

PIM
Internet-Draft
Intended status: Informational
Expires: 22 January 2026

T. Eckert, Ed.
Futurewei Technologies USA
Y. Liu
China Mobile
Y. Zhu
China Telecom
C. Lin
New H3C Technologies
21 July 2025

Proposal for experimentation with stateless IPv6 multicast source
routing in IPv6-only SRv6 networks via new IPv6 Multicast Routing
Headers (MRH)
draft-eckert-pim-mrh-experiment-02

Abstract

This document proposes experimentation to evaluate benefits and feasibility of an IPv6 routing extension header based architecture, implementation and deployment of stateless IPv6 multicast specifically for IPv6 only networks using SRv6 for IP unicast.

This experimentation intends to explore options to support easier proliferation of technologies developed by BIER-WG by providing an IPv6/SRv6 network optimized packet header and per-hop forwarding mechanisms than BIERin6. It also discusses how this work related to exploring advanced in-packet tree encoding mechanisms for both IPv6/SRv6 networks as well as BIER networks as a related effort.

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 22 January 2026.

Copyright Notice

Copyright (c) 2025 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. Overview	2
2. Background: The road towards BIER	4
3. BIER for IPv6 networks	8
4. Challenges of BIER with IPv6	9
4.1. Operational, architectural, performance	9
4.2. Architectural and functional	10
4.3. IETF process	10
5. List of target benefits / directions	11
6. Initial proposed MRH definitions	13
6.1. MRH Header	13
6.2. BIER MRH Sub-Type format	14
6.3. Further MRH Sub-Type considerations	15
7. Informative References	16
Appendix A. Changelog	18
Appendix B. History	19
Authors' Addresses	19

1. Overview

This document proposes experimentation to evaluate benefits and feasibility of an IPv6 routing extension header based multicast forwarding. Experimentation should include implementation and experimental deployment to evaluate the feasibility and benefits of this approach for IPv6 only networks, especially those using SRv6 for unicast compared to alternative approaches such as using [I-D.ietf-bier-bierin6].

This document is mostly meant to be a process document to vet support for the direction, and if that exists, it would be followed up by an appropriate IPv6 extension header draft proposal as well as other deemed necessary documents, such as an architectural document to describe howto apply the SRv6 SID semantic to the addresses used (or indicated) vby the IPv6+extension header for stateless multicast.

The new IPv6 routing extension header would initially be using using one or two of the [RFC4727] Experiment Routing Types (253, 254). Upon successfull completion of experimentation, assignment of a permanent Routing Type could be requested.

The new routing extension header should inherit all re-useable aspects of the pre-existing unicast routing headers (RPL source route header and SRH), so that no unnecessary re-invention of already applicable SRv6 technology is done.

Likewise, the new routing extension header should in one option explore how to best re-use the IETF BIER established stateless multicast forwarding architecture, [RFC8279] while also complying to [RFC8200], and only exceeding it, when seen beneficial.

Experimentation is also meant to investigate how multiple different encodings of the stateless multicast forwarding information could best be encoded in such multicast routing extension header options; these are henceforth called MRH - Multicast Routing Header.

For example, different encodings could use different Routing Type code points as used for example across the different compressed unicast routing header options such as CRH-16, CRH-32. In another option, different encoding options could be selected from a further code-point within a single MRH Routing Type header.

The experimentation should include two different encoding options. One is the aforementioned [RFC8296] architecture based stateless multicast forwarding information (bitstring plus metadata).

The second encoding should be an initial proposed encoding for stateless tree encoding within the MRH, such as derived from, but not necessarily [I-D.eckert-pim-rtts-forwarding]. Note that like also in unicast SRv6 and its various forms of compressed routing header options, ultimtely different type of networks may prefer different encodings for multicast, and hence ensuring the design for extensibility is one core goal of this experimentation.

This document does not (yet) propose an exact list of target documents required to perform the experiment or which WG it should best happen, but observes the following:

- * 6MAN-WG is the responsible working group for IPv6 extension headers, but it requires a show of sufficient demand for the extension header in technology cases like this. One of the goals of the initial work with 6MAN-WG should be to determine how much of the encoding variations could be done without repeated work for 6MAN-WG, but instead by only work of the more expert work for such encodings, such as BIER. This would follow the logic, by which different semantics of QoS information in the DSCP field where also not performed by 6MAN-WG but other working groups (such as TSVWG).
- * SPRING-WG is the overall responsible WG for the IPv6/SRv6 architecture as well as its unicast forwarding designs. SPRING-WG has delegated responsibility for IPv6 multicast aspects to PIM-WG. Hence this document is also positioned for PIM-WG for first considerations.
- * BIER-WG is the working group which has introduced stateless multicast replication to the IETF, and where all the experience with hardware forwarding implementation for stateless multicast replication exists. Given how the goal of this experimentation is to not re-invent any aspect of BIER-WG except for those aspects that will support better adoption of the technology to IPv6-only/SRv6 networks, it is highly desirable to involve BIER in any of the aspects of the experiment where this expertise can be drawn upon and where consistency with existing BIER approaches needs to be vetted.
- * New encoding options for stateless multicast forwarding should not only be defined for encapsulation within an IPv6 MRH, but if BIER-WG is also interested in that encoding, then the encoding should be specified agnostic to header encoding, such that it could support encapsulation both into IPv6/MRH as well as BIER/[RFC8296] headers.

2. Background: The road towards BIER

This background section is intended as an overview to motivate why operators have shown interest in an IPv6/SRv6-SRH aligned solution for stateless multicast for many operational reasons, and why BIER-WG did arrive at a different conclusion for technology when adopting [I-D.ietf-bier-bierin6]. This is written in the hope that the experimentation proposed by this document is understood as a complementary technology that is ultimately meant to proliferate the BIER-WG spearheaded technology into more markets with minimum additional standardization and development effort.

Before BIER, multicast solutions were defined as extensions of the unicast (inter)network solution run in the network. RFC1112 defined IP Multicast which re-used IPv4 ([RFC791]) packet headers and added multicast semantic through a range of dedicated-to-multicast IPv4 destination address range. IPv6 kept this architecture but already included it in [RFC2460]. In result, so-called dual-plane IPv4/IPv6 network did also have to run dual-plane IPv4/IPv6 multicast. In networks that only could or wanted to run a single version of IP, solutions for encapsulation of one version over the other also exist for both unicast and multicast.

In MPLS networks, initially IP multicast was run by using IPv4 multicast in parallel to MPLS for unicast, but operators expected that the protocol stack for multicast both in the forwarding plane as well as the control plane was adjusted to leverage the same protocols as for MPLS unicast, so that at least in name the variety of protocols that needed to be run in the network for unicast and multicast was not increased over an MPLS unicast only network. Even though implementations of PIM for an MPLS forwarding plane already existed since 1998, the PIM approach was explicitly not chosen because this was a protocol unfamiliar to MPLS network operators. Instead, the industry and thereafter the IETF chose to embed the necessary multicast functionality into the dominant MPLS unicast protocols LDP and RSVP-TE. This resulted in mLDP (multicast LDP) and RSVP-TE/P2MP (Point to Multipoint).

Likewise, PIM overlay signaling for multicast VPN services was also re-invented due to operator demand by bringing similar multicast group membership signaling into BGP, thus allowing service providers to completely run on only BGP and IGP but without PIM for multicast. Whether the forwarding plane uses IPv4/IPv6 or MPLS. This was done even though no other than operational reasons of familiarity by operators with BGP were brought forward versus the already well deployed option of using PIM for overlay signaling in those VPN solutions.

While this approach of having the multicast solution be embedded in the networks unicast solution does have a wide range of benefits, it also came with downsides. When classical MPLS solutions with LDP were superseded by SR-MPLS network solution for unicast, this was also done to eliminate LDP from the network, which had a range of problems co-existing with unicast IGPs, and/or arguably unnecessary complexity and duplication of protocol state. But in the wake of this unicast network architecture evolution towards unicast SR-MPLS, operators also asked to eliminate multicast MPLS using mLDP and RSVP-TE/P2MP.

More fundamentally, multicast solutions including all the aforementioned ones are based on explicit multicast tree state, which is managed hop-by-hop in the network. This is completely contrary to what operators are used to do with MPLS or IP unicast. In unicast, the network, especially the transit nodes (called P-nodes in service provider networks) only carry topology state (routing tables), but not state belonging to or influenced by individual subscribers of the network. Subscribers may send arbitrary traffic, but that will not impact the IP/MPLS unicast routing tables on P-nodes.

In IP/MPLS multicast, this is not the case. When a subscriber starts multicast sender and receiver applications, then they will cause mLDP, PIM or BGP signaling to propagate through the network including transit service provider hops and ultimately create multicast tree state, which is similar in nature to unicast routing tables: It requires hardware forwarding resources that can get exhausted, and it requires control plane activity that may put undesirable load onto the control plane CPU not only during creation or change of the applications, but even more so during reconvergence due to topological changes in the network (failures, recoveries).

While advanced multicast VPN protocol options do allow service providers to put bounds on this state (I-PMSI state and aggregated S-PMSI state), this too is causing additional complexity and diagnostic/troubleshooting overhead.

Even when there is no misbehavior in the network, keeping track and troubleshooting multicast state hop-by-hop in the network can be a real operational challenge, and even with aforementioned multicast VPN mechanisms in place, it still is the equivalent of having unicast (BGP) VPN subscriber routing table entry related state on P-routers, whereas in unicast those P-routers only carry a completely subscriber agnostic IGP routing table.

In result of all these operational experiences by operators, several of them opted to move from their prior (stateful) multicast solution to one where there would be no multicast state at all across the service provider core network, instead replicating multicast traffic on the ingress PE as unicast traffic, one copy each to each egress PE that required the packet. This was much later standardized as ([RFC7988]). This is called "multicast ingress replication"

BIER was invented out of the early recognition of these operational challenges and in fear that the non-scalability of multicast ingress replication would ultimately lead to the demise of solutions relying on network based multicast. The BIER architecture, [RFC8279] defines a mechanism by which a packet header which includes a bitstring and associated metadata can source-route and replicate the packet hop-by-

hop across P-routers to a set of egress (PE) routers using only the same type of routing information as already present in a service provider network - addressing information for the core networks P/PE routers specific to BIER, but not to any subscriber information.

With BIER, no multicast tree state ever needs to exist on the transit (P) routers. The BIER technology also makes it easy to evolve from multicast ingress replication solutions, by simply adding the BIER to the P/PE routers and letting the ingress router determine the BIER header bitstring instead of creating a separate copy per packet to each required PE. Even partial deployment of BIER across P/PE routers is well considered in the BIER architecture and feasible automatically without additional provisioning, albeit this is likely not fully supported by existing implementations today.

The BIER architecture does not specify the way such a BIER header was to be packetized. Initially during the work on BIER it was seen as a real possibility to create per-unicast-network-design encapsulations as it was done in before for IPv4/IPv6 and MPLS. Nevertheless, when work towards the first BIER encapsulation(s) progressed, it became obvious that the definition of multiple different encapsulations was at the non-mature and non-adopted state of the technology too ambitious and time consuming (note that the same may be said about the current evolution towards multiple compressed unicast routing headers).

Instead, BIER-WG chose an approach where the metadata required for different type of networks was coalesced into a single BIER header approach, which ultimately became [RFC8296] - "Encapsulation for BIER in MPLS and Non-MPLS Networks". This header includes a DSCP field for use in IP network, and the first 32 bits mimic exactly a 32-bit word of an MPLS stack.

When BIER packets are propagated through an MPLS network, there is no end-to-end MPLS label stack or even MPLS "packet" in that BIER packet. Logically the per-hop BIER forwarding happens solely at the BIER layer, and on every hop the packet is encapsulated into a single-LSE MPLS frame. In a network with full BIER support, where BIER routers are L2 adjacenty, this is functionally equivalent to simply encapsulate the BIER packet into an ethernet frame with a BIER ethernet type. The core reasons for the MPLS encapsulation was the MPLS network operator preference to only see MPLS frames on the wires and the possible simplification of MPLS centric router hardware forwarding plane designs to demultiplex packets easier on an LSE than on an ethertype. An actual functional benefit of the MPLS encapsulation exists, in a partial deployment, when the next BIER router is not L2 adjacent, and the MPLS label could for example be an SR-MPLS label to the closest BIER capable router.

As a result of this originally "optimized-for-MPLS" approach, architecturally [RFC8296] marked for BIER the desire to embrace an approach where the BIER multicast solution should be seen as a "one-size-fits-all" multicast network designs: To avoid the problem that any change/evolution in unicast forwarding technologies would create undesirable asks for change in the multicast technology - and hence cause unnecessary technology churn.

Of course, this approach for the BIER forwarding plane does not prevent possible churn on the necessary BIER control plane. If for example use of IGP in networks (as is the standard today in SR networks) should fall out of fashion, and would be replaced by some (maybe AI controlled ?) SDN-controller control plane, then the same would be necessary also for BIER.

3. BIER for IPv6 networks

Operators of IPv6-only networks using SRv6 for unicast traffic steering and service management expressed interest in adopting stateless source-routed technologies for multicast across their networks.

The BIER architecture [RFC8279] was the obvious starting point to provide this functionality. Additional service requirements such as traffic engineering could be solved via BIER extensions such as BIER-TE ([RFC9262]) without IPv6 specific changes required.

Some SRv6 networks are amongst the largest Service Provider networks in the world, hence there is likely going to be more of an upper-end scaling evaluation to be done for SRv6 networks.

However, the core challenge for IPv6-only network operators comes through the "one-size-fits-all" model adopted via [RFC8296] by BIER. By itself, this RFC is insufficient to operate BIER in a network without MPLS. For this reason, [I-D.ietf-bier-bierin6] (BIERin6) defines how to operate BIER in an IPv6 only network based on the requirements stated in [I-D.draft-ietf-bier-ipv6-requirements]. This approach repeats the MPLS network approach for BIER in IPv6 networks: end-to-end forwarding is via an [RFC8296] BIER header and BIER hop-by-hop an IPv6 specific IPv6 "tunnel" header can be added if that is so desired, to make the packet appear as IPv6 on the wire. In a full deployment those L2 IPv6 encapsulations could even use link-local IPv6 addresses.

In addition, the encoding and control plane signaling of the BIER metadata that needs change from its [RFC8296] MPLS bindings also requires new drafts, [I-D.ietf-bier-non-mpls-bift-encoding] and [I-D.ietf-bier-lsr-non-mpls-extensions].

4. Challenges of BIER with IPv6

4.1. Operational, architectural, performance

The main challenge is that the requirements that lead to BIERin6 did not take the operational and architectural preferences of operators running IPv6-only networks into account by replicating the model developed with primarily MPLS in mind. For operators of IPv6/SRv6 networks, BIERin6 marks a significant departure from their unicast IPv6 network architecture and operations.

One key difference between IPv6 and MPLS designs is that the cost of headers and encaps/decap. When operators in an IPv6 network expect for operational reasons to see only IPv6 packets on the wire, the BIERin6 approach would require per-L2-hop additional IPv6 tunnel encapsulations, which is a significant overhead, whereas the MPLS equivalent is no additional encapsulation overhead at all, because the BIER header itself already start with a 32 bit word that serves as an MPLS LSE and single-entry label stack to demultiplex to BIER.

If instead (as proposed to be experimented in this document), the [RFC8200] approach of IPv6 extension header source-routing was used, then this additional IPv6 per-hop encapsulation header was not required, but the end-to-end IPv6 header (plus extension header) that is always required would suffice.

Note too, that this difference also holds true, when there is only a partial deployment of stateless multicast replication capable routers, because by virtue of the pre-existing end-to-end IPv6 header and [RFC8200] defined forwarding rules, the forwarding across non-stateless-multicast-capable IPv6 routers would solely require forwarding based on the IPv6 destination address of the IPv6 (base) header.

In addition to overhead on the wire, different type of forwarding planes may also incur different degrees of performance loss when per L2-hop packets actually need to be decapsulated and re-encapsulated into link-local address IPv6 headers. Keeping an encapsulation IPv6 header and re-writing it hop-by with not only the destination address (as required by [RFC8200], but also the source-address might equally be a challenge if hardware is not specifically optimized for that. The author has the unfortunate experience with this type of hop-by-hop IPv6 tunneling in a completely different use-case domain, where it has served as the core obstacle to adoption (see [RFC8994]).

4.2. Architectural and functional

While these differences between BIERin6 and extension header based approaches are mostly router-implementation, performance and operator preference, there is also a more fundamental issue with BIER which the proposed extension header resolves, and that is that the packet on the wire can not be identified as an IPv6 multicast packet, and hence the wide range of collateral forwarding plane functions that have already been defined in routers to manage IPv6 multicast traffic are not applicable or would require a lot more complex, variable length lookup of the IPv6 multicast destination address across protocol layer boundaries. These features are listed further below in the document.

Another way to look at this functional difference is that BIER was designed to require an IP Multicast flow overlay so that BIER could - maybe similar to MPLS - be a common "L2.5" technology, whereas IPv6 multicast is designed as an end-to-end L3 technology just like IPv6 unicast, and the IPv6 extension header approach is the only obvious way to achieve this, and minimize or completely eliminate additional layering overheads necessary otherwise.

4.3. IETF process

In the discussions about the best BIER for IPv6 network solution, one of the foremost argument by those in favor of BIERin6 was also explicitly to prove investment protection for those vendors who had already invested into [RFC8279] and [RFC8296]. While it certainly makes sense to support commercial goals in IETF, this specific ask was never admitted in any prior technology decision between MPLS and IPv6 solutions nor for other Multicast technologies (e.g.: introducing BGP instead of PIM overlay signaling was providing all the necessary functionality, or introducing MPLS multicast forwarding when many operators had deployed native IPv4 multicast in MPLS networks). Nor was it done in the case of other technologies, such as OSPF vs. IS-IS. Instead, if and when different competing designs existed due to customer demand, the IETF always tried to provide equally optimized solutions for each customer network type and let the market decide.

While technically similar approaches may pose additional work in the IETF, they have typically always shown to be beneficial for the overall set of customers of IETF standards, broadening the scope of applicability to more candidate customers. The hope of this experimentation is equally that this would hold true for the proliferation of original BIER technologies into IPv6/SRv6 only networks - more so than what BIERin6 alone could achieve (in the opinion of the author). Expecting for BIERin6 to make BIER successful in IPv6-only/SRv6 networks is a highly risky proposition and can easily result in less than more adoption of the technology.

Hence this document proposes for the IETF to at least embrace experimentation with the IPv6 extension header based options which do provide the best technically and operational solutions for IPv6-only/SRv6 networks. Especially given the recent varied work on different unicast steering header options for SRv6 unicast, it seems highly unlikely that the industry could not endure an additional encoding alternative to [RFC8296] for stateless multicast replication in IPv6 networks.

5. List of target benefits / directions

The following is a candidate list of benefits/technical targets of the solutions

The solution uses an IPv6 routing extension header in the same fashion as SRv6 unicast does this ([RFC8754]) for high-speed networks or [RFC6554] for IoT networks. Ideally, it should be sufficient to have a single new IPv6 routing extension header for stateless multicast (instead of two for unicast for different networks).

For operational safety, it seems prudent to allocate a new routing extension header code point so that this technology can be introduced into networks which already run either of the existing unicast extension headers without having to change any of the unicast specific forwarding plane code - because demultiplexing between unicast and multicast would already be able to happen at the extension header code point.

If instead it is seen by IPv6 experts that it is equally safe and feasible to encode the information necessary for stateless multicast into the existing SRH extension header, using similar tricks to how compressed unicast steering is encoded ([I-D.ietf-spring-srv6-srh-compression]), and that this is preferable, then this could of course equally be considered.

All forwarding should follow the [RFC8200] and [RFC8986] principles to the extent that they are applicable to packets that need to be replicated. This specifically means that on each steering or replication hop, the IPv6 header destination address gets rewritten by the next steering/replication hop IPv6 address ("SID") derived from the extension header.

While application software stacks for other network protocols beside IP do exist, they are by far not as widely and easily accessible across wide ranges of platforms/operating systems. Use of IPv6 extension headers is the likely most easy proliferable solution to bring stateless multicast replication into the software realm. This is not only useful for softwareization of traditional routers with new implementations, or decomposition thereof into network function application code, but even more so for actual classical end-to-end applications using IP multicast. Enabling such applications, to solely rely on stateless multicast, for example in data centers is an explicit additional market space that this effort intends to support.

To directly support the established IP multicast service interfaces of ASM and SSM, the extension header must support to directly indicate IP multicast. Technically this means that the extension header also needs to be able to carry an IPv6 multicast destination address directly, namely when there is the need to indicate an IPv6 multicast packet. This is a significant difference over SRv6 unicast and BIER. BIER specifically does support IP multicast only in the way MPLS does it: as an L2.5 underlay, which can encapsulate an IP multicast packet. The stateless IPv6 multicast solution intends to avoid this duplication of IPv6 header when it is used end-to-end.

With this architecture, IPv6 multicast applications could be made to rely on stateless IPv6 multicast if the socket/network layer in the multicast host can simply insert the routing extension header indicating the stateless distribution tree - without the need for any additions IPv6 in IPv6 encapsulations or other complexity. This for example could easily be something to make deployment of IPv6 multicast easier in data center environments where the extension header data could be requested from and provided by a network controller.

Carrying the IPv6 multicast destination address in a fixed offset part of an IPv6 extension headers makes it possible (at somewhat higher parsing cost) to maintain traditional operational benefits of IP multicast: It allows per-hop operation of IP specific forwarding plane features:

- * IP multicast IPfix for accounting, billing and performance quality troubleshooting

- * IP multicast group address (range) derived QoS handling (common in several network and well matching [RFC8986] "programming" goals).
- * IPmulticast group range derived accounting, billing, policiing or other policies.

6. Intial proposed MRH definitions

6.1. MRH Header

This chapter gives a non-normative idea of how the extension header structures to be defined by the experiment could look like

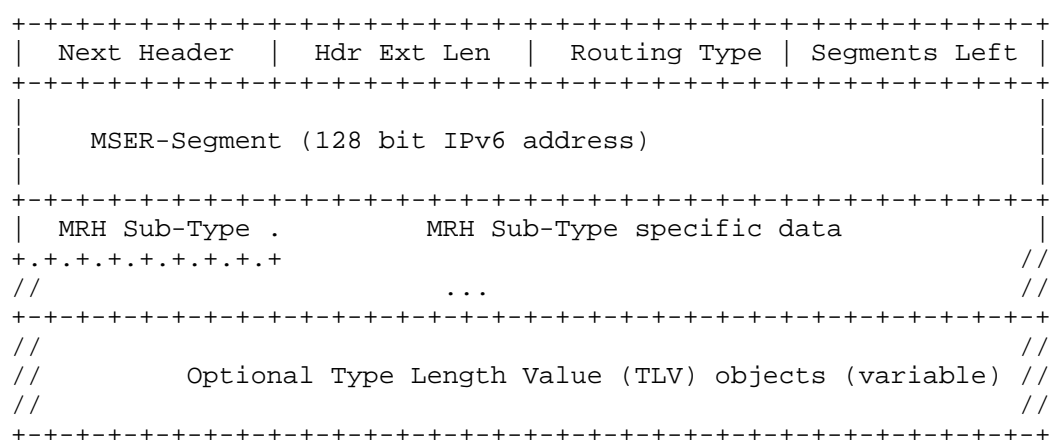


Figure 1: MRH format

The actual "Multicast Routing Header" is identified by one "Routing Type" value. If multiple encodings ("MRH Sub-Type") are required, then multiple "Routing Types" would need to be assigned (for experimentation only two are available), or instead, a new "MRH Sub-Type" demultiplexer field could be defined. With a worst-case small number of different MRH Sub-Types of maybe ≤ 5 required, it may be deemed appropriate by 6MAN-WG to allocate up to e.g.: four Routing Types before requiring a Sub-Type encoding to avoid Routing Type exhaustion.

"Segments Left" was defined by [RFC8200]. It may not actually be useful in all encodings, so one of the experimentation question is whether it would be acceptable to avoid wasting this space if it is not needed, or to reuse it for a more appropriate purpose.

The "MSER-Segment" ("Multicast Source Exit Router Segment") can carry the IPv6 multicast packets destination address. This is necessary when an MRH is to be used to carry an actual IPv6 Multicast packet. Alternatively, this MSER-Segment can carry a non-IPv6 multicast group range IPv6 address. In this case it is a group-SID, indicating for example programmability parameters to the receivers.

When deploying MRH with BIER-like overlays, the MSER Segment may not be required. If one wanted to avoid wasting those 128 bits for such use-cases, then appropriate encoding options should be found (different Routing Type). However, one of the big benefits for IPv6 from using MRH (as opposed to a BIER header) could come from the ability to always have this destination IPv6 address present in a fixed location in the IPv6 (extended) header to trigger various collateral forwarding plane functions, as mentioned elsewhere in this document. Therefore, the functional (not performance) preference would be to always include this field. Likewise, in many applications a "shared IPv6 SID" for all receivers of the packet seems likely very useful, even if its semantic is not that of an IPv6 multicast group address.

The "MRH Sub-Type specific data" if present may carry the encoding for a particular method of stateless IPv6 multicast forwarding

6.2. BIER MRH Sub-Type format

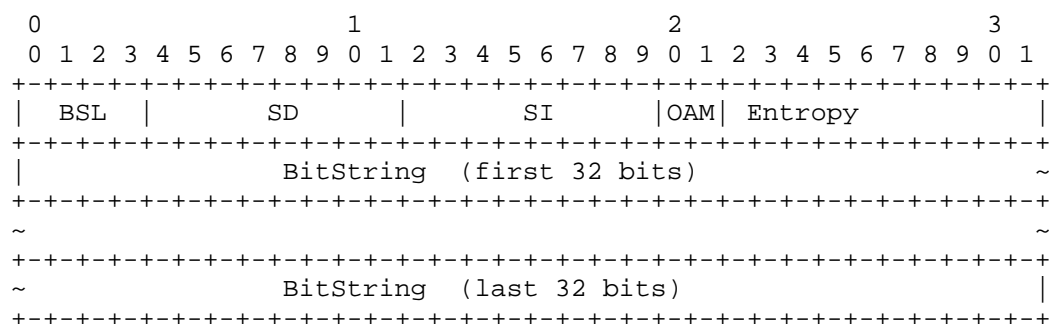


Figure 2: BIER MRH Sub-Type format

Figure 2 is an example initial proposal for encoding of BIER (and BIER-TE, [RFC9262]) into an MRH Sub-Type.

Most of the signaling elements of the [RFC8296] BIER header are not required:

BFIR-id is not required because in an IPv6 environment, the IPv6 (base) header IPv6 source address (which can be a SID) serves the same purpose.

If necessary to maintain existing signaling for 16-bit BFR-ID values, then an appropriate SID format for the 128 bit address with such a 16 bit field can be specified.

Likewise, TTL, DSCP and Ver fields are not required as they already exist in the IPv6 header.

The Proto field is not required, because the "Next Header" field in the MRH serves the same purpose.

Nibble, TC and S fields are not required because they are artefacts of the BIER header for MPLS.

The remaining signaling elements keep their existing semantic but are slightly different encoded:

[I-D.ietf-bier-non-mpls-bift-encoding] proposes to encode BSL, SD and SI into the BIFT-id field. This proposal picks up the same encoding, eliminating therefore also the separate BSL field.

OAM is maintained unchanged. BitString is maintained unchanged.

Entropy is shortened to allow fitting the non-bitstring signaling elements into 32 bits. 1024 different path options are more than enough, so no functional deterioration is expected.

6.3. Further MRH Sub-Type considerations

The encoding for other "MRH Sub-Type specific data" fields are not presented here. For example, if RTS ([I-D.eckert-pim-rtts-forwarding] was used as a Sub-Type, then that field would be encoded according to that drafts header, specified in [I-D.eckert-pim-rtts-forwarding], section 4.3.

Independent of MRH Sub-Type, the per-hop forwarding rules need to be specified, should be the same as IPv6 unicast source routing: Every hop determines for every packet copy a next-hop ipv6 next-hop address, which will be copied into the IPv6 header destination address field.

Different from IPv6 unicast, the different MRH Sub-Type encodings will require different modifications to the MRH header. In the BIER Sub-Type case for example, bits in the Bitstring need to be cleared. In headers such as RTS, it may be necessary to update two index

fields in the Sub-Type data to indicate to the next-hop the active part of the Sub-Type header that needs to be parsed. This is equivalent to the Segments-Left field in IPv6 unicast routing header cases.

7. Informative References

[I-D.draft-ietf-bier-ipv6-requirements]

McBride, M., Xie, J., Geng, X., Dhanaraj, S., Asati, R., Zhu, Y., Mishra, G. S., and Z. J. Zhang, "BIER IPv6 Requirements", Work in Progress, Internet-Draft, draft-ietf-bier-ipv6-requirements-09, 28 September 2020, <<https://datatracker.ietf.org/doc/html/draft-ietf-bier-ipv6-requirements-09>>.

[I-D.eckert-msr6-problem-statement]

Eckert, T. T., "Yet another Problem Statement for IPv6 Multicast Source Routing (MSR6)", Work in Progress, Internet-Draft, draft-eckert-msr6-problem-statement-00, 24 October 2022, <<https://datatracker.ietf.org/doc/html/draft-eckert-msr6-problem-statement-00>>.

[I-D.eckert-msr6-rbs]

Eckert, T. T., Geng, X., Zheng, X., Meng, R., and F. Li, "Recursive Bitstring Structure (RBS) for Multicast Source Routing over IPv6 (MSR6)", Work in Progress, Internet-Draft, draft-eckert-msr6-rbs-01, 24 October 2022, <<https://datatracker.ietf.org/doc/html/draft-eckert-msr6-rbs-01>>.

[I-D.eckert-pim-rtts-forwarding]

Eckert, T. T., Menth, M., Lindner, S., and Y. Liu, "Stateless Multicast Replication with Segment Routed Recursive Tree Structures (RTS)", Work in Progress, Internet-Draft, draft-eckert-pim-rtts-forwarding-03, 6 July 2025, <<https://datatracker.ietf.org/doc/html/draft-eckert-pim-rtts-forwarding-03>>.

[I-D.ietf-bier-bierin6]

Zhang, Z., Zhang, Z. J., Wijnands, I., Mishra, M. P., Bidgoli, H., and G. S. Mishra, "Supporting BIER in IPv6 Networks (BIERin6)", Work in Progress, Internet-Draft, draft-ietf-bier-bierin6-11, 2 March 2025, <<https://datatracker.ietf.org/doc/html/draft-ietf-bier-bierin6-11>>.

[I-D.ietf-bier-lsr-non-mpls-extensions]

Dhanaraj, S., Yan, G., Wijnands, I., Psenak, P., Zhang, Z. J., and J. Xie, "LSR Extensions for BIER non-MPLS Encapsulation", Work in Progress, Internet-Draft, draft-ietf-bier-lsr-non-mpls-extensions-03, 7 February 2024, <<https://datatracker.ietf.org/doc/html/draft-ietf-bier-lsr-non-mpls-extensions-03>>.

[I-D.ietf-bier-non-mpls-bift-encoding]

Wijnands, I., Mishra, M. P., Xu, X., and H. Bidgoli, "An Optional Encoding of the BIFT-id Field in the non-MPLS BIER Encapsulation", Work in Progress, Internet-Draft, draft-ietf-bier-non-mpls-bift-encoding-04, 30 May 2021, <<https://datatracker.ietf.org/doc/html/draft-ietf-bier-non-mpls-bift-encoding-04>>.

[I-D.ietf-spring-srv6-srh-compression]

Cheng, W., Filsfils, C., Li, Z., Decraene, B., and F. Clad, "Compressed SRv6 Segment List Encoding (CSID)", Work in Progress, Internet-Draft, draft-ietf-spring-srv6-srh-compression-23, 6 February 2025, <<https://datatracker.ietf.org/doc/html/draft-ietf-spring-srv6-srh-compression-23>>.

[RFC2460] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, DOI 10.17487/RFC2460, December 1998, <<https://www.rfc-editor.org/rfc/rfc2460>>.

[RFC4727] Fenner, B., "Experimental Values In IPv4, IPv6, ICMPv4, ICMPv6, UDP, and TCP Headers", RFC 4727, DOI 10.17487/RFC4727, November 2006, <<https://www.rfc-editor.org/rfc/rfc4727>>.

[RFC6554] Hui, J., Vasseur, JP., Culler, D., and V. Manral, "An IPv6 Routing Header for Source Routes with the Routing Protocol for Low-Power and Lossy Networks (RPL)", RFC 6554, DOI 10.17487/RFC6554, March 2012, <<https://www.rfc-editor.org/rfc/rfc6554>>.

[RFC791] Postel, J., "Internet Protocol", STD 5, RFC 791, DOI 10.17487/RFC0791, September 1981, <<https://www.rfc-editor.org/rfc/rfc791>>.

[RFC7988] Rosen, E., Ed., Subramanian, K., and Z. Zhang, "Ingress Replication Tunnels in Multicast VPN", RFC 7988, DOI 10.17487/RFC7988, October 2016, <<https://www.rfc-editor.org/rfc/rfc7988>>.

- [RFC8200] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", STD 86, RFC 8200, DOI 10.17487/RFC8200, July 2017, <<https://www.rfc-editor.org/rfc/rfc8200>>.
- [RFC8279] Wijnands, IJ., Ed., Rosen, E., Ed., Dolganow, A., Przygienda, T., and S. Aldrin, "Multicast Using Bit Index Explicit Replication (BIER)", RFC 8279, DOI 10.17487/RFC8279, November 2017, <<https://www.rfc-editor.org/rfc/rfc8279>>.
- [RFC8296] Wijnands, IJ., Ed., Rosen, E., Ed., Dolganow, A., Tantsura, J., Aldrin, S., and I. Meilik, "Encapsulation for Bit Index Explicit Replication (BIER) in MPLS and Non-MPLS Networks", RFC 8296, DOI 10.17487/RFC8296, January 2018, <<https://www.rfc-editor.org/rfc/rfc8296>>.
- [RFC8754] Filsfils, C., Ed., Dukes, D., Ed., Previdi, S., Leddy, J., Matsushima, S., and D. Voyer, "IPv6 Segment Routing Header (SRH)", RFC 8754, DOI 10.17487/RFC8754, March 2020, <<https://www.rfc-editor.org/rfc/rfc8754>>.
- [RFC8986] Filsfils, C., Ed., Camarillo, P., Ed., Leddy, J., Voyer, D., Matsushima, S., and Z. Li, "Segment Routing over IPv6 (SRv6) Network Programming", RFC 8986, DOI 10.17487/RFC8986, February 2021, <<https://www.rfc-editor.org/rfc/rfc8986>>.
- [RFC8994] Eckert, T., Ed., Behringer, M., Ed., and S. Bjarnason, "An Autonomic Control Plane (ACP)", RFC 8994, DOI 10.17487/RFC8994, May 2021, <<https://www.rfc-editor.org/rfc/rfc8994>>.
- [RFC9262] Eckert, T., Ed., Menth, M., and G. Cauchie, "Tree Engineering for Bit Index Explicit Replication (BIER-TE)", RFC 9262, DOI 10.17487/RFC9262, October 2022, <<https://www.rfc-editor.org/rfc/rfc9262>>.

Appendix A. Changelog

00: addd new co-authors

00: initial version.

Appendix B. History

This document is an evolution from the process whose last draft was [I-D.eckert-msr6-problem-statement]. That informational draft covers in a more concise way two core problem areas

1. The Desire and need for an IPv6 routing extension header based solution architecture for stateless IPv6 multicast source routing in support of IPv6/SSRv6-only networks. This is covered in P4...P7 and according explanationsj.
2. The desire and need for an easier to operate and better to scale stateless replication mechanism instad of [RFC8279], such as proposals like [I-D.eckert-pim-rtts-forwarding] or [I-D.eckert-msr6-rbs]. These are covered in P1...P3,P8 and according explanations.

Point 1 in that problem state document can be resolved by an IPv6 routing extension header based solution, which is what this document explains in more detail and proposes to introduce through experimental IETF work.

It is thus overlapping with that problem statments P4...P7. Such an IPv6 routing extension header based solution can and should leverage both an [RFC8279] based payload, which would allow a minimal change and new development from existing BIER specifciations, replacing only [RFC8296] as the encapsulation and replacing the need for [I-D.ietf-bier-bierin6].

Point 2 is not detaillled in this document except to explain what aspects in the new IPv6 routing extension header are required to enable such alternative encodings compared to the existing BIER bitstring.

Authors' Addresses

Toerless Eckert (editor)
Futurewei Technologies USA
United States of America
Email: tte@cs.fau.de

Yisong Liu
China Mobile
China
Email: liuyisong@chinamobile.com

Yongqing Zhu
China Telecom
China
Email: zhuyq8@chinatelecom.cn

Changwang Lin
New H3C Technologies
China
Email: linchangwang.04414@h3c.com