Internet Engineering Task Force
Internet-Draft
Intended status: Experimental
Expires: July 25, 2013

M. Goyal, Ed.
University of Wisconsin
Milwaukee
E. Baccelli
INRIA
A. Brandt
Sigma Designs
J. Martocci
Johnson Controls
January 21, 2013

A Mechanism to Measure the Routing Metrics along a Point-to-point Route in a Low Power and Lossy Network

draft-ietf-roll-p2p-measurement-08

Abstract

This document specifies a mechanism that enables an RPL router to measure the aggregated values of given routing metrics along an existing route towards another RPL router in a low power and lossy network, thereby allowing the router to decide if it wants to initiate the discovery of a better route.

Status of this Memo

Copyright Notice

Table of Contents

1. Introduction

Point to point (P2P) communication between arbitrary routers in a Low power and Lossy Network (LLN) is a key requirement for many applications [RFC5826][RFC5867]. The IPv6 Routing Protocol for LLNs (RPL) [RFC6550] constrains the LLN topology to a Directed Acyclic Graph (DAG) built to optimize the routing costs to reach the DAG's root. The P2P routing functionality, available under RPL, has the following key limitations:

- o The P2P routes are restricted to use the DAG links only. Such P2P routes may potentially be suboptimal and may lead to traffic congestion near the DAG root.

- o RPL is a proactive routing protocol and hence requires all P2P routes to be established ahead of the time they are used. Many LLN applications require the ability to establish P2P routes "on demand".

To ameliorate situations, where the core RPL's P2P routing functionality does not meet the application requirements, [I-D.ietf-roll-p2p-rpl] describes P2P-RPL, an extension to core RPL. P2P-RPL provides a reactive mechanism to discover P2P routes that meet the specified routing constraints [RFC6551]. In some cases, the application requirements or the LLN's topological features allow a router to infer these routing constraints implicitly. For example, the application may require the end-to-end loss rate and/or latency along the route to be below certain thresholds or the LLN topology may be such that a router can safely assume its destination to be less than a certain number of hops away from itself.

When the existing routes are deemed unsatisfactory but the router does not implicitly know the routing constraints to be used in P2P-RPL route discovery, it may be necessary for the router to measure the aggregated values of the routing metrics along the existing route. This knowledge will allow the router to frame reasonable

routing constraints to discover a better route using P2P-RPL. For example, if the router determines the aggregate ETX [RFC6551] along an existing route to be "x", it can use "ETX < x*y", where y is a certain fraction, as the routing constraint for use in P2P-RPL route discovery. Note that it is important that the routing constraints are not overly strict; otherwise the P2P-RPL route discovery may fail even though a route, much better than the one currently being used, exists.

This document specifies a mechanism that enables an RPL router to measure the aggregated values of the routing metrics along an existing route to another RPL router in an LLN, thereby allowing the

router to decide if it wants to discover a better route using P2P-RPL and determine the routing constraints to be used for this purpose. Thus, the utility of this mechanism is dependent on the existence of P2P-RPL, which is targeting publication as an Experimental RFC. It makes sense, therefore, for this document also to target publication as an Experimental RFC. As more operational experience is gained using P2P-RPL, it is hoped that the mechanism described in this document will also be used, and feedback will be provided to the ROLL working group on the utility and benefits of this document.

1.1. 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].

This document uses terminology from [RFC6550] and [I-D.ietf-roll-p2p-rpl]. Additionally, this document defines the following terms.

Start Point: The Start Point refers to the RPL router that initiates the measurement process defined in this document and is the start point of the P2P route being measured.

End Point: The End Point refers to the RPL router at the end point of the P2P route being measured.

Intermediate Point: An RPL router, other than the Start Point and the

End Point, on the P2P route being measured.

The following terms, already defined in [I-D.ietf-roll-p2p-rpl], have been redefined in this document in the following manner.

Forward direction: The direction from the Start Point to the End Point.

Backward direction: The direction from the End Point to the Start Point.

2. Overview

The mechanism described in this document can be used by a Start Point in an LLN to measure the aggregated values of selected routing metrics along a P2P route to an End Point within the LLN. The route is measured in the Forward direction. Such a route could be a Source Route [I-D.ietf-roll-p2p-rpl] or a Hop-by-hop Route

[I-D.ietf-roll-p2p-rpl] established using RPL [RFC6550] or P2P-RPL [I-D.ietf-roll-p2p-rpl]. Such a route could also be a "mixed" route with the initial part consisting of hop-by-hop ascent to the root of a non-storing DAG [RFC6550] and the final part consisting of a source-routed descent to the End Point. The Start Point decides what metrics to measure and sends a Measurement Request message, carrying the desired routing metric objects, along the route. If a Source Route is being measured, the Measurement Request carries the route inside an Address vector. If a Hop-by-hop Route is being measured, the Measurement Request identifies the route by its RPLInstanceID [RFC6550] (and, in case the RPLInstanceID is a local value, the Start Point's IPv6 address associated with the route). On receiving a Measurement Request, an Intermediate Point updates the routing metric values inside the message and forwards it to the next hop on the route. Thus, the Measurement Request accumulates the values of the routing metrics for the complete route as it travels towards the End Point. Upon receiving the Measurement Request, the End Point unicasts a Measurement Reply message, carrying the accumulated values of the routing metrics, back to the Start Point. Optionally, the Start Point may allow an Intermediate Point to generate the Measurement Reply if the Intermediate Point already knows the relevant routing metric values along rest of the route.

The Measurement Request may include an Address vector that serves one of the following functions:

- o To accumulate a Source Route for End Point's use: If a Hop-by-hop Route with a local RPLInstanceID is being measured, the Start Point may require each Intermediate Point to add its IPv6 address to an Address vector inside the Measurement Request. The Source Route, thus accumulated, can be used by the End Point to reach the Start Point. In particular, the End Point may use the accumulated Source Route to send the Measurement Reply back to the Start Point. In this case, the Start Point includes a suitably-sized Address vector in the Measurement Request. The size of the Address vector puts a hard limit on the length of the accumulated route. An Intermediate Point is not allowed to modify the size of the Address vector and must discard a received Measurement Request if the Address vector is not large enough to contain the complete route.

- o To carry the Source Route being measured: The Start Point may insert an Address vector inside the Measurement Request to carry the Source Route being measured. Also, the root of a global non-storing DAG may insert an Address vector, carrying a Source Route from itself to the End Point, inside a Measurement Request message if this message had been traveling along this DAG so far. In both cases, an Intermediate Point is not allowed to modify an existing

Address vector before forwarding the Measurement Request further. In other words, an Intermediate Point is not allowed to modify the Source Route along which the Measurement Request is currently traveling.

3. The Measurement Object (MO)

This document defines two new RPL Control Message types, the Measurement Object (MO), with code TBD1, and the Secure MO, with code TBD2. An MO serves as both Measurement Request and Measurement Reply.

3.1. Format of the base MO

```
 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
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 | RPLInstanceID | Compr |T|H|A|R|B|I| SequenceNo|   Num   | Index |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |                                                               |
 |                      Start Point Address                      |
 |                                                               |
 |                                                               |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |                                                               |
 |                       End Point Address                       |
 |                                                               |
 |                                                               |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |                                                               |
 .                      Address[0..Num-1]                          .
 .                                                               .
 |                                                               |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |                                                               |
 .                  Metric Container Option(s)                   .
 .                                                               .
 |                                                               |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 1: Format of the base Measurement Object (MO)

The format of a base MO is shown in Figure 1. A base MO consists of the following fields:

- RPLInstanceID: This field specifies the RPLInstanceID of the Hop-by-hop Route along which the Measurement Request travels (or traveled initially until it switched over to a Source Route).

- Compr: In many LLN deployments, IPv6 addresses share a well known, common prefix. In such cases, the common prefix can be elided when specifying IPv6 addresses in the Start Point/End Point Address fields and the Address vector. The "Compr" field, a 4-bit

unsigned integer, is set by the Start Point to specify the number of prefix octets that are elided from the IPv6 addresses in Start Point/End Point Address fields and the Address vector. The Start Point will set the Compr value to zero if full IPv6 addresses are to be carried in the Start Point Address/End Point Address fields and the Address vector.

- Type (T): This flag is set to one if the MO represents a Measurement Request. The flag is set to zero if the MO is a Measurement Reply.

- Hop-by-hop (H): The Start Point MUST set this flag to one if (at least the initial part of) the route being measured is hop-by-hop. In that case, the Hop-by-hop Route is identified by the RPLInstanceID, the End Point Address and, if the RPLInstanceID is a local value, the Start Point Address (required to be same as the DODAGID of the route being measured) fields inside the Measurement Request. The Start Point MUST set this flag to zero if the route being measured is a Source Route specified in the Address vector. An Intermediate Point MUST set the H flag in an outgoing Measurement Request to the same value that it had in the corresponding incoming Measurement Request unless it is the root of the non-storing global DAG, identified by the RPLInstanceID, along which the Measurement Request had been traveling so far and the Intermediate Point intends to insert a Source Route inside the Address vector to direct it towards the End Point. In that case, the Intermediate Point MUST set the H flag to zero.

- Accumulate Route (A): A value 1 in this flag indicates that the Measurement Request is accumulating a Source Route for use by the End Point to send the Measurement Reply back to the Start Point. Route accumulation is allowed (i.e., this flag MAY be set to one) inside a Measurement Request only if it travels along a Hop-by-hop Route represented by a local RPLInstanceID (i.e., H = 1, RPLInstanceID has a local value). In this case, an Intermediate Point adds its unicast IPv6 address (after eliding Compr number of prefix octets) to the Address vector in the manner specified in Section 5.3. In other cases, this flag MUST be set to zero on transmission and ignored on reception. Route accumulation is not allowed when the Measurement Request travels along a Hop-by-hop

Route with a global RPLInstanceID, i.e., along a global DAG,

because:

* The DAG's root may need the Address vector to insert a Source Route to the End Point; and

* The End Point can presumably reach the Start Point along this global DAG (identified by the RPLInstanceID field).

o Reverse (R): A value 1 in this flag inside a Measurement Request indicates that the Address vector contains a complete Source Route from the Start Point to the End Point, which can be used, after reversal, by the End Point to send the Measurement Reply back to the Start Point. This flag MAY be set to one inside a Measurement Request only if a Source Route, from the Start Point to the End Point, is being measured. Otherwise, this flag MUST be set to zero on transmission and ignored on reception.

o Back Request (B): A value 1 in this flag serves as a request to the End Point to send a Measurement Request towards the Start Point. On receiving a Measurement Request with the B flag set to one, the End Point SHOULD generate a Measurement Request to measure the cost of its current (or the most preferred) route to the Start Point. Receipt of this Measurement Request would allow the Start Point to know the cost of the back route from the End Point to itself and thus determine the round-trip cost of reaching the End Point.

o Intermediate Reply (I): A value 1 in this flag serves as a permission to an Intermediate Point to generate a Measurement Reply if it knows the aggregated values of the routing metrics being measured for the rest of the route. Setting this flag to one may be useful in scenarios where the Hop Count [RFC6551] is the routing metric of interest and an Intermediate Point (e.g. the root of a non-storing global DAG or a common ancestor of the Start Point and the End Point in a storing global DAG) may know the Hop Count of the remainder of the route to the End Point. This flag MAY be set to one only if a Hop-by-hop Route with a global RPLInstanceID is being measured (i.e., H = 1, RPLInstanceID has a global value). Otherwise, this flag MUST be set to zero on transmission and ignored on reception.

o SequenceNo: A 6-bit sequence number, assigned by the Start Point, that allows the Start Point to uniquely identify a Measurement Request and the corresponding Measurement Reply.

o Num: This field indicates the number of elements, each (16 - Compr) octets in size, inside the Address vector. If the value of

this field is zero, the Address vector is not present in the MO.

- Index: If the Measurement Request is traveling along a Source Route contained in the Address vector (i.e., H = 0), this field indicates the index in the Address vector of the next hop on the route. If the Measurement Request is traveling along a Hop-by-hop Route with a local RPLInstanceID and the Route Accumulation is on (i.e., H = 1, RPLInstanceID has a local value, A = 1), this field indicates the index in the Address vector where an Intermediate Point receiving the Measurement Request must store its IPv6 address. Otherwise, this field MUST be set to zero on transmission and ignored on reception.

- Start Point Address: A unicast IPv6 address of the Start Point after eliding Compr number of prefix octets. If the Measurement Request is traveling along a Hop-by-hop Route and the RPLInstanceID field indicates a local value, the Start Point Address field MUST specify the DODAGID value that, along with the RPLInstanceID and the End Point Address, uniquely identifies the Hop-by-hop Route being measured.

- End Point Address: A unicast IPv6 address of the End Point after eliding Compr number of prefix octets.

- Address[0..Num-1]: A vector of unicast IPv6 addresses (with Compr number of prefix octets elided) representing a Source Route:

  * Each element in the vector has size (16 - Compr) octets.

  * The total number of elements inside the Address vector is given by the Num field.

  * The Start Point and End Point addresses MUST NOT be included in the Address vector.

  * The Address vector MUST NOT contain any multicast addresses.

  * If the Start Point wants to measure a Hop-by-hop Route with a local RPLInstanceID and accumulate a Source Route for the End Point's use (i.e., the Measurement Request has the H flag set to 1, RPLInstanceID set to a local value and the A flag set to 1), it MUST include a suitably-sized Address vector in the Measurement Request. As the Measurement Request travels over the route being measured, the Address vector accumulates a Source Route that can be used by the End Point, after reversal, to reach (and, in particular, to send the Measurement Reply

back to) the Start Point. The route MUST be accumulated in the Forward direction but the IPv6 addresses in the accumulated

route MUST be reachable in the Backward direction. An Intermediate Point adding its address to the Address vector MUST NOT modify the size of the Address vector.

* If the Start Point wants to measure a Source Route, it MUST include an Address vector, containing the route being measured, inside the Measurement Request. Similarly, if the Measurement Request had been traveling along a global non-storing DAG so far, the root of this DAG may insert an Address vector, containing a Source Route from itself to the End Point, inside the Measurement Request. In both cases, the Source Route inside the Address vector MUST consist of IPv6 addresses reachable in the Forward direction. Further, in both cases, an Intermediate Point MUST NOT modify the contents of the existing Address vector before forwarding the Measurement Request further. In other words, an Intermediate Point MUST NOT modify the Source Route along which the Measurement Request is currently traveling. The Start Point MAY set the R flag in the Measurement Request to one if the Source Route inside the Address vector can be used by the End Point, after reversal, to reach (and, in particular, to send the Measurement Reply back to) the Start Point. In other words, the Start Point MAY set the R flag to one only if all the IPv6 addresses in the Address vector are reachable in the Backward direction.

o Metric Container Options: A Measurement Request MUST contain one or more Metric Container options [RFC6550] to accumulate the values of the selected routing metrics in the manner described in [RFC6551] for the route being measured.

Section 4 describes how does a Start Point set various fields inside a Measurement Request in different cases. Section 5 describes how does an Intermediate Point process a received Measurement Request before forwarding it further. Section 6 describes how does the End Point process a received Measurement Request and generate a Measurement Reply. Finally, Section 7 describes how does the Start Point process a received Measurement Reply. In the following discussion, any reference to discarding a received Measurement Request/Reply with "no further processing" does not preclude updating

the appropriate error counters or any similar actions.

3.2. Secure MO

A Secure MO follows the format in Figure 7 of [RFC6550], where the base format is the base MO shown in Figure 1.

4. Originating a Measurement Request

A Start Point sets various fields inside the Measurement Request it generates in the manner described below. The Start Point MUST also include the routing metric objects [RFC6551] of interest inside one or more Metric Container options inside the Measurement Request. The Start Point then determines the next hop on the route being measured. If a Hop-by-hop route is being measured (i.e., H = 1), the next hop is determined using the RPLInstanceID, the End Point Address and, if RPLInstanceID is a local value, the Start Point Address fields in the Measurement Request. If a Source Route is being measured (i.e., H = 0), the Address[0] element inside the Measurement Request contains the next hop address. The Start Point MUST discard the Measurement Request if:

- o the next hop address is not a unicast address; or
- o the next hop is not on-link; or
- o the next hop is not in the same RPL routing domain as the Start Point.

Otherwise, depending on the routing metrics, the Start Point must initiate the routing metric objects inside the Metric Container options by including the routing metric values for the first hop on the route being measured. Finally, the Start Point MUST unicast the Measurement Request to the next hop on the route being measured.

4.1. When Measuring A Hop-by-hop Route with a Global RPLInstanceID

If a Hop-by-hop Route with a global RPLInstanceID is being measured (i.e., H = 1, RPLInstanceID has a global value), the MO MUST NOT

contain an Address vector and various MO fields MUST be set in the following manner:

o RPLInstanceID: MUST be set to the RPLInstanceID of the route being measured.

o Compr: MUST be set to specify the number of prefix octets that are elided from the IPv6 addresses in Start Point/End Point Address fields.

o Type (T): MUST be set to one since the MO represents a Measurement Request.

o Hop-by-hop (H): MUST be set to one.

o Accumulate Route (A): This flag MUST be set to zero.

o Reverse (R): This flag MUST be set to zero.

o Back Request (B): This flag MAY be set to one to request the End Point to send a Measurement Request to the Start Point.

o Intermediate Reply (I): This flag MAY be set to one if the Start Point expects an Intermediate Point to know the values of the routing metrics being measured for the remainder of the route.

o SequenceNo: Assigned by the Start Point so that it can uniquely identify the Measurement Request and the corresponding Measurement Reply.

o Num: This field MUST be set to zero.

o Index: This field MUST be set to zero.

o Start Point Address: MUST be set to a unicast IPv6 address of the Start Point after eliding Compr number of prefix octets.

o End Point Address: MUST be set to a unicast IPv6 address of the End Point after eliding Compr number of prefix octets.

4.2. When Measuring A Hop-by-hop Route with a Local RPLInstanceID With Route Accumulation Off

If a Hop-by-hop Route with a local RPLInstanceID is being measured and the Start Point does not want the MO to accumulate a Source Route for the End Point's use, the MO MUST NOT contain the Address vector and various MO fields MUST be set in the following manner:

- RPLInstanceID: MUST be set to the RPLInstanceID of the route being measured.

- Compr: MUST be set to specify the number of prefix octets that are elided from the IPv6 addresses in Start Point/End Point Address fields.

- Type (T): MUST be set to one since the MO represents a Measurement Request.

- Hop-by-hop (H): MUST be set to one.

- Accumulate Route (A): This flag MUST be set to zero.

- Reverse (R): This flag MUST be set to zero.

- Back Request (B): This flag MAY be set to one to request the End Point to send a Measurement Request to the Start Point.

- Intermediate Reply (I): This flag MUST be set to zero.

- SequenceNo: Assigned by the Start Point so that it can uniquely identify the Measurement Request and the corresponding Measurement Reply.

- Num: This field MUST be set to zero.

- Index: This field MUST be set to zero.

- Start Point Address: This field MUST contain the DODAGID value (after eliding Compr number of prefix octets) associated with the route being measured.

o End Point Address: MUST be set to a unicast IPv6 address of the End Point after eliding Compr number of prefix octets.

4.3. When Measuring A Hop-by-hop Route with a Local RPLInstanceID With Route Accumulation On

If a Hop-by-hop Route with a local RPLInstanceID is being measured and the Start Point desires the MO to accumulate a Source Route for the End Point to send the Measurement Reply message back, the MO MUST contain a suitably-sized Address vector and various MO fields MUST be set in the following manner:

- RPLInstanceID: MUST be set to the RPLInstanceID of the route being measured.

- Compr: MUST be set to specify the number of prefix octets that are elided from the IPv6 addresses in Start Point/End Point Address fields and the Address vector.

- Type (T): MUST be set to one since the MO represents a Measurement Request.

- Hop-by-hop (H): MUST be set to one.

- Accumulate Route (A): This flag MUST be set to one.

- Reverse (R): This flag MUST be set to zero.

- Back Request (B): This flag MAY be set to one to request the End Point to send a Measurement Request to the Start Point.

- Intermediate Reply (I): This flag MUST be set to zero.

- SequenceNo: Assigned by the Start Point so that it can uniquely identify the Measurement Request and the corresponding Measurement Reply.

- Num: This field MUST specify the number of address elements, each (16 - Compr) octets in size, that can fit inside the Address

vector.

o Index: This field MUST be set to zero to indicate the position in the Address vector where the next hop must store its IPv6 address.

o Start Point Address: This field MUST contain the DODAGID value (after eliding Compr number of prefix octets) associated with the route being measured.

o End Point Address: MUST be set to a unicast IPv6 address of the End Point after eliding Compr number of prefix octets.

o Address vector: The Address vector must be large enough to accomodate a complete Source Route from the End Point to the Start Point. All the bits in the Address vector field MUST be set to zero.

4.4. When Measuring A Source Route

If a Source Route is being measured, the Start Point MUST set various MO fields in the following manner:

o RPLInstanceID: MUST be set to the binary value 10000000.

o Compr: MUST be set to specify the number of prefix octets that are elided from the IPv6 addresses in Start Point/End Point Address fields and the Address vector.

o Type (T): MUST be set to one since the MO represents a Measurement Request.

o Hop-by-hop (H): MUST be set to zero.

o Accumulate Route (A): This flag MUST be set to zero.

o Reverse (R): This flag SHOULD be set to one if the Source Route in the Address vector can be reversed and used by the End Point to

send the Measurement Reply message back to the Start Point. Otherwise, this flag MUST be set to zero.

o Back Request (B): This flag MAY be set to one to request the End

Point to send a Measurement Request to the Start Point.

- o Intermediate Reply (I): This flag MUST be set to zero.

- o SequenceNo: Assigned by the Start Point so that it can uniquely identify the Measurement Request and the corresponding Measurement Reply.

- o Num: This field MUST specify the number of address elements, each (16 - Compr) octets in size, inside the Address vector.

- o Index: This field MUST be set to zero to indicate the position in the Address vector of the next hop on the route.

- o Start Point Address: MUST be set to a unicast IPv6 address of the Start Point after eliding Compr number of prefix octets.

- o End Point Address: MUST be set to a unicast IPv6 address of the End Point after eliding Compr number of prefix octets.

- o Address vector:

  * The Address vector MUST contain a complete Source Route from the Start Point to the End Point (excluding the Start Point and the End Point).

  * The IPv6 addresses (with Compr prefix octets elided) in the Address vector MUST be reachable in the Forward direction.

  * If the R flag is set to one, the IPv6 addresses (with Compr prefix octets elided) in the Address vector MUST also be reachable in the Backward direction.

  * Each address appearing in the Address vector MUST be a unicast address.

5. Processing a Measurement Request at an Intermediate Point

A router (an Intermediate Point or the End Point) MAY discard a received MO with no processing to meet any policy-related goal. Such policy goals may include the need to reduce the router's CPU load or to enhance its battery life or to prevent misuse of this mechanism by unauthorized nodes.

A router MUST discard a received MO with no further processing if the value in the Compr field inside the received message is more than what the router considers the length of the common prefix used in IPv6 addresses in the LLN to be.

On receiving an MO, if a router chooses to process the packet further, it MUST check if one of its IPv6 addresses is listed as either the Start Point or the End Point Address. If neither, the router considers itself an Intermediate Point and MUST process the received MO in the following manner.

An Intermediate Point MUST discard the packet with no further processing if the received MO is not a Measurement Request (i.e., T = 0).

Next, the Intermediate Point determines the type of the route being measured (by checking the values of the H flag and the RPLInstanceID field) and processes the received MO accordingly in the manner specified next.

5.1. When Measuring A Hop-by-hop Route with a Global RPLInstanceID

If a Hop-by-hop Route with a global RPLInstanceID is being measured (i.e. H = 1 and RPLInstanceID has a global value), the Intermediate Point MUST process the received Measurement Request in the following manner.

If the Num field inside the received Measurement Request is not set to zero, thereby implying that an Address vector is present, the Intermediate Point MUST discard the received message with no further processing.

If the Intermediate Reply (I) flag is set to one in the received Measurement Request and the Intermediate Point knows the values of the routing metrics, specified in the Metric Container options, for the remainder of the route, it MAY generate a Measurement Reply on the End Point's behalf in the manner specified in Section 6.1 (after including in the Measurement Reply the relevant routing metric values for the complete route being measured). Otherwise, the Intermediate Point MUST process the received message in the following manner.

The Intermediate Point MUST then determine the next hop on the route being measured using the RPLInstanceID and the End Point Address. If the Intermediate Point is the root of the non-storing global DAG along which the received Measurement Request had been traveling so far, it MUST process the received Measurement Request in the following manner:

o If the router does not know how to reach the End Point, it MUST discard the Measurement Request with no further processing and MAY send an ICMPv6 Destination Unreachable (with Code 0 - No Route To Destination) error message to the Start Point.

o Otherwise, unless the router determines the End Point itself to be the next hop, the router MUST make the following changes in the received Measurement Request:

   * Set the H, A, R and I flags to zero (the A and R flags should already be zero in the received message).

   * Leave remaining fields unchanged (the Num field would be modified in next steps). Note that the RPLInstanceID field identifies the non-storing global DAG along which the Measurement Request traveled so far. This information MUST be preserved so that the End Point may use this DAG to send the Measurement Reply back to the Start Point.

   * Insert a new Address vector inside the Measurement Request and specify a Source Route to the End Point inside the Address vector as per the following rules:

      + The Address vector MUST contain a complete route from the router to the End Point (excluding the router and the End Point);

      + The IPv6 addresses (with Compr prefix octets elided) in the Address vector MUST be reachable in the Forward direction;

      + Each address appearing in the Address vector MUST be a unicast address.

   * Specify in the Num field the number of address elements in the Address vector.

   * Set the Index field to zero to indicate the position in the Address vector of the next hop on the route. Thus, Address[0] element contains the address of the next hop on the route.

The Intermediate Point MUST then complete the processing of the received Measurement Request as specified in Section 5.5.

5.2. When Measuring A Hop-by-hop Route with a Local RPLInstanceID With Route Accumulation Off

If a Hop-by-hop Route with a local RPLInstanceID is being measured and the route accumulation is off (i.e., H = 1, RPLInstanceID has a

local value, A = 0), the Intermediate Point MUST process the received Measurement Request in the following manner.

If the Num field inside the received Measurement Request is not set to zero, thereby implying that an Address vector is present, the Intermediate Point MUST discard the received message with no further processing.

The Intermediate Point MUST then determine the next hop on the route being measured using the RPLInstanceID, the End Point Address and the Start Point Address (which represents the DODAGID of the route being measured). If the Intermediate Point can not determine the next hop, it MUST discard the Measurement Request with no further processing and MAY send an ICMPv6 Destination Unreachable (with Code 0 - No Route To Destination) error message to the Start Point. Otherwise, the Intermediate Point MUST complete the processing of the received Measurement Request as specified in Section 5.5.

5.3. When Measuring A Hop-by-hop Route with a Local RPLInstanceID With Route Accumulation On

If a Hop-by-hop Route with a local RPLInstanceID is being measured and the route accumulation in on (i.e., H = 1, RPLInstanceID has a local value, A = 1), the Intermediate Point MUST process the received Measurement Request in the following manner.

If the Num field inside the received Measurement Request is set to zero, thereby implying that an Address vector is not present, the Intermediate Point MUST discard the received message with no further processing.

The Intermediate Point MUST then determine the next hop on the route being measured using the RPLInstanceID, the End Point Address and the

Start Point Address (which represents the DODAGID of the route being measured). If the Intermediate Point can not determine the next hop, it MUST discard the Measurement Request with no further processing and MAY send an ICMPv6 Destination Unreachable (with Code 0 - No Route To Destination) error message to the Start Point. If the index field has value Num - 1 and the next hop is not same as the End Point, the Intermediate Point MUST drop the received Measurement Request with no further processing. In this case, the next hop would have no space left in the Address vector to store its address. Otherwise, the router MUST store one of its unicast IPv6 addresses (after eliding Compr prefix octets) at location Address[Index] and then increment the Index field. The IPv6 address added to the Address vector MUST be reachable in the Backward direction.

The Intermediate Point MUST then complete the processing of the

received Measurement Request as specified in Section 5.5.

5.4. When Measuring A Source Route

If a Source Route is being measured (i.e., H = 0), the Intermediate Point MUST process the received Measurement Request in the following manner.

If the Num field inside the received Measurement Request is set to zero, thereby implying that an Address vector is not present, the Intermediate Point MUST discard the received message with no further processing.

The Intermediate Point MUST verify that the Address[Index] element lists one of its unicast IPv6 addresses, failing which it MUST discard the Measurement Request with no further processing. The Intermediate Point MUST then increment the Index field and use the Address[Index] element as the next hop (unless Index value is now Num). If the Index value is now Num, the Intermediate Point MUST use the End Point Address as the next hop.

The Intermediate Point MUST then complete the processing of the received Measurement Request as specified in Section 5.5.

5.5. Final Processing

The Intermediate Point MUST drop the received Measurement Request with no further processing:

o If the next hop address is not a unicast address; or

o If the next hop is not on-link; or

o If the next hop is not in the same RPL routing domain as the Intermediate Point.

Next, the Intermediate Point MUST update the routing metric objects, inside the Metric Container option(s) inside the Measurement Request, either by updating the aggregated value for the routing metric or by attaching the local values for the metric inside the object. An Intermediate Point can only update the existing metric objects and MUST NOT add any new routing metric object to the Metric Container. An Intermediate Point MUST drop the Measurement Request with no further processing if it cannot update a routing metric object specified inside the Metric Container.

Finally, the Intermediate Point MUST unicast the Measurement Request to the next hop.

6. Processing a Measurement Request at the End Point

On receiving an MO, if a router chooses to process the message further and finds one of its unicast IPv6 addresses listed as the End Point Address, the router considers itself the End Point and MUST process the received MO in the following manner.

The End Point MUST discard the received message with no further processing if it is not a Measurement Request (i.e., T = 0).

If the received Measurement Request traveled on a Hop-by-hop Route with a local RPLInstanceID with route accumulation on (i.e., H = 1, RPLInstanceID has a local value and A = 1), elements Address[0] through Address[Index - 1] in the Address vector contain a complete Source Route from the Start Point to the End Point (excluding the Start Point and the End Point), which the End Point MAY use, after reversal, to reach the Start Point.

If the received Measurement Request traveled on a Source Route and

the Reverse flag is set to one (i.e., H = 0, R = 1), elements Address[0] through Address[Num - 1] in the Address vector contain a complete Source Route from the Start Point to the End Point (excluding the Start Point and the End Point), which the End Point MAY use, after reversal, to reach the Start Point.

The End Point MUST update the routing metric objects in the Metric Container options if required and MAY note the measured values for the complete route (especially, if the received Measurement Request is likely a response to an earlier Measurement Request that the End Point had sent to the Start Point with B flag set to one).

The End Point MUST generate a Measurement Reply message as specified in Section 6.1. If the B flag is set to one in the received Measurement Request, the End Point SHOULD generate a new Measurement Request to measure the cost of its current (or the most preferred) route to the Start Point. The routing metrics used in the new Measurement Request MUST include the routing metrics specified in the received Measurement Request.

6.1. Generating the Measurement Reply

A Measurement Reply MUST have the Type (T) flag set to zero and need not contain the Address vector. The following fields inside a Measurement Reply MUST have the same values as they had inside the corresponding Measurement Request: RPLInstanceID, Compr, SequenceNo, Start Point Address, End Point Address and Metric Container Option(s). The remaining fields inside a Measurement Reply may have any value and MUST be ignored on reception at the Start Point. The

received Measurement Request MAY trivially be converted into a Measurement Reply by setting the Type (T) flag to zero.

A Measurement Reply MUST be unicast back to the Start Point:

o If the Measurement Request traveled along a global DAG, identified by the RPLInstanceID field, the Measurement Reply MAY be unicast back to the Start Point along the same DAG.

o If the Measurement Request traveled along a Hop-by-hop Route with a local RPLInstanceID and accumulated a Source Route from the Start Point to the End Point, this Source Route MAY be used after

reversal to send the Measurement Reply back to the Start Point.

o If the Measurement Request traveled along a Source Route and the R flag inside the received message is set to one, the End Point MAY reverse the Source Route contained in the Address vector and use it to send the Measurement Reply back to the Start Point.

7. Processing a Measurement Reply at the Start Point

When a router receives an MO, it examines if one of its unicast IPv6 addresses is listed as the Start Point Address. If yes, the router is the Start Point and MUST process the received message in the following manner.

If the Start Point discovers that the received MO is not a Measurement Reply or if it has no recollection of sending the corresponding Measurement Request, it MUST discard the received message with no further processing.

The Start Point can use the routing metric objects inside the Metric Container to evaluate the metrics for the measured P2P route. If a routing metric object contains local metric values recorded by routers on the route, the Start Point can make use of these local values by aggregating them into an end-to-end metric according to the aggregation rules for the specific metric. A Start Point is then free to interpret the metrics for the route according to its local policy.

8. Security Considerations

The mechanism defined in this document can potentially be used by a compromised router to send bogus Measurement Requests to arbitrary End Points. Such Measurement Requests may cause CPU overload in the routers in the network, drain their batteries and cause traffic

congestion in the network. Note that some of these problems would occur even if the compromised router were to generate bogus data traffic to arbitrary destinations.

Since a Measurement Request can travel along a Source Route specified

in the Address vector, some of the security concerns that led to the deprecation of Type 0 routing header [RFC5095] may be valid here. To address such concerns, the mechanism described in this document includes several remedies:

o This document requires that a route inserted inside the Address vector must be a strict Source Route and must not include any multicast addresses.

o This document requires that an MO message must not cross the boundaries of the RPL routing domain where it originated. A router must not forward a received MO message further if the next hop belongs to a different RPL routing domain. Hence, any security problems associated with the mechanism would be limited to one RPL routing domain.

o This document requires that a router must drop a received Measurement Request if the next hop address is not on-link or if it is not a unicast address.

The measurement mechanism described in this document may potentially be used by a rogue node to find out key information about the LLN, e.g., the topological features of the LLN (such as the identity of the key nodes in the topology) or the remaining energy levels [RFC6551] in the LLN routers. This information can potentially be used to attack the LLN. To protect against such misuse, this document allows RPL routers implementing this mechanism to not process MO messages (or process such messages selectively) based on a local policy. Further, an LLN deployment may use Secure MO Section 3.2 messages to invoke RPL-provided security mechanisms and prevent misuse of the measurement mechanism by unauthorized nodes.

9. IANA Considerations

This document defines two new RPL messages:

o "Measurement Object" (see Section 3.1), assigned a value TBD1 from the "RPL Control Codes" space [to be removed upon publication: http://www.iana.org/assignments/rpl/rpl.xml#control-codes] [RFC6550]. IANA is requested to allocate TBD1 from the range 0x00-0x7F to indicate a message without security enabled. The string TBD1 in this document should be replaced by the allocated

value. These last two sentences should be removed before publication.

- "Secure Measurement Object" (see Section 3.2), assigned a value TBD2 from the "RPL Control Codes" space [to be removed upon publication: http://www.iana.org/assignments/rpl/rpl.xml#control-codes] [RFC6550]. IANA is requested to allocate TBD2 from the range 0x80-0xFF to indicate a message with security enabled. The string TBD2 in this document should be replaced by the allocated value. These last two sentences should be removed before publication.

Code	Description	Reference
TBD1	Measurement Object	This document
TBD2	Secure Measurement Object	This document

RPL Control Codes

10. Acknowledgements

Authors gratefully acknowledge the contributions of Adrian Farrel, Joel Halpern, Matthias Philipp, Pascal Thubert, Richard Kelsey and Zach Shelby in the development of this document.

11. References

11.1. Normative References

[I-D.ietf-roll-p2p-rpl]
Goyal, M., Baccelli, E., Philipp, M., Brandt, A., and J. Martocci, "Reactive Discovery of Point-to-Point Routes in Low Power and Lossy Networks", draft-ietf-roll-p2p-rpl-15 (work in progress), December 2012.

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

11.2. Informative References

[RFC5095] Abley, J., Savola, P., and G. Neville-Neil, "Deprecation of Type 0 Routing Headers in IPv6", RFC 5095, December 2007.

[RFC5826] Brandt, A., Buron, J., and G. Porcu, "Home Automation Routing Requirements in Low-Power and Lossy Networks", RFC 5826, April 2010.

[RFC5867] Martocci, J., De Mil, P., Riou, N., and W. Vermeylen, "Building Automation Routing Requirements in Low-Power and Lossy Networks", RFC 5867, June 2010.

[RFC6550] Winter, T., Thubert, P., 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, March 2012.

[RFC6551] Vasseur, JP., Kim, M., Pister, K., Dejean, N., and D. Barthel, "Routing Metrics Used for Path Calculation in Low-Power and Lossy Networks", RFC 6551, March 2012.

Authors' Addresses

Mukul Goyal (editor)
University of Wisconsin Milwaukee
3200 N Cramer St
Milwaukee, WI 53211
USA

Phone: +1 414 2295001
Email: mukul@uwm.edu

Emmanuel Baccelli
INRIA

Phone: +33-169-335-511
Email: Emmanuel.Baccelli@inria.fr
URI: http://www.emmanuelbaccelli.org/

Anders Brandt
Sigma Designs
Emdrupvej 26A, 1.
Copenhagen, Dk-2100

Denmark

Phone: +45 29609501
Email: abr@sdesigns.dk

Jerald Martocci
Johnson Controls
507 E Michigan Street
Milwaukee 53202
USA

Phone: +1 414 524 4010
Email: jerald.p.martocci@jci.com