Internet Engineering Task Force (IETF)
Request for Comments: 6998
Category: Experimental
ISSN: 2070-1721

M. Goyal, Ed.
Univ. of Wisconsin Milwaukee
E. Baccelli
INRIA
A. Brandt
Sigma Designs
J. Martocci
Johnson Controls

August 2013

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

Abstract

This document specifies a mechanism that enables a Routing Protocol for Low-power and Lossy Networks (RPL) router to measure the aggregated values of given routing metrics along an existing route towards another RPL router, thereby allowing the router to decide if it wants to initiate the discovery of a better route.

Status of This Memo

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 home and commercial building automation 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:

- 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.

- RPL is a proactive routing protocol and hence requires that all P2P routes 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 an application's requirements, [RFC6997] 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's requirements or the LLN's topological features allow a router to infer these routing constraints implicitly. For example, the application may require that the end-to-end loss rate and/or latency along the route be below certain thresholds, or the LLN topology may be such that a router can safely assume that its destination is 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 (expected transmission count) [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 not be overly strict; otherwise, the P2P-RPL route discovery may fail even though a route exists that is much better than the one currently being used.

This document specifies a mechanism that enables a 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 [RFC6997]. The hope is that experiments with P2P-RPL and the mechanism defined in this document will result in feedback on the utility and benefits of this document, so that a Standards Track version of this document can then be developed.

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].

Additionally, this document uses terminology from [RFC6550], [RFC6554], and [RFC6997]. Further terminology may be found in [ROLL-TERMS]. This document defines the following terms:

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

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

Intermediate Point: A RPL router, other than the Start Point and the End Point, on the P2P route being measured.

The following terms, as already defined in [RFC6997], are redefined in this document in the following manner:

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

Reverse 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 or a Hop-by-hop Route established using RPL [RFC6550] or P2P-RPL [RFC6997]. 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, if 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 the rest of the route.

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

- To accumulate a Source Route for the End Point's use: If a Hop-by-hop Route with a local RPLInstanceID is being measured, the Start Point may require that each Intermediate Point add its global or unique-local 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. This Source Route must consist of global or unique-local IPv6 addresses. An Intermediate Point is not allowed to modify an 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.

3. The Measurement Object (MO)

This document defines two new RPL control message types: the Measurement Object (MO), with code 0x06; and the Secure MO, with code 0x86. 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|   SeqNo   |  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.

- o 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.

- o 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 fields inside the Measurement Request. Here, the Start Point Address field is required to be the same as the DODAGID (the identifier of the Destination-Oriented DAG (DODAG) root) [RFC6550] of the route being measured. The Start Point MUST set the H 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, except under the following circumstance: If the Intermediate Point is the root of the non-storing global DAG along which the Measurement Request had been traveling so far and it intends to insert a Source Route inside the Address vector to direct the Measurement Request towards the End Point, then it MUST set the H flag to zero.

- o Accumulate Route (A): A value of 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 MUST NOT be used (i.e., this flag MUST NOT be set to one) inside a Measurement Request, unless it travels along a Hop-by-hop Route represented by a local RPLInstanceID (i.e., H = 1 and RPLInstanceID has a local value). Route accumulation MAY be used (i.e., this flag MAY be set to one) if the Measurement Request is traveling along a Hop-by-hop Route with a local RPLInstanceID. In this case, if the route accumulation is on, an Intermediate Point adds its unicast global/unique-local 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 of 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 of 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 of 1 in this flag serves as 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 and RPLInstanceID has a global value). Otherwise, this flag MUST be set to zero on transmission and ignored on reception.

- o SeqNo: This is 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.

- o 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, and 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: This is a unicast global or unique-local 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: This is a unicast global or unique-local IPv6 address of the End Point after eliding Compr number of prefix octets.

- Address[0..Num-1]: This field is a vector of unicast global or unique-local 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 one, RPLInstanceID set to a local value, and the A flag set to one), 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 Reverse direction. An Intermediate Point MUST add only a global or unique-local IPv6 address to the Address vector and 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 only of global or unique-local IPv6 addresses that are 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 Reverse direction.

- 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 a Start Point sets various fields inside a Measurement Request in different cases. Section 5 describes how an Intermediate Point processes a received Measurement Request before forwarding it further. Section 6 describes how the End Point processes a received Measurement Request and generates a Measurement Reply. Finally, Section 7 describes how the Start Point processes 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 shown in Figure 7 of [RFC6550], where the base format is the base MO shown in Figure 1. Sections 6.1, 10, and 19 of [RFC6550] describe the RPL security framework. These sections are applicable to the use of Secure MO messages as well, except as constrained in this section. An LLN deployment MUST support the use of Secure MO messages so that it has the ability to invoke RPL-provided security mechanisms and prevent misuse of the measurement mechanism by unauthorized routers.

The Start Point determines whether Secure MO messages are to be used in a particular route measurement and, if yes, the Security Configuration (see definition in [RFC6997]) to be used for that purpose. The Start Point MUST NOT set the "Key Identifier Mode" field to a value of 1 inside this Security Configuration, since this setting indicates the use of a per-pair key, which is not suitable for securing the Measurement Request messages that travel over multiple hops. A router (an Intermediate Point or the End Point) participating in a particular route measurement

- o MUST generate a Secure MO message (a Measurement Request or a Measurement Reply) if the received Measurement Request is a Secure MO. The Security Configuration used in generating a Secure MO message MUST be the same as the one used in the received message.

- o MUST NOT generate a Secure MO message if the received Measurement Request is not a Secure MO.

A router MUST discard a received Measurement Request if it cannot follow the above-mentioned rules. If the Start Point sends a Measurement Request in a Secure MO message using a particular Security Configuration, it MUST discard the corresponding Measurement Reply it receives with no further processing, unless the Measurement Reply is received in a Secure MO message generated with the same Security Configuration as the one used in the Measurement Request.

In the following discussion, any reference to an MO message is also applicable to a Secure MO message, unless noted otherwise.

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 ensure that

- o the next-hop address is a unicast address, and

- o the next hop is on-link, and

- o the next hop is in the same RPL routing domain [RFC6554] as the Start Point,

failing which the Start Point MUST discard the Measurement Request without sending. 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.

The Start Point MUST maintain state for a just-transmitted Measurement Request, for a lifetime duration that is large enough to allow the corresponding Measurement Reply to return. This state consists of the RPLInstanceID, the SeqNo, and the End Point Address fields of the Measurement Request. The lifetime duration for this state is locally determined by the Start Point and may be deployment specific. This state expires when the corresponding Measurement Reply is received or when the lifetime is over, whichever occurs first. Failure to receive the corresponding Measurement Reply before the expiry of a state may occur due to a number of reasons, including the unwillingness on the part of an Intermediate Point or the End Point to process the Measurement Request. The Start Point should take such possibilities into account when deciding whether to generate another Measurement Request for this route. The Start Point MUST discard a received Measurement Reply with no further processing if the state for the corresponding Measurement Request has already expired.

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 and 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: This field MUST be set to the RPLInstanceID of the route being measured.

- o Compr: This field 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): This flag MUST be set to one, since the MO represents a Measurement Request.

- o Hop-by-hop (H): This flag 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 that the End Point 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 SeqNo: This is 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: This field MUST be set to a unicast global/unique-local IPv6 address of the Start Point after eliding Compr number of prefix octets.
- o End Point Address: This field MUST be set to a unicast global/unique-local 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:

- o RPLInstanceID: This field MUST be set to the RPLInstanceID of the route being measured.
- o Compr: This field 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): This flag MUST be set to one, since the MO represents a Measurement Request.
- o Hop-by-hop (H): This flag 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 that the End Point send a Measurement Request to the Start Point.
- o Intermediate Reply (I): This flag MUST be set to zero.
- o SeqNo: This is 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: This field MUST contain the DODAGID value (after eliding Compr number of prefix octets) associated with the route being measured. This DODAGID MUST also be a global or unique-local IPv6 address of the Start Point.
- o End Point Address: This field MUST be set to a unicast global or unique-local 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:

- o RPLInstanceID: This field MUST be set to the RPLInstanceID of the route being measured.
- o Compr: This field 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): This flag MUST be set to one, since the MO represents a Measurement Request.
- o Hop-by-hop (H): This flag MUST be set to one.
- o Accumulate Route (A): This flag MUST be set to one.
- o Reverse (R): This flag MUST be set to zero.

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

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

- o SeqNo: This is 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, 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. This DODAGID MUST also be a global or unique-local IPv6 address of the Start Point.

- o End Point Address: This field MUST be set to a unicast global or unique-local IPv6 address of the End Point after eliding Compr number of prefix octets.

- o Address vector: The Address vector must be large enough to accommodate 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: This field does not have any significance when a Source Route is being measured and hence can be set to any value.

- o Compr: This field 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): This flag MUST be set to one, since the MO represents a Measurement Request.

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

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

- 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.

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

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

- SeqNo: This is 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, inside the Address vector.

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

- Start Point Address: This field MUST be set to a unicast global or unique-local IPv6 address of the Start Point after eliding Compr number of prefix octets.

- End Point Address: This field MUST be set to a unicast global or unique-local IPv6 address of the End Point after eliding Compr number of prefix octets.

- 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).

  * Each address appearing in the Address vector MUST be a unicast global or unique-local IPv6 address. Further, each address MUST have the same prefix as the Start Point Address and the End Point Address. This prefix, whose length in octets is specified in the Compr field, MUST be elided from each address.

  * The IPv6 addresses in the Address vector MUST be reachable in the Forward direction.

  * If the R flag is set to one, the IPv6 addresses in the Address vector MUST also be reachable in the Reverse direction.

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, in order to meet any policy-related goals. Such policy goals may include the need to reduce the router's CPU load, or to enhance its battery life, or to prevent the 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 to be the length of the common prefix used in IPv6 addresses in the LLN.

On receiving an MO, if a router chooses to process the packet further, it MUST determine whether or not one of its IPv6 addresses is listed as either the Start Point or the End Point Address. If not, 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). This is because the End Point unicasts a Measurement Reply directly to the Start Point. So, the Intermediate Point treats a transiting Measurement Reply as a data packet and not a RPL control message.

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 (as 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 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 [RFC4443] 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 the remaining fields unchanged (the Num field would be modified in the 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).

    - + Each address appearing in the Address vector MUST be a unicast global or unique-local IPv6 address. Further, each address MUST have the same prefix as the Start Point Address and the End Point Address. This prefix, whose length in octets is specified in the Compr field, MUST be elided from each address.

    - + The IPv6 addresses in the Address vector MUST be reachable in the Forward direction.

If the router cannot insert an Address vector satisfying the rules mentioned above, 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 [RFC4443] to the Start Point.

* 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, the 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, and 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 cannot 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 [RFC4443] 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 is on (i.e., H = 1, RPLInstanceID has a local value, and 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 cannot 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 [RFC4443] to the Start Point. If the index field has value Num - 1 and the next hop is not the 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 IPv6 addresses at location Address[Index] and then increment the Index field. The IPv6 address added to the Address vector MUST have the following properties:

- o This address MUST be a unicast global or unique-local address.
- o This address MUST have the same prefix as the Start Point Address and the End Point Address. This prefix, whose length in octets is specified in the Compr field, MUST be elided before the address is added to the Address vector.
- o This address MUST be reachable in the Reverse direction.

If the router does not have an IPv6 address that satisfies the properties mentioned above, it MUST discard the Measurement Request with no further processing.

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 global or unique-local IPv6 addresses (minus the prefix whose length in octets is specified in the Compr field), 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 the 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 objects 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 global or unique-local IPv6 addresses (minus the prefix whose length in octets is specified in the Compr field) 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, which the End Point MAY use, after reversal, to reach the Start Point. Note that the Source Route in the Address vector does not include the Start Point and the End Point addresses, and that the individual addresses do not include the common prefix whose length in octets is specified in the Compr field.

If the received Measurement Request traveled on a Source Route and the Reverse flag is set to one (i.e., H = 0 and 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, which the End Point MAY use, after reversal, to reach the Start Point. Again, the Source Route in the Address vector does not include the Start Point and the End Point addresses, and the individual addresses do not include the common prefix whose length in octets is specified in the Compr field.

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 the 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, SeqNo, 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 can, therefore, 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 the MO to see 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 no longer maintains state for 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

In general, the security considerations for the route measurement mechanism described in this document are similar to those for RPL (as described in Section 19 of the RPL specification [RFC6550]). Sections 6.1 and 10 of [RFC6550] describe RPL's security framework, which provides data confidentiality, authentication, replay protection, and delay protection services. This security framework is applicable to the route measurement mechanism described here as well, after taking into account the constraints specified in Section 3.2.

This document requires that all routers participating in a secure invocation of the route measurement process use the Security Configuration chosen by the Start Point. The intention is to avoid compromising the overall security of the route measurement due to some routers using a weaker Security Configuration. A router is allowed to participate in a "secure" route measurement only if it can support the Security Configuration in use, which also specifies the key in use. It does not matter whether the key is preinstalled or dynamically acquired after proper authentication. The router must have the key in use before it can process or generate Secure MO messages. Hence, from the perspective of the route measurement mechanism, there is no distinction between the "preinstalled" and "authenticated" security modes described in the RPL specification [RFC6550]. Of course, if a compromised router has the key being used, it could cause the route measurement to fail, or worse, insert wrong information in Secure MO messages.

A rogue router acting as the Start Point could use the route measurement mechanism defined in this document to measure routes from itself to other routers and thus find out key information about the LLN, e.g., the topological features of the LLN (such as the identity of the key routers in the topology) or the remaining energy levels [RFC6551] in the routers. This information can potentially be used to attack the LLN. A rogue router could also use this mechanism to send bogus Measurement Requests to arbitrary End Points. If sufficient Measurement Requests are sent, then it 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.

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. For example, an LLN deployment might require the use of Secure MO messages generated using a key that could be obtained only after proper authentication. Note that this document requires that an LLN deployment support Secure MO messages so that such policies can be enforced where considered essential.

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 headers [RFC5095] may be valid here. To address such concerns, the mechanism described in this document includes several remedies, in the form of the following requirements:

- o A route inserted inside the Address vector must be a strict Source Route and must not include any multicast addresses.

- o 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 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.

9. IANA Considerations

This document defines two new RPL messages:

- o "Measurement Object" (see Section 3.1), assigned a value of 0x06 from the "RPL Control Codes" space [RFC6550].

- o "Secure Measurement Object" (see Section 3.2), assigned a value of 0x86 from the "RPL Control Codes" space [RFC6550].

Code	Description	Reference
0x06	Measurement Object	This document
0x86	Secure Measurement Object	This document

RPL Control Codes

10. Acknowledgements

The authors gratefully acknowledge the contributions of Ralph Droms, 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

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

[RFC4443] Conta, A., Deering, S., and M. Gupta, "Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification", RFC 4443, March 2006.

[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.

[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, March 2012.

[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, August 2013.

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.

[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.

[ROLL-TERMS] Vasseur, JP., "Terminology in Low power And Lossy Networks", Work in Progress, March 2013.

Authors' Addresses

Mukul Goyal (editor)
University of Wisconsin Milwaukee
3200 N. Cramer St.
Milwaukee, WI 53201
USA

Phone: +1-414-229-5001
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, WI 53202
USA

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