel. Receiver driven distribution trees built hop-by-hop are an efficient way to dynamically build and scale very large membership fanout. It can be argued that a receiver driven tree's radius can scale infinitely without impact to any upstream segment or node for that tree. But it does then require forwarding state for each tree, or pre-flow state.

The joins propagate upstream from the receiver toward the source or root of the tree, following the unicast routing table. But this reverse path may differ from the optimal unicast forwarding path from the source to the receiver. The result is multicast traffic potentially taking a different forwarding path than unicast traffic between the same to network endpoints. This can often complicate network and traffic engineering.

Each of the existing multicast solutions today, native or overlay, builds and maintains forwarding state per flow, or aggregates some flows into a subset of flow-states. On the surface this may look like an unbounded problem, but in actuality the flow state is only present along the branches of the tree, and no one router needs to maintain global tree state. Router state capacity is not infinite, and this coupling of receiver actions to network state is a potential Denial of Service (DoS) vector. Most implementations today have provided filtering and state-limiting capabilities to secure the multicast infrastructure from this vulnerability.

Increasing multicast forwarding state can also negatively impact network convergence performance. Unicast is only concerned with topology, and any topology changes can converge in a relatively bounded amount of time. The same topology change requires the multicast protocol to rebuild the forwarding state for every active flow. The resulting multicast convergence times are directly dependent on the amount of flow state affected by the convergence event. In extreme cases, the sending, receiving, and processing of the join state for all active flows can exceed the flow state timers resulting in a race condition in which convergence never occurs. Today's implementations have had to incorporate various proprietary solutions to improve network convergence times in large flow-state multicast deployments.

The pros and cons of receiver driven state are as follows:

Pros:

Infinitely scales distribution radius

Aligns with receiver driven join model

Cons:

Potential state DoS vector

Host driven network events

Unbounded per-flow state

Unicast/Multicast traffic divergence

Non-deterministic join latency

Convergence times increasing with flow-state

6. Multicast Virtual Private Networks

Multicast Virtual Private Networks [RFC6513](MVPN) are solutions which allow a core network to transit edge network multicast flows over a core transit network to and from only those MVPN member nodes, without exposing the edge network addressing into the core network forwarding state. The solutions attempt to minimize core state by aggregating trees per-VRF/PE. But this aggregation has the side affect of sending all multicast traffic from that VRF/PE to all other VRF/PE members, whether or not they have down stream flow state.

Various optimizations are available to selectively de-aggregate flow state to better constrain the traffic distribution to only those VRF/PEs with active state. This becomes a trade off between unwanted traffic and an increase in core flow state. These solutions are often data driven resulting in core router state being triggered by date and receiver events.

In addition to a potential BGP route explosion due to an MVPN deployment scale as discussed in section 2, another issue with MVPN relates to architectures used when MVPN deployments require both video-distribution-like model, well served by point-to-multipoint (P2MP) connectivity, and many-to-many model requiring Multipoint-to-multipoint (MP2MP) connectivity. Today, if both models are deployed in a single network, either MP2MP or a mesh of P2MP trees needs to be established, or dual P2MP/MP2MP mLDP architecture may be used, or MP2MP mLDP can be used for both P2MP and MP2MP connectivity. None of those models is optimal as each requires a trade-off between supported protocols, optimal delivery, and operational complexity.

7. Overlay

Deering had the correct insight to assume not every node in a network would be capable of natively transiting multicast flows. The migration to PIM was an attempt to move to a completely native model, which was the right direction. But in this move it also abandoned any other solution for incorporating an underlay into the topology for those portions of the network which for whatever reason do not support native multicast. Early deployments of PIM often incorporated static Generic Routing Encapsulation [RFC2784](GRE) tunnels between PIM domains in an attempt to create an inter domain multicast deployment.

Static tunneling has its use cases and benefits, but it is not the ideal tool to dynamically stitch together a large and topologically diverse receiver population. A receiver driven distribution model would be better served with a receiver driving overlay mechanism. This would indicate that when overlay was removed from the tree building protocol it should have migrated to IGMPv3 and Multicast Listener Discovery [RFC3810](MLDv2), the membership protocol, but it was seen as a necessary requirement at that time. To fill this requirement today Automatic Multicast Tunnels (AMT) is being progressed as the overlay standard for bridging multicast interested receivers over unicast only intermediate networks.

8. Summary

Multicast began with a heavily overloaded protocol DVMRP, and has evolved over time by removing functionality from this all-in-one solution, and off-loading certain function to either more specialized protocols or existing protocols and functions. Multicast has what may be the unique distinction of starting very complex, but evolving through more simple stages along the way. It may be time to consider the next step in the evolution toward simplicity.

Today we depend on receiver driven joins propagating end-to-end from receivers toward sources, and maintaining per-flow state in every node along the path. This state crosses administrative domains. Unicast has a simple model where local specificity stays local and does not directly impact the global table. Multicast state has no administrative boundaries today. It may be beneficial to consider the autonomy of networks in the path, and their specific topology and requirements. PIM successfully utilizes the available routing table for RPF checks and joins. This route table may also be considered as a source of topology information for a set of receiver nodes within a given network.

9. IANA Considerations

This memo includes no request to IANA.

All drafts are required to have an IANA considerations section (see Guidelines for Writing an IANA Considerations Section in RFCs [RFC5226] for a guide). If the draft does not require IANA to do anything, the section contains an explicit statement that this is the case (as above). If there are no requirements for IANA, the section will be removed during conversion into an RFC by the RFC Editor.

10. Security Considerations

All drafts are required to have a security considerations section. See RFC 3552 [RFC3552] for a guide.

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

- [RFC0966] Deering, S. and D. Cheriton, "Host groups: A multicast extension to the Internet Protocol", RFC 966, December 1985.
- [RFC1058] Hedrick, C., "Routing Information Protocol", RFC 1058, June 1988.
- [RFC1075] Waitzman, D., Partridge, C., and S. Deering, "Distance Vector Multicast Routing Protocol", RFC 1075, November 1988.
- [RFC1112] Deering, S., "Host extensions for IP multicasting", STD 5, RFC 1112, August 1989.
- [RFC2629] Rose, M., "Writing I-Ds and RFCs using XML", RFC 2629, June 1999.
- [RFC2784] Farinacci, D., Li, T., Hanks, S., Meyer, D., and P. Traina, "Generic Routing Encapsulation (GRE)", RFC 2784, March 2000.
- [RFC3031] Rosen, E., Viswanathan, A., and R. Callon, "Multiprotocol Label Switching Architecture", RFC 3031, January 2001.

- [RFC3376] Cain, B., Deering, S., Kouvelas, I., Fenner, B., and A. Thyagarajan, "Internet Group Management Protocol, Version 3", RFC 3376, October 2002.
- [RFC3552] Rescorla, E. and B. Korver, "Guidelines for Writing RFC Text on Security Considerations", BCP 72, RFC 3552, July 2003.
- [RFC3810] Vida, R. and L. Costa, "Multicast Listener Discovery Version 2 (MLDv2) for IPv6", RFC 3810, June 2004.
- [RFC3973] Adams, A., Nicholas, J., and W. Siadak, "Protocol Independent Multicast - Dense Mode (PIM-DM): Protocol Specification (Revised)", RFC 3973, January 2005.
- [RFC4271] Rekhter, Y., Li, T., and S. Hares, "A Border Gateway Protocol 4 (BGP-4)", RFC 4271, January 2006.
- [RFC4601] Fenner, B., Handley, M., Holbrook, H., and I. Kouvelas, "Protocol Independent Multicast - Sparse Mode (PIM-SM): Protocol Specification (Revised)", RFC 4601, August 2006.
- [RFC4607] Holbrook, H. and B. Cain, "Source-Specific Multicast for IP", RFC 4607, August 2006.
- [RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 5226, May 2008.
- [RFC6513] Rosen, E. and R. Aggarwal, "Multicast in MPLS/BGP IP VPNs", RFC 6513, February 2012.

Appendix A. Additional Stuff

This becomes an Appendix.

Authors' Addresses

Greg Shepherd (editor)
Cisco
170 W. Tasman Dr.
San Jose
US

Email: gjshep@gmail.com

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

Email: andrew.dolganow@alcatel-lucent.com

Arkadiy Gulko (editor)
Thomson Reuters

Email: arkadiy.gulko@thomsonreuters.com

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

IJ. Wijnands, Ed.
Cisco Systems, Inc.
E. Rosen, Ed.
Juniper Networks, Inc.
A. Dolganow
Alcatel-Lucent
T. Przygienda
Ericsson
S. Aldrin
Huawei Technologies
March 6, 2015

Multicast using Bit Index Explicit Replication
draft-wijnands-bier-architecture-05

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 September 7, 2015.

Copyright Notice

Copyright (c) 2015 IETF Trust and the persons identified as the document authors. All rights reserved.

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

Table of Contents

1. Introduction	3
2. The BFR Identifier and BFR-Prefix	5
3. Encoding BFR Identifiers in BitStrings	6
4. Layering	8
4.1. The Routing Underlay	8
4.2. The BIER Layer	9
4.3. The Multicast Flow Overlay	10
5. Advertising BFR-ids and BFR-Prefixes	10
6. BIER Intra-Domain Forwarding Procedures	12
6.1. Overview	12
6.2. BFR Neighbors	13
6.3. The Bit Index Routing Table	14
6.4. The Bit Index Forwarding Table	14
6.5. The BIER Forwarding Procedure	15
6.6. Examples of BIER Forwarding	17
6.6.1. Example 1	18
6.6.2. Example 2	18
6.7. Equal Cost Multi-path Forwarding	20
6.7.1. Non-deterministic ECMP	21
6.7.2. Deterministic ECMP	22
6.8. Prevention of Loops and Duplicates	24
6.9. When Some Nodes do not Support BIER	24
6.10. Use of Different BitStringLengths within a Domain	26
7. IANA Considerations	27
8. Security Considerations	27
9. Acknowledgements	27
10. Contributor Addresses	27
11. References	29
11.1. Normative References	29

11.2. Informative References	29
Authors' Addresses	29

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" (BFRs). 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 BFRs 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:

- o by convention, the least significant (rightmost) bit in the BitString is "bit 1", and the most significant (leftmost) bit is "bit BitStringLength".
- o if a BIER-encapsulated packet has an SI of n , and a BitString with bit k set, then the packet must be delivered to the BFER whose BFR-id (in the sub-domain to which the packet has been assigned) is $n \cdot \text{BitStringLength} + k$.

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 BFRs 13 and 26 is via BFR-B, but the best path to BFR 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 BFRs 13 and 26, and BFR-C will forward the packet only towards BFR 235. This ensures that each BFR 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 BFRs. Further, since the set of BFRs for a given packet is explicitly encoded into the BIER header, the packet is not sent to any BFR 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 BFRs in the same number of bits that it would take to carry the IPv6 address of a single BFR. 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 BFR 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 BFRs. 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 BFRs, it is RECOMMENDED to assign all the BFR-ids from the range [1,256]. If there are more than 256 BFRs, 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

- o SI is the integer part of the quotient $(N-1)/\text{BitStringLength}$
- o 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 $((N-1) \bmod \text{BitStringLength}) + 1$.

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 BFR 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 BFRs 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 $\langle \text{BFR}(k), \text{BFR}(k+1), \dots, \text{BFR}(k+n) \rangle$ such that $\text{BFR}(k+j)$ is "adjacent" to $\text{BFR}(k+j+1)$ (for $0 \leq j < n$).

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:

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

- o The imposition by a BFIR of a BIER header on a multicast data packet.
- o The procedures for forwarding BIER-encapsulated packets, and for modifying the BIER header during transit.

4.3. The Multicast Flow Overlay

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

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

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

- o 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.

- o 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.

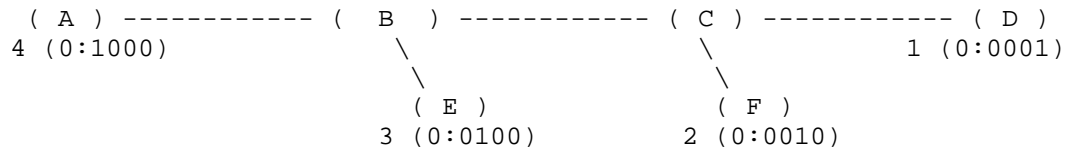


Figure 1: BIER Topology 1

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 sent 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				BFR-B BIFT				BFR-C BIFT			
Id	F-BM	NBR		Id	F-BM	NBR		Id	F-BM	NBR	
1	0111	B		1	0011	C		1	0001	D	
2	0111	B		2	0011	C		2	0010	F	
3	0111	B		3	0100	E		3	1100	B	
4	1000	A		4	1000	A		4	1100	B	

Figure 5: BIFTs for Forwarding Examples

6.6.1. Example 1

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

- o Since the packet's BitString is 0001, BFIR-A finds that the first bit in the string is bit 1. Looking at entry 1 in its BIFT, BFR-A determines that the bit mask F-BM is 0111 and the BFR-NBR is BFR-B.
- o BFR-A then makes a copy of the packet, and applies F-BM to the copy: Copy->BitString &= 0111. The copy's Bitstring is now 0001 (0001 & 0111).
- o The copy is now sent to BFR-B.
- o BFR-A then updates the packet's BitString by applying the inverse of the F-BM: Packet->Bitstring &= ~F-BM. As a result, the packet's BitString is now 0000 (0001 & 1000).
- o As the packet's BitString is now zero, the forwarding procedure is complete.

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

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

6.6.2. Example 2

This example is similar to Example 1, except that BFIR-A has learned from the multicast flow layer that both BFER-D and BFER-E are interested in a given multicast flow. If BFIR-A receives a packet of that flow from outside the BIER domain, BFIR-A applies the BIER

encapsulation to the packet. The encapsulation must be such that the SI is zero. The encapsulation also includes a BitString with two bits set: bit 1 is set (as in example 1) to indicate that BFR-D is a BFER for this packet, and bit 3 is set to indicate that BFR-E is a BFER for this packet. I.e., the BitString (assuming again a BitStringLength of 4) is 0101. To forward the packet, BFIR-A follows the procedures of Section 6.5:

- o Since the packet's BitString is 0101, BFIR-A finds that the first bit in the string is bit 1. Looking at entry 1 in its BIFT, BFR-A determines that the bit mask F-BM is 0111 and the BFR-NBR is BFR-B.
- o BFR-A then makes a copy of the packet, and applies the F-BM to the copy: Copy->BitString &= 0111. The copy's Bitstring is now 0101 (0101 & 0111).
- o The copy is now sent to BFR-B.
- o BFR-A then updates the packet's BitString by applying the inverse of the F-BM: Packet->Bitstring &= ~F-BM. As a result, the packet's BitString is now 0000 (0101 & 1000).
- o As the packet's BitString is now zero, the forwarding procedure is complete.

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

- o Since the packet's BitString is 0101, BFR-B finds that the first bit in the string is bit 1. Looking at entry 1 in its BIFT, BFR-B determines that the bit mask F-BM is 0011 and the BFR-NBR is BFR-C.
- o BFR-B then makes a copy of the packet, and applies the F-BM to the copy: Copy->BitString &= 0011. The copy's Bitstring is now 0001 (0101 & 0011).
- o The copy is now sent to BFR-C.
- o BFR-B then updates the packet's BitString by applying the inverse of the F-BM: Packet->Bitstring &= F-BM. As a result, the packet's BitString is now 0100 (0101 & 1100).
- o Now BFR-B finds the next bit in the packet's (modified) BitString. This is bit 3. Looking at entry 3 in its BIFT, BFR-B determines that the F-BM is 0100 and the BFR-NBR is BFR-E.

- o BFR-B then makes a copy of the packet, and applies the F-BM to the copy: $\text{Copy} \rightarrow \text{BitString} \&= 0100$. The copy's Bitstring is now 0100 (0100 & 0100).
- o The copy is now sent to BFR-E.
- o BFR-B then updates the packet's BitString by applying the inverse of the F-BM: $\text{Packet} \rightarrow \text{Bitstring} \&= \sim \text{F-BM}$. As a result, the packet's BitString is now 0000 (0100 & 1011).
- o 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.

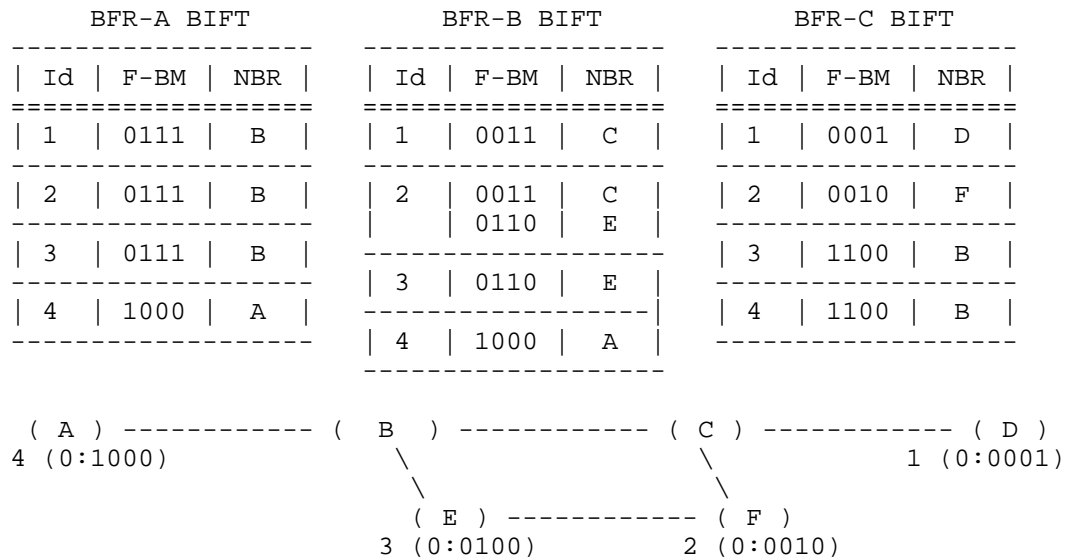


Figure 6: Example of ECMP

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.

- o 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.
- o All the packets in this sequence have the same entropy value, call it "Q".
- o 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:

- o Beginning at time t0, the multicast flow in question needs to be received ONLY by BFER-F;
- o Beginning at a later time, t1, the flow needs to be received by both BFER-D and BFER-F.
- o 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 t_0 until t_1 , the flow will travel to F via the path A-B-E-F. From t_1 until t_2 , the flow will travel to F via the path A-B-C-F. And from t_2 , 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 X_1 and X_2), and four paths to destination Y (call them Y_1 , Y_2 , Y_3 , and Y_4). If there are, say, four BIFTs, one BIFT would have paths X_1 and Y_1 , one would have X_1 and Y_2 , one would have X_2 and Y_3 , and one would have X_2 and Y_4 . 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.

1. BFR B looks in turn at each of B's child nodes on the BIER-SPF tree.
2. 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.

3. 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. (If for some reason the unicast tunnel cannot be an MPLS tunnel, any other kind of tunnel can be used, as long as it is possible to encapsulate MPLS within that kind of tunnel.)

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.

A simpler variant of step 2 above would be the following:

If one of the child nodes does not support BIER, BFR B removes that node from the tree. All BFRs that are reached through that child node are then re-parented on the tree, so that BFR B now becomes their parent.

This variant is simpler because the set of BFERs that are reached through a particular child node of B can be determined from the F-BM in the BIFT. However, if this variant is used, the results are less optimal, because packets will be unicast directly from B to the BFERs that are reachable through the non-BIER child node.

When using a unicast MPLS 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:

- o BFR-A MAY discard the packet.
- o 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.)
- o 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, Xiaohu Xu 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-smg-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.

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 June 7, 2015.

Copyright Notice

Copyright (c) 2014 IETF Trust and the persons identified as the document authors. All rights reserved.

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

Table of Contents

1. Introduction	2
2. The BIER-MPLS Label	3
3. BIER Header	5
4. Imposing and Processing the BIER Encapsulation	8
5. IANA Considerations	10
6. Security Considerations	10
7. Acknowledgements	10
8. Contributor Addresses	10
9. References	12
9.1. Normative References	12
9.2. Informative References	12
Authors' Addresses	12

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:

- o an MPLS label (which we will call the "BIER-MPLS label"); this label appears at the bottom of a packet's MPLS label stack.
- o 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 BFRs 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.

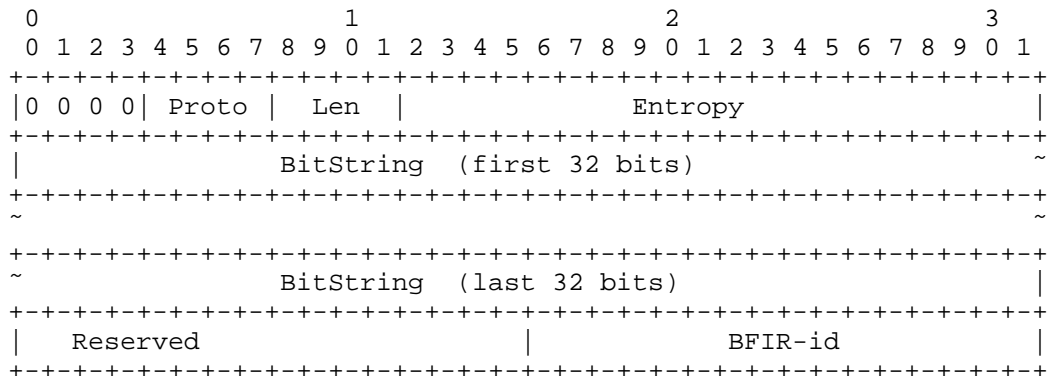


Figure 1: BIER Header

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 BFIR-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 $\log_2(k)-5$. 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 BFRs 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 BFRs.
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 BFRs:
 - 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 BFRs 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 BFR 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 prepended to the packet. This label stack [RFC3032] will contain one label, the aforementioned BIER-MPLS label. The "S" bit MUST be set, indicating the end of the MPLS label stack. The TTL field of this label stack entry is set according to policy. 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:

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