

ROLL
Internet-Draft
Updates: [6282](#),6550,6775 (if approved)
Intended status: Standards Track
Expires: January 25, 2019

P. Thubert, Ed.
Cisco
July 24, 2018

RPL-BIER
draft-thubert-roll-bier-02

Abstract

This specification extends RPL to provide unicast and multicast routing based on bitStrings such as used in Bit Index Explicit Replication and its source-routed Traffic Engineering variant, which correspond to RPL storing and Non-Storing Modes respectively.

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 January 25, 2019.

Copyright Notice

Copyright (c) 2018 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 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	5
3.	Applicability	5
4.	Extensions to RFC 6550	5
4.1.	RPL-BIER MOPs	6
4.1.1.	RPL-BIER Non-Storing Mode	6
4.1.2.	RPL-BIER Storing Mode	6
4.2.	BitString Information	7
5.	Updating the 6LoWPAN Framework	8
5.1.	Extensions to RFC 6775	8
5.2.	Extensions to RFC 6282	9
5.3.	New Neighbor Discovery Options and Messages	9
5.3.1.	6LoWPAN ND Bit Position Option	9
5.3.2.	Address Mapping Message	10
6.	BitString formats	11
6.1.	Bit-by-bit BitStrings	11
6.1.1.	Allocating a Bit Position in a Bit-by-bit BitString	12
6.1.2.	Aggregation of Bit-by-bit BitStrings	13
6.1.3.	Forwarding Based on Bit-by-bit BitStrings	13
6.1.4.	Reliable Multicast based on Bit-by-bit BitStrings	14
6.2.	Bloom Filters	14
6.2.1.	Computing and Saving Bloom Filters	15
6.2.2.	Forwarding based on Bloom Filters	15
6.2.3.	Hash Functions Distribution	15
7.	Implementation Status	15
8.	Security Considerations	15
9.	IANA Considerations	15
10.	Acknowledgments	15
11.	References	15
11.1.	Normative References	16
11.2.	Informative References	16
	Author's Address	18

[1.](#) Introduction

The design of Low Power and Lossy Networks (LLNs) is generally focused on saving energy, which is the most constrained resource of all. Other design constraints, such as a limited memory capacity, duty cycling of the LLN devices and low-power lossy transmissions, derive from that primary concern.

The IETF produced the "Routing Protocol for Low Power and Lossy Networks" [[RFC6550](#)] (RPL) to provide routing services within such constraints. In order to cope with lossy transmissions, RPL forms Direction-Oriented Directed Acyclic Graphs (DODAGs) that provide multiple forwarding solutions to most of the intermediate hops.

Because it is designed to adapt to fuzzy connectivity with lazy control, RPL can only provide a best effort routability, connecting most of the LLN nodes, most of the time.

RPL is a Distance-Vector protocol, which, compared to link-state protocols, limits the amount of topological knowledge that needs to be installed and maintained in each node. RPL also leverages Routing Stretch to reduce further the amount of control traffic and routing state that is required to operate the protocol. Finally, broken routes may be fixed lazily and on-demand, based on dataplane inconsistency discovery, which avoids wasting energy in the proactive repair of unused paths.

RPL enables various trade-offs between routing stretch, amount of routing state and packet size, with the introduction of different modes of operation (MOPs):

- o With Reactive Discovery of Point-to-Point (P2P) Routes (aka on-demand) [[RFC6997](#)][I-D.ietf-roll-aodv-rpl], a limited number of routes are optimized on-demand, between select pairs of devices for which a service with some particular characteristics is needed.
- o In Storing Mode of operation (aka Storing Mode), Multipoint to Point (MP2P) routes from the LLN nodes to the root and Point to Multipoint (P2MP) routes from the root to the LLN nodes are optimized, but P2P routes between LLN nodes are stretched via a common parent. In Storing Mode, RPL requires that nodes maintain routing information for the maximum number of child nodes in their sub-DODAG. If a network is composed of similar nodes, this means that each node should be able to store a number of routes that is in the order of the total number of nodes in the network.
- o In Non-Storing Mode of operation (aka Non-Storing Mode), MP2P and P2MP routes are also optimized, but downwards routes can only be computed by the root and P2MP forwarding relies on strict source-routing. This increases the size of the packets and limits the depth of the DODAG. The Non-Storing Mode also results in

additional stretch for P2P routes, whereby all packets must transit via the root following the default route all the way up, to be then source-routed down.

In order to alleviate the issues above and improve the existing multicast operations, this specification introduces the use of bitStrings in RPL LLNs. BitStrings are already used in the art as a datapath analog to one or more IPv6 [[RFC8200](#)] addresses:

- o With the Bit Index Explicit Replication (BIER), introduced in the "BIER Architecture" [[RFC8279](#)], each bit in a bitString represents a particular destination. This specification introduces a RPL-

BIER Storing Mode that applies BIER operations to RPL Storing Mode.

- o "Traffic Engineering for Bit Index Explicit Replication" [[I-D.eckert-bier-te-arch](#)] (BIER-TE) adds support for Traffic Engineering by explicit hop-by-hop forwarding and loose hop forwarding of packets along a unicast route. With BIER-TE, each bit in a bitString represents a particular intermediate link. This specification introduces a RPL-BIER Non-Storing Mode that applies BIER-TE operations to RPL Non-Storing Mode.

This specification provides new signaling in RPL to enable RPL-BIER operations. In addition to classical bitStrings, this specification proposes a new technique that derives from Bloom Filters. This technique provides elasticity to the size of the bitString, which is not constrained to a fixed number of entries, and a trade-off between the number of bits that are needed for routing and the chances to waste energy forwarding down a path where no target exist. The Bloom Filters mechanism applies to RPL-BIER in both modes of operations.

In order to carry routing information in a concise fashion in a Low-Power Wireless Personal Area Network (6LoWPAN) for Route-Over use cases, the 6LoWPAN adaptation layer framework [[RFC4944](#)] [[RFC6282](#)] was extended with the 6LoWPAN Routing Header (6LoRH) specification [[RFC8138](#)], which uses the 6LoWPAN Paging Dispatch [[RFC8025](#)]. The original specification includes the formats necessary for RPL such as the Source Route Header (SRH) and is intended to be extended for additional routing artifacts. A companion document to this, the "6LoRH for BitStrings" [[I-D.thubert-6lo-bier-dispatch](#)], specification, proposes new 6LoRH formats to enable the concise

encoding of the BIER bitStrings and of the Bloom Filters described therein.

In the current practice of LLN networks, the Non-Storing Mode is largely favored, because it does not place a restriction on how large a network formed of a particular device can become. One major benefit of introducing bitStrings is that the amount of state that is required for routing in Storing Mode is no more growing in the order of the total number of nodes in the network but linearly with the number of children attached to the RPL router. In other words, the maximum number of children that a router is willing to accept determines both the size of the Neighbor Cache for 6LoWPAN Neighbor Discovery (6LoWPAN ND) [[RFC6775](#)] [I-D.ietf-6lo-rfc6775-update] and the size of the routing table that is required in RPL-BIER Storing Mode.

[2.](#) Terminology

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

The Terminology used in this document is consistent with and incorporates that described in Terms Used in Routing for Low-Power and Lossy Networks (LLNs). [[RFC7102](#)].

Other terms in use in LLNs are found in Terminology for Constrained-Node Networks [[RFC7228](#)].

The term "byte" is used in its now customary sense as a synonym for "octet".

"RPL", "RPL Packet Information" (RPI) and "RPL Instance" are defined in the "RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks" [[RFC6550](#)] specification.

The terms Bit-Forwarding Egress Routers (BFR), BFR-id and bitString

are defined in [[RFC8279](#)]. A bitString indicates a continuous sequence of bits indexed by an offset in the sequence. The leftmost bit is bit 0 and corresponds to the value 0x80 of the leftmost byte in the BitString. With this specification, a bitString maybe used to encode one or more unsigned integer(s) as a bit position in the bitString (bit-by-bit), or a Bloom filter.

[3.](#) Applicability

BIER and other bit-indexed methods that would leverage bitStrings will generally require additional information in the packet to complement the bitString. For instance, BIER has the concept of a BFR-id and an Entropy value in the BIER header. Since those additional fields depend on the bit-indexed method, they are expected to be transported separately from the bitString. This specification concentrates on the bitString and a group identifier which enables a network to grow beyond the size of one bitString.

TBD Do we need entropy ?

[4.](#) Extensions to [RFC 6550](#)

This specification introduces two new modes of operation for RPL, the RPL-BIER Non-Storing Mode which is discussed in [Section 4.1.1](#) and the RPL-BIER Storing Mode which is discussed in [Section 4.1.2](#). A new

Control Message Options (CMO) is introduced to transport the bitStrings in [Section 4.2](#).

[4.1.](#) RPL-BIER MOPs

In RPL-BIER modes of operation, one or more RPL Target Option are replaced by a new BitString Information Option which represent the advertised target(s) by a combination of a bitString and control information.

[4.1.1.](#) RPL-BIER Non-Storing Mode

In Non-Storing RPL-BIER, a bit in classical BIER bit-by-bit bitStrings (see [Section 6.1](#)) or a set of bits in a Bloom Filter (see [Section 6.2](#)) is associated to a next-hop from the perspective of an

intermediate router. RPL Non-Storing Mode DAO messages are used to advertise the relation between a target and its parent (transit) directly to the root.

If multiple Targets Options were to be placed consecutively to factorize a Transit Information Option (TIO) in a classical RPL Non-Storing Mode DAO message, they are replaced by a single BIO with the aggregated bitString that represents all these targets.

[4.1.2.](#) RPL-BIER Storing Mode

In RPL-BIER Storing Mode, a bit in classical BIER Bit-by-bit bitStrings (see [Section 6.1](#)) or a set of bits in a Bloom Filter (see [Section 6.2](#)) is associated to a final destination. RPL Storing Mode DAO messages are used to advertise recursively the targets to the parent(s) all the way to the root.

The BitString Information Option(s) in the DAO message contain collectively an aggregated bitString that represents the advertising node and all of its descendants. Parents save the bitString per child, and use it to forward down the DODAG as discussed in [Section 6.1.3](#).

The Transit Information Option is not used. The lack of transit information is compensated by a more frequent transmission of DAO messages and a full use of the RPL data plane loop avoidance and inconsistency detection mechanisms ([section 11.2 of \[RFC6550\]](#)), in collaboration with a periodic 6LoWPAN ND (re)registration that maintains the 6LBR and the root aware of which devices are actually present in the network with the associated lifetime and sequence information.

[4.2.](#) BitString Information

[Section 6.7 of \[RFC6550\]](#) specifies optional CMOs to be placed in RPL messages such as the Destination Advertisement Object (DAO) message. The RPL Target Option and the Transit Information Option (TIO) are such optional fields; the former indicates a node to be reached and the latter specifies a parent that can be used to reach that node. Options may be factorized; one or more contiguous TIOs apply to the

one or more contiguous Target options that immediately precede the TIOs in the RPL message.

This specification introduces a new CMO, the BitString Information option (BIO). A BIO is used as an alternate to one or more Target Option(s) in Storing and Non-Storing Mode DAOs using bit-by-bit bitStrings, and its format is as follows:

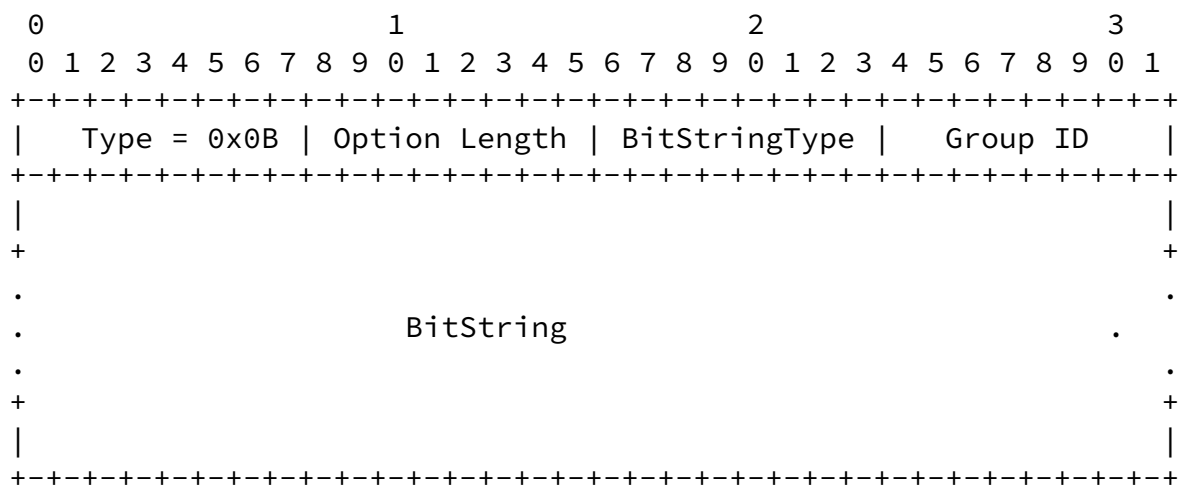


Figure 1: BitString Information option format

Option Type: 0x0B (to be confirmed by IANA)
Option Length: In bytes; variable, depending on the Type.
BitString Type: Indicates the size of the bitString field as indicated in Table 1 below.

BitString Type	BitString Size
15	8 bits
16	16 bits
17	48 bits
18	96 bits
19	160 bits

Group ID : 8-bit unsigned integer. The Group ID for the bit-by-bit bitString.

BitString: 8 to 160 bits, depending on the Type.

5. Updating the 6LoWPAN Framework

5.1. Extensions to [RFC 6775](#)

It is noted that RPL does not provide a Duplicate Address Detection (DAD) and relies on 6LoWPAN ND to ensure that addresses are unique within the network. For that purpose, a 6LoWPAN Border Router (6LBR) maintains the list of addresses that are currently in use in the network that it serves. In the case of a RPL LLN, the 6LBR is typically collocated with the RPL root, and serves the RPL DODAG. With 6LoWPAN ND[RFC6775] [[I-D.ietf-6lo-rfc6775-update](#)], a Duplicate Address Request (DAR) / Duplicate Address Confirmation (DAC) exchange is used to perform the DAD operation. Scalability is achieved by federating multiple DODAGs with IPv6 Backbone Routers (6BBRs) [[I-D.ietf-6lo-backbone-router](#)].

In that context, it makes sense to also leverage the 6LBR to ensure that a tuple (groupID, bitString) is assigned unequivocally to an IPv6 address for the bit-by-bit operation. This specification extends the role of a 6LBR to 1) assign the tuple to the IPv6 address and 2) resolve an IPv6 address into a tuple. To achieve this, [RFC 6775](#) is updated as follows:

- o A BIER Address Resolution (BAR) / BIER Address Confirmation (BAC) exchange is introduced for the purpose of the bitString lookup operation (see [Section 6.1.1](#)).
- o A new Bit Position Option (BPO) is introduced to carry the corresponding bit position bitString in 6LoWPAN ND exchanges. The BPO is transported in BAC, NA and DAC messages in response to BAR, NS and DAR messages, respectively (see [Section 5.3.1](#)).

Type:	38 (to be confirmed by IANA)
Length:	8-bit unsigned integer. The length of the option (including the type and length fields) in units of 8 bytes. The Length MUST be set to 1.
Group ID:	8-bit unsigned integer. The GroupID for the bit-by-bit bitString.

Bit Position: 8-bit unsigned integer. The offset in the the bit-by-bit bitString.

[5.3.2.](#) Address Mapping Message

For the multihop lookup exchanges between a 6LR that needs to perform a Header Compression including the bitString for the destination, and its 6LBR, which knows if the mapping exists and what it is, this specification introduces two new ICMPv6 message called the BIER Address Resolution (BAR) and the BIER Address Confirmation (BAC). We avoid reusing the NS and NA messages for this purpose, since these messages are not subject to the hop limit=255 check as they are forwarded by intermediate 6LRs.

The BAR and BAC use the same ICMPv6 type value which this specification, allocates for a generic Address Mapping service, but use different Codes. This is done to save addressable space in the ICMPv6 type values which is getting crowded, and because it is expected that in the future, other mapping techniques may be needed as well.

The Status field and Information Lifetime are not meaningful in the BAR message. When and only when the BAC message carries a status of 0, indicating success, the Information Lifetime must contain valid information and the message must carry a Bit Position Option ([Section 5.3.1](#)).

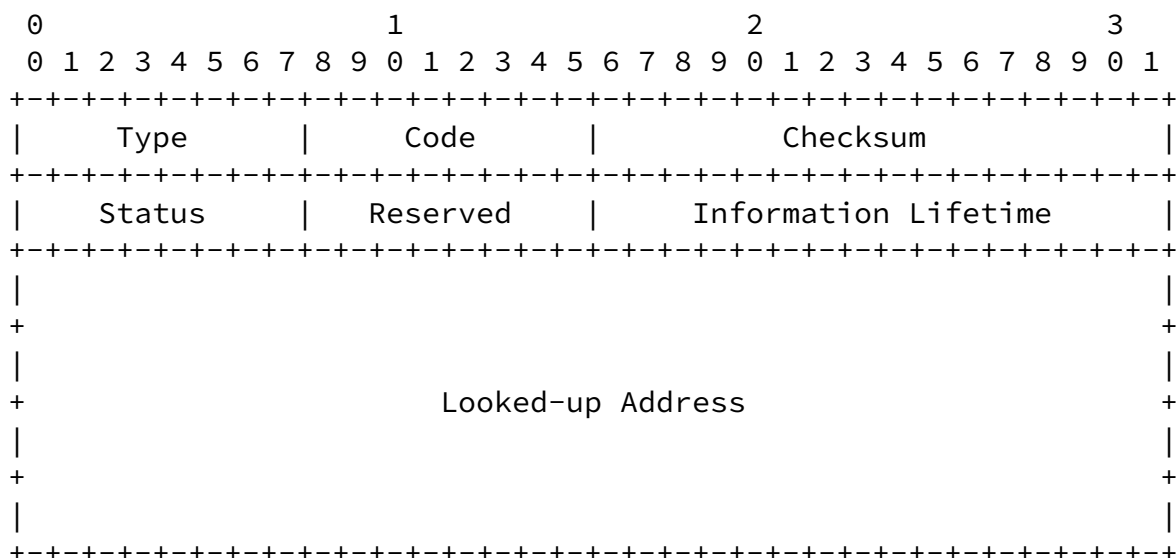


Figure 3: Address Mapping Message format

Message Fields

Type: 160 (to be confirmed by IANA)
Code: 8-bit unsigned integer. 1 for BAR and 2 for BAC.
Other values are reserved.
Checksum:: The ICMP checksum. See [[RFC4443](#)].

Thubert

Expires January 25, 2019

[Page 10]

Internet-Draft

RPL-BIER

July 2018

Status: 8-bit unsigned integer. Indicates the status of the lookup in the BAC. See Table 2 below.

Status	Description
0	Success
1	Looked-up Address not Found
2..255	Allocated using Standards Action [RFC8126]

Information Lifetime: 16-bit unsigned integer. The length of time in units of 60 seconds (relative to the time the packet is received) that this mapping information is valid. A value of all zero bits (0x0) assumes a default value of 10,000 (~one week).

Looked-up Address: 128-bit field. Carries the IPv6 address that is being mapped.

6. BitString formats

This specification introduces two BitString formats, the bit-by-bit and the Bloom Filter.

6.1. Bit-by-bit BitStrings

In the bit-by-bit case, each bit is mapped in an unequivocal fashion with a single addressable resource in the network. In RPL-BIER Storing Mode, this is an IPv6 address as advertised in RPL Storing Mode DAO messages, whereas in RPL-BIER Non-Storing Mode, this is a

parent-child relationship as advertised in RPL Non-Storing Mode DAO messages.

if every node in a large network is given one or more bits in a bit-by-bit bitString, then the bitString may grow very large and lead to undesirably large overhead in the data plane. BIER allows to divide a potentially the very large abstract bitString into segments, aka groups, indexed by a groupID.

In the protocol elements that use a bitString, only the relevant group(s) are transported, and the advertised bitString is in fact the segment that corresponds to the groupID. It results that a bit position in the large abstract bitString is advertised either as the tuple (groupID, segment) or the tuple (groupID, bit position within segment).

Thubert

Expires January 25, 2019

[Page 11]

Internet-Draft

RPL-BIER

July 2018

For simplicity, when dealing with protocol elements, the specification uses the term bitString to refer to the tuple (groupID, segment) and a bitwise operation between bitStrings is really a bitwise operation between segments of a same groupID.

TBD: do we allow multiple (groupID, bitString) tuples in one packet?

[6.1.1](#). Allocating a Bit Position in a Bit-by-bit BitString

Several methods may be used to associate a bit in a biString to an IPv6 address. In order to guarantee interoperability, this specification RECOMMENDS that all implementations provide at least the method described therein.

With 6LoWPAN ND, a 6LoWPAN Node (6LN) registers with address(es) to one or more 6LoWPAN Router [[RFC6775](#)] to perform Duplicate Address Detection (DAD). As part of the DAD process, the 6LN validates that a Global Unicast Address (GUA) or a Unique Local Address (ULA) is effectively unique using a unicast DAR/DAC exchange with the 6LBR (this procedure is updated in [[I-D.ietf-6lo-rfc6775-update](#)]).

In a network that supports this specification, the 6LBR maintains a configurable number of groups (up to 32, indexed by groupID), and for each group, it maintains a pool of free bit positions. Upon a new registration, the 6LBR selects a groupID and a free bit position and

associates it to the IPv6 address.

The bitString Size in any given group should be configurable. The policy for selecting the groupID for a new registration is left to implementation. It is noted that in large networks that require multiple groups, associating groups with immediate children of the root may be an option to limit the number of groups that the RPL routers must be aware of.

If the 6LBR accepts the registration, then it returns a DAC message with a status of 0 indicating success, adding a 6LoWPAN ND Bit Position Option ([Section 5.3.1](#)) to the DAC message to indicate the groupID and bit.

The 6LR maintains a binding cache entry (BCE) for the 6LN based on successful DAC messages. With this specification, the 6LR also stores the matching between the address and the bitString and uses it for searching its children when forwarding packets in Non-Storing Mode (see [Section 6.1.3](#)).

If the 6LN child does not support the BIER encoding (e.g. [[I-D.thubert-6lo-bier-dispatch](#)]), then the packet is converted in a format that the child supports (e.g. [[RFC8138](#)]).

[6.1.2.](#) Aggregation of Bit-by-bit BitStrings

BitStrings are aggregated by a 'OR' operation so that all the bits that are set in either bitString is set in the resulting bitString. In the concise form of a tuple (groupID, bitString), the aggregation is done on a group-by-group basis, between segments of a same group.

In RPL-BIER Storing Mode, the bit-by-bit BitStrings are passed from child to parent using DAO messages, in a fashion similar to RPL Storing Mode [[RFC6550](#)]. The BitString Information option (Figure 1) is used in replacement of the Target option. A DAO message contains one BIO per group, and the parent that receives the messages associates the BIO information to the advertising child. In order to build a DAO message, the parent regenerates its own BIO, one per group, by aggregating the bitStrings from all of its children with its own, and places the resulting BIOs in the DAO message.

[6.1.3.](#) Forwarding Based on Bit-by-bit BitStrings

Forwarding is based on matching a bitString in a packet with those of children. For unicast packets, only one matching child gets the packet. For multicast packets, all matching children get a copy. Matches are found by scanning all children and performing bitwise operations as follows.

In order to search for a match, a reference bitString is initialized with the destination bitString in the packet. A match is found with a child if the bitwise 'AND' between the reference bitString and the bitString stored for that child does not result in a NULL bitString of all zeroes.

In Non-Storing Mode, a packet is copied to all matching children, which are found by trying all children.

In Non-Storing Mode, if a child is selected for forwarding, then an 'XOR' operation is performed between the reference bitString and the bitString resulting from the 'AND' operation. If the 'XOR' operation does not result in a NULL bitString, denoting that more children should get the packet, then the result of the 'XOR' operation becomes the new reference bitString and the search continues. The 'XOR' operation allows to stop the search loop as soon as all matches are found; it also avoids forwarding twice to a same destination along different downwards path in the DODAG.

[6.1.4.](#) Reliable Multicast based on Bit-by-bit BitStrings

Multicast from the root to a collection of target 6LNs can be made reliable with the following operation:

A multicast packet is identified by a unique packetID which is also found in the acknowledgments. The root signals the set of targets with a destination bitString that has the bits set for each of them, and the message is forwarded as described on [Section 6.1.3](#).

Listeners acknowledge with an acknowledgment packet that contains the

same information, the packetID and the bitString representing the listener. The bitStrings in acknowledgment packets are aggregated recursively on the way back as indicated in [Section 6.1.2](#).

The root aggregates the bitStrings from its children into one acknowledgment bitString. It then checks that the acknowledgment bitString is correct, by an 'AND' operation with the destination bitString that should result in the acknowledgment bitString. If this is not the case, bits that are set in the acknowledgment bitString and not in the destination bitString are in the acknowledgment bitString.

The root generates the bitString of the devices that did not acknowledge the multicast message by a bitwise 'XOR' operation between the destination bitString and the acknowledgment bitString, and may use it to selectively retry the multicast.

[6.2.](#) Bloom Filters

A Bloom Filter can be seen as an additional compression technique for the bitString representation. A Bloom Filter may generate false positives, which, in the case of BIER, result in undue forwarding of a packet down a path where no listener exists.

As an example, the Constrained-Cast [[I-D.bergmann-bier-ccast](#)] specification employs Bloom Filters as a compact representation of a match or non-match for elements in a set that may be larger than the number of bits in the BitString.

In the case of a Bloom Filter, a number of Hash functions must be run to obtain a multi-bit signature of an encoded element. This specification uses the 5-bits Control field to signal an Identifier of the set of Hash functions being used to generate a certain bitString, so as to enable the migration from a set of Hash functions to the next.

[6.2.1.](#) Computing and Saving Bloom Filters

[6.2.2.](#) Forwarding based on Bloom Filters

[6.2.3.](#) Hash Functions Distribution

[7.](#) Implementation Status

TBD

[8.](#) Security Considerations

TBD

[9.](#) IANA Considerations

This document extends the IANA registry created by [RFC 6550](#) for RPL Control Codes as follows:

+-----+-----+-----+		
Code	Description	Reference
+-----+-----+-----+		
0x0B	bitString	This document
+-----+-----+-----+		

RPL Control Codes

This document is updating the registry created by [RFC 6550](#) for the RPL 3-bit Mode of Operation (MOP) as follows:

+-----+-----+-----+		
MOP value	Description	Reference
+-----+-----+-----+		
6	RPL-BIER Non-Storing Mode of operation	This document
7	RPL-BIER Storing Mode of operation	This document
+-----+-----+-----+		

DIO Mode of operation

[10.](#) Acknowledgments

[11.](#) References

11.1. Normative References

- [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>>.
- [RFC4443] Conta, A., Deering, S., and M. Gupta, Ed., "Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification", STD 89, [RFC 4443](#), DOI 10.17487/RFC4443, March 2006, <<https://www.rfc-editor.org/info/rfc4443>>.
- [RFC6550] Winter, T., Ed., Thubert, P., Ed., Brandt, A., Hui, J., Kelsey, R., Levis, P., Pister, K., Struik, R., Vasseur, JP., and R. Alexander, "RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks", [RFC 6550](#), DOI 10.17487/RFC6550, March 2012, <<https://www.rfc-editor.org/info/rfc6550>>.
- [RFC6775] Shelby, Z., Ed., Chakrabarti, S., Nordmark, E., and C. Bormann, "Neighbor Discovery Optimization for IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs)", [RFC 6775](#), DOI 10.17487/RFC6775, November 2012, <<https://www.rfc-editor.org/info/rfc6775>>.
- [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/info/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/info/rfc8279>>.

11.2. Informative References

- [I-D.bergmann-bier-ccast] Bergmann, O., Bormann, C., Gerdes, S., and H. Chen, "Constrained-Cast: Source-Routed Multicast for RPL", [draft-bergmann-bier-ccast-02](#) (work in progress), October 2016.

[I-D.eckert-bier-te-arch]

Eckert, T., Cauchie, G., Braun, W., and M. Menth, "Traffic Engineering for Bit Index Explicit Replication BIER-TE", [draft-eckert-bier-te-arch-06](#) (work in progress), November 2017.

[I-D.ietf-6lo-backbone-router]

Thubert, P., "IPv6 Backbone Router", [draft-ietf-6lo-backbone-router-06](#) (work in progress), February 2018.

[I-D.ietf-6lo-rfc6775-update]

Thubert, P., Nordmark, E., Chakrabarti, S., and C. Perkins, "Registration Extensions for 6LoWPAN Neighbor Discovery", [draft-ietf-6lo-rfc6775-update-21](#) (work in progress), June 2018.

[I-D.ietf-roll-aodv-rpl]

Anamalamudi, S., Zhang, M., Sangi, A., Perkins, C., Anand, S., and B. Liu, "Asymmetric AODV-P2P-RPL in Low-Power and Lossy Networks (LLNs)", [draft-ietf-roll-aodv-rpl-04](#) (work in progress), July 2018.

[I-D.thubert-6lo-bier-dispatch]

Thubert, P., Brodard, Z., Jiang, H., and G. Texier, "A 6loRH for BitStrings", [draft-thubert-6lo-bier-dispatch-04](#) (work in progress), January 2018.

[RFC4944] Montenegro, G., Kushalnagar, N., Hui, J., and D. Culler, "Transmission of IPv6 Packets over IEEE 802.15.4 Networks", [RFC 4944](#), DOI 10.17487/RFC4944, September 2007, <<https://www.rfc-editor.org/info/rfc4944>>.

[RFC6282] Hui, J., Ed. and P. Thubert, "Compression Format for IPv6 Datagrams over IEEE 802.15.4-Based Networks", [RFC 6282](#), DOI 10.17487/RFC6282, September 2011, <<https://www.rfc-editor.org/info/rfc6282>>.

[RFC6997] Goyal, M., Ed., Baccelli, E., Philipp, M., Brandt, A., and J. Martocci, "Reactive Discovery of Point-to-Point Routes in Low-Power and Lossy Networks", [RFC 6997](#),

DOI 10.17487/RFC6997, August 2013,
<<https://www.rfc-editor.org/info/rfc6997>>.

- [RFC7102] Vasseur, JP., "Terms Used in Routing for Low-Power and Lossy Networks", [RFC 7102](#), DOI 10.17487/RFC7102, January 2014, <<https://www.rfc-editor.org/info/rfc7102>>.

Thubert

Expires January 25, 2019

[Page 17]

Internet-Draft

RPL-BIER

July 2018

- [RFC7228] Bormann, C., Ersue, M., and A. Keranen, "Terminology for Constrained-Node Networks", [RFC 7228](#), DOI 10.17487/RFC7228, May 2014, <<https://www.rfc-editor.org/info/rfc7228>>.
- [RFC8025] Thubert, P., Ed. and R. Cragie, "IPv6 over Low-Power Wireless Personal Area Network (6LoWPAN) Paging Dispatch", [RFC 8025](#), DOI 10.17487/RFC8025, November 2016, <<https://www.rfc-editor.org/info/rfc8025>>.
- [RFC8126] Cotton, M., Leiba, B., and T. Narten, "Guidelines for Writing an IANA Considerations Section in RFCs", [BCP 26](#), [RFC 8126](#), DOI 10.17487/RFC8126, June 2017, <<https://www.rfc-editor.org/info/rfc8126>>.
- [RFC8138] Thubert, P., Ed., Bormann, C., Toutain, L., and R. Cragie, "IPv6 over Low-Power Wireless Personal Area Network (6LoWPAN) Routing Header", [RFC 8138](#), DOI 10.17487/RFC8138, April 2017, <<https://www.rfc-editor.org/info/rfc8138>>.

Author's Address

Pascal Thubert (editor)
Cisco Systems, Inc
Building D
45 Allée des Ormes - BP1200
MOUGINS - Sophia Antipolis 06254
FRANCE

Phone: +33 497 23 26 34
Email: pthubert@cisco.com

Thubert

Expires January 25, 2019

[Page 18]