

Workgroup: Network Working Group

Internet-Draft: draft-xu-msr6-rbs-01

Published: 30 March 2022

Intended Status: Standards Track

Expires: 1 October 2022

Authors: B. Xu      X. Geng      T. Eckert

         Huawei      Huawei      Futurewei Technologies

## **RBS(Recursive BitString Structure) for Multicast Source Routing over IPv6**

### **Abstract**

This document defines a new type of segment: End.RBS, and the corresponding packet processing procedures over the IPv6 data plane for the MSR6(Multicast Source Routing over IPv6) TE solutions.

### **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 <https://datatracker.ietf.org/drafts/current/>.

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

This Internet-Draft will expire on 1 October 2022.

### **Copyright Notice**

Copyright (c) 2022 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with

respect to this document. Code Components extracted from this document must include Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

## Table of Contents

- [1. Introduction](#)
- [2. Terminologies](#)
- [3. Explicit Multicast Path with RBS](#)
  - [3.1. RBS Architecture](#)
  - [3.2. Recursive encoding in packet](#)
  - [3.3. RBS BIFT](#)
- [4. End.RBS Segment Definition](#)
- [5. RBS Sub-TLV](#)
- [6. Illustration](#)
- [7. IANA Considerations](#)
- [8. Security Considerations](#)
- [9. Acknowledgements](#)
- [10. Normative References](#)
- [Authors' Addresses](#)

## 1. Introduction

MSR6(Multicast Source Routing over IPv6) is an IPv6 based multicast source routing (MSR6) solution, defined in [[I-D.cheng-spring-ipv6-msr-design-consideration](#)], which leverages the benefits of source routing over IPv6 data plane to provide simplified multicast TE and BE service in an IPv6 network without unnecessary multicast tree status and complex control plane protocols. MSR6 needs to reuse the advantages of SRv6 and BIER to implement source routing.

MSR6 has two basic modes of forwarding: one is based on Shortest Path First(SPF), which is called MSR6 BE mode; the other is based on traffic engineered, which is called MSR6 TE mode. [[I-D.geng-msr6-traffic-engineering](#)], [[I-D.chen-pim-srv6-p2mp-path](#)] and [[I-D.geng-msr6-rlb-segment](#)] have introduced structured segment list by defining arguments in each segment.

This document defines IPv6 based RBS [[I-D.eckert-bier-cgm2-rbs](#)] which provides an optional solution for MSR6 TE. A new type of segment End.RBS and the corresponding RBS type sub-TLV in MRH defined in [[I-D.geng-msr6-traffic-engineering](#)], which could indicate multicast tree in the a recursive bitstring and save the header overhead.

## 2. Terminologies

MSR6: Multicast Source Routing over IPv6, defined in .

MRH: Multicast Routing Header, a new type of Routing Header which is used for MSR6 [[I-D.cheng-spring-ipv6-msr-design-consideration](#)].

Replication Endpoint: the intermediate node of a multicast tree, which replicates packet and forwards the packet to the downstream nodes. For MSR6, the Replication Node is called Replication Endpoint which can be indicated by the MSR6 Segment and replicate packets according to the multicast source routing information encapsulation in the MSR6 header of the packet.

BFR: Bit-Forwarding Router, a router support RBS.

BIFT: Bit Index Forwarding Table, locally to BFR.

RU: RecursiveUnit, a Bit String is to be parsed by BFR along the multicast tree of the packet, defined in [\[I-D.eckert-bier-cgm2-rbs\]](#)

### 3. Explicit Multicast Path with RBS

This section describes the encoding of explicitit multicast path with RecursiveUnit BitString Structure (RBS) .

### 3.1. RBS Architecture

An explicit multicast path is encapsulated with RBS as shown in Figure 1.

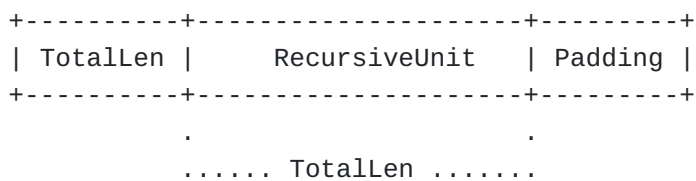


Figure 1: Architecture of RBS Address

For the reference encoding, TotalLen is an 16-bit field that counts the size of the RecursiveUnit in bits, permitting for up to 65535 Bit long RBS addresses.

The Rsv filed, which is defined in [\[I-D.eckert-bier-cgm2-rbs\]](#) , is omitted in this scenario.

Padding is used to align the RBS address as required by the IPv6 encapsulation.

### 3.2. Recursive encoding in packet

This section uses a hierarchical multicast tree as an example to describe the RecursiveUnit coding format.

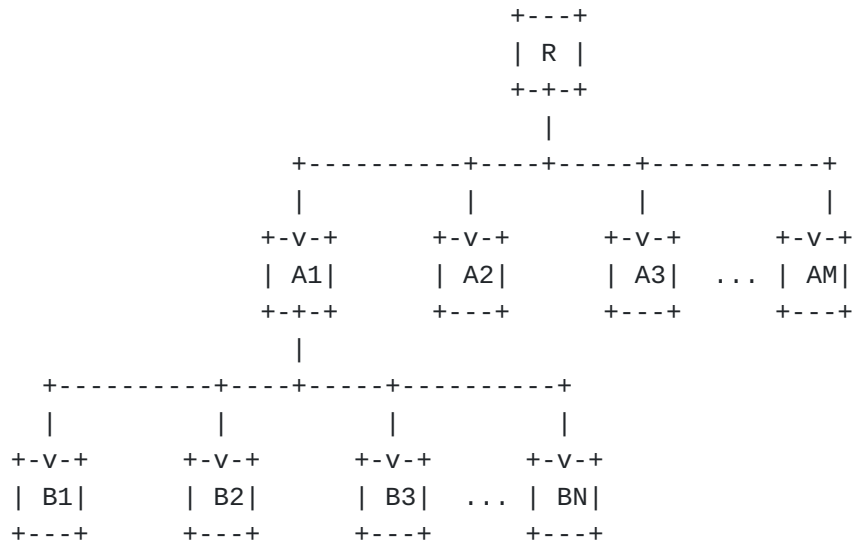


Figure 2: Hierarchical multicast tree

As Shown in Figure 2, the whole explicit multicast path should be encapsulated (See Section 3.1) in-packet, which will be parsed by each Router along the delivery tree.

The RecursiveUnit filed is structured as shown in Figure 3. To abbreviate the size of the figure, we use AF for AddressingFiled, and RU for RecursiveUnit In the following figures.

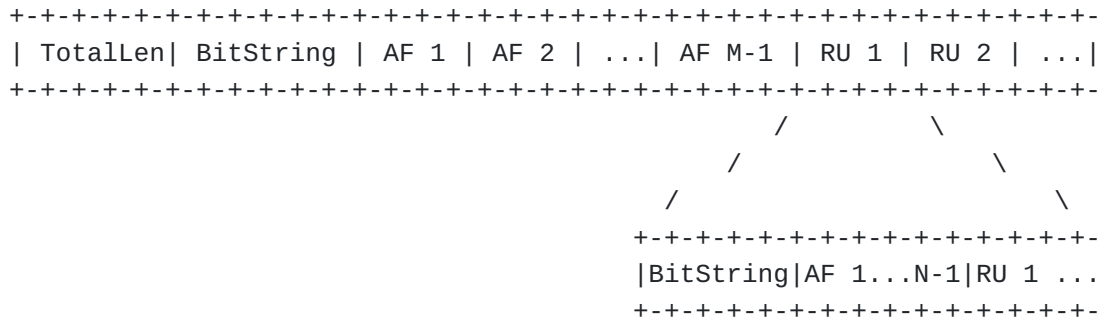


Figure 3: RecursiveUnit Filed Structure

The BitString field guides the first-hop node 'R' to locally duplicate packets and forwarding. The length of BitString matches the Maxnumber of adjacencies in node 'R' (See Section 3.3).

The AddressingField consists M-1 fields. Each filed is an 8-bits filed and the value of it is the length of relative RecursiveUnit, and may be the offset in some scenario . The length of last RecursiveUnit M could be caculated by TotalLen.

And each RecursiveUnit is structured in same mechnism as shown in Figure 3.

### 3.3. RBS BIFT

RBS BIFT as shown in Figure 4 are containing for each BP an adjacency.

```

+---+-----+-----+
|BP|RecuFlag | Adjacency|
+---+-----+-----+
| 1|          1|adjacent BFR|
+---+-----+-----+
| 2|          0|  punt/host|
+---+-----+-----+
|   .....   ...   |
+---+-----+-----+
| N|          ...|          ... |
+---+-----+-----+

```

Figure 4: RBS BIFT

The BP of the BIFT are all local to the BFR. When a BFR receives a packet encapsulated with RBS, it expects that the BitString field length must be matched with N, which is configured by BFR.

## 4. End.RBS Segment Definition

When the packet is received by an Replication Endpoint and the DA of this packet is a local SID with the function of End.RBS, the packet will be replicated based on the RBS sub-TLV defined in section 5. The DA of the replicated packets is replaced by the End.RBS for the next Replication Endpoints.

The behavior of End.RBS is defined in section 5 of [[I-D.eckert-bier-cgm2-rbs](#)].

## 5. RBS Sub-TLV

MRH defined in [[I-D.geng-msr6-traffic-engineering](#)] is as follows:

```

      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
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| Next Header | Hdr Ext Len | Routing Type | Segments Left |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| MRH Sub-type |              Reserved              |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
//                                                    //
//          Optional Type Length Value objects (variable)          //
//                                                    //
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

Figure 5.MRH of RBS Type Encapsulation

MRH Sub-type: 8-bit identifier of the sub-type. The sub-type of RBS is to be assigned by IANA.

Segments Left: MUST be set to 0 when the MRH sub-type is RBS sub-type.

Type Length Value objects: Must have RBS sub-TLV when the MRH sub-type is RBS sub-type.

A "RBS" type sub-TLV is defined for RBS in the feild of Optional Type Length Value Objects. The format is shown as below

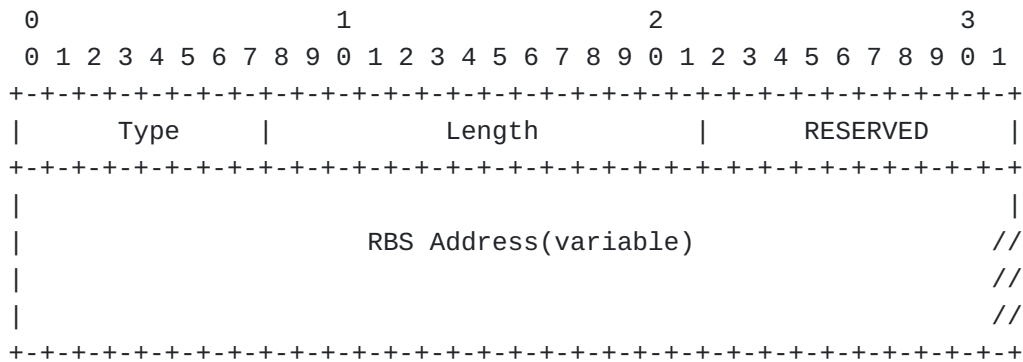


Figure 6.RBS Sub-TLV

Type: 8-bit identifier of the type of sub-TLV. The type of RBS option is to be assigned by IANA.

Length: 16-bit unsigned integer indicates the length of the option Data field of this option, in octets. The value of Opt Data Len of RBS option depends on the encoding of multicast tree, according to the mechanism defined in section 3.

RBS Address: defined in [[I-D.eckert-bier-cgm2-rbs](#)]. The packet is forwarded based on the multicast tree indicated by the RBS Address.

## 6. Illustration

Figure 7 shows an example for RBS forwarding.

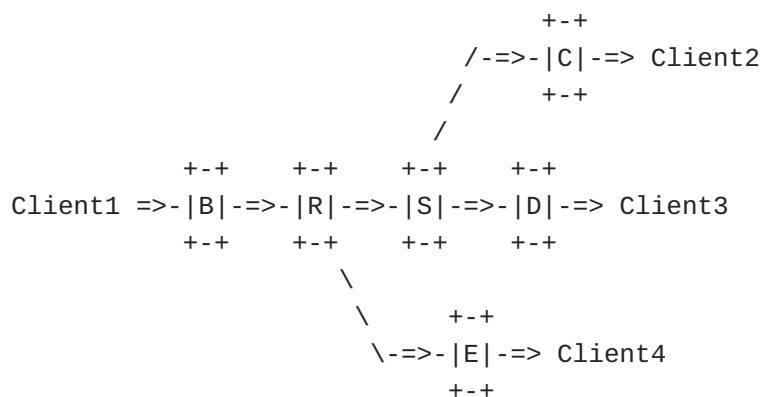


Figure 7: Example Network Topology

A packet from Client1 connected to BFR B is intended to be replicated to Client2,3,4.

The encapsulation of RBS at BFR-B is shown in Figure 8.

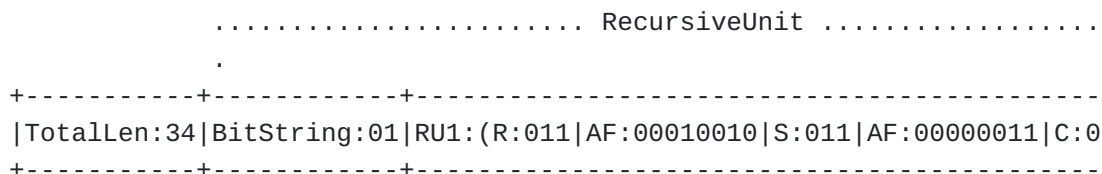


Figure 8: Encapsulation of RBS at BFR-B

Since there is only one RecursiveUnit, the AddressingField is omitted at BFR-B.

BFR-B rewrites the RBS by replacing the RecursiveUnit with RecursiveUnit 1 and adjusts the TotalLen and Padding fields.

And BFR-R receives the packet with RBS, which has been processed by BFR-B, shown in Figure 9.

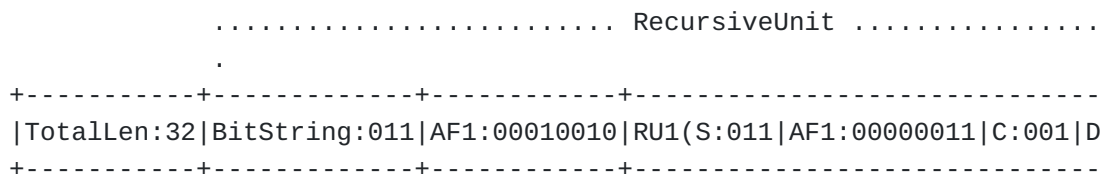


Figure 9: Encapsulation of RBS at BFR-R

And BFR-R parse the Bitstring field using BIFT shown in Figure 10.

Because there are two recursive BP set in the BitString for R, one AddressingFiled is required to indicate the length of RecursiveUnit 1.

```

+---+-----+-----+
|BP|RecuFlag |Adjacency|
+---+-----+-----+
| 1|          1|        B |
+---+-----+-----+
| 2|          1|        S |
+---+-----+-----+
| 3|          1|        E |
+---+-----+-----+

```

Figure 10: RBS BIFT on BFR-R

BFR-R accordingly creates one copy for BFR-S using RecursiveUnit 1, and only copy for BFR-E using RecursiveUnit 2, updating Padding accordingly for each copy.

BFR-S receives from BFR-R the packet as shown in Figure 11.

```

..... RecursiveUnit .....
.
+-----+-----+-----+-----+-----+
|TotalLen:18|BitString:011|AF1:00000011|RU1(C:001)RU2(D:0001)|Padding|
+-----+-----+-----+-----+-----+

```

Figure 11: Encapsulation of RBS at BFR-S

BFR-E receives from BFR-R the packet as shown in Figure 12.

```

+-----+-----+-----+
|TotalLen:32|BitString:001|Padding|
+-----+-----+-----+

```

Figure 12: Encapsulation of RBS at BFR-E

BFR-E would impose or rewrite a unicast encapsulation to make the packet become a unicast packet directed to Client 4.

The procedures for processing of the packet on BFR-S are very much the same as on BFR-R.

BFR-C receives from BFR-R the packet as shown in Figure 13. And it will make the packet become a unicast packet directed to Client 2.

```

+-----+-----+-----+
|TotalLen:3 |BitString:001|Padding|
+-----+-----+-----+

```

Figure 13: Encapsulation of RBS at BFR-C

BFR-D receives from BFR-R the packet as shown in Figure 14. And it will make the packet become a unicast packet directed to Client 3.



```

+-----+-----+-----+
|TotalLen:4 |BitString:0001|Padding|
+-----+-----+-----+

```

Figure 14: Encapsulation of RBS at BFR-D

The brief of RBS BitString conversion is shown in Figure 15.

```

+-----+
|{S=S
+-----+
|[BitS
+=====
| (C-M
+=====
/-----
/-----
+-----+
|{S=B , D=R} | |{S=R , D=S} | / |{S=S ,
+-----+
|[BitStr=011]| |[BitStr=011]| / |[BitStr
+=====+
|(C-MC Pkt)| |(C-MC Pkt)| |(C-MC Pkt)| / |(C-MC
+=====+ +-+ +=====+ +-+ +=====+ +-+ +=====+
Client1---=>---|B|-----=>-----|R|-----=>-----|S|-----=>---
+-+ +-+ +-+
+-----+ \
|{S=R , D=E}| \ +-+
+-----+ \--=>--|E|-----=>----- Client4
|[BitStr=001]| +-+ +=====+
+=====+ |(C-MC Pkt)|
| (C-MC Pkt) | +=====+
+=====+

```

Figure 15: Brief of RBS BitString coversion

## 7. IANA Considerations

This document makes no request of IANA.

Note to RFC Editor: this section may be removed on publication as an RFC.

## 8. Security Considerations

## 9. Acknowledgements

## 10. Normative References

[I-D.chen-pim-srv6-p2mp-path]

Chen, H., McBride, M., Fan, Y., Li, Z., Geng, X., Toy,  
M., Mishra, G. S., Wang, A., Liu, L., and X. Liu,

"Stateless SRv6 Point-to-Multipoint Path", Work in Progress, Internet-Draft, draft-chen-pim-srv6-p2mp-path-05, 11 November 2021, <<https://www.ietf.org/archive/id/draft-chen-pim-srv6-p2mp-path-05.txt>>.

**[I-D.cheng-spring-ipv6-msr-design-consideration]**

Cheng, W., Mishra, G., Li, Z., Wang, A., Qin, Z., and C. Fan, "Design Consideration of IPv6 Multicast Source Routing (MSR6)", Work in Progress, Internet-Draft, draft-cheng-spring-ipv6-msr-design-consideration-01, 25 October 2021, <<https://www.ietf.org/archive/id/draft-cheng-spring-ipv6-msr-design-consideration-01.txt>>.

**[I-D.eckert-bier-cgm2-rbs]** Eckert, T. and B. (. Xu, "Carrier Grade Minimalist Multicast (CGM2) using Bit Index Explicit Replication (BIER) with Recursive BitString Structure (RBS) Addresses", Work in Progress, Internet-Draft, draft-eckert-bier-cgm2-rbs-01, 9 February 2022, <<https://www.ietf.org/archive/id/draft-eckert-bier-cgm2-rbs-01.txt>>.

**[I-D.geng-msr6-rlb-segment]** Geng, X., Li, Z., and J. Xie, "RLB (Replication through Local Bitstring) Segment for Multicast Source Routing over IPv6", Work in Progress, Internet-Draft, draft-geng-msr6-rlb-segment-00, 10 February 2022, <<https://www.ietf.org/archive/id/draft-geng-msr6-rlb-segment-00.txt>>.

**[I-D.geng-msr6-traffic-engineering]** Geng, X., Li, Z., and J. Xie, "IPv6 Multicast Source Routing Traffic Engineering", Work in Progress, Internet-Draft, draft-geng-msr6-traffic-engineering-01, 7 March 2022, <<https://www.ietf.org/archive/id/draft-geng-msr6-traffic-engineering-01.txt>>.

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

## Authors' Addresses

Bing Xu  
Huawei

Email: [bing.xu@huawei.com](mailto:bing.xu@huawei.com)

Xuesong Geng  
Huawei

Email: [gengxuesong@huawei.com](mailto:gengxuesong@huawei.com)

Toerless Eckert  
Futurewei Technologies

Email: [tte@cs.fau.de](mailto:tte@cs.fau.de)