

Internet Engineering Task Force
Internet-Draft
Intended status: Experimental
Expires: August 8, 2013

M. Goyal, Ed.
University of Wisconsin
Milwaukee
E. Baccelli
INRIA
A. Brandt
Sigma Designs
J. Martocci
Johnson Controls
February 4, 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-09

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, thereby allowing the router to decide if it wants to initiate the discovery of a better route.

Status of this Memo

This Internet-Draft is submitted to IETF in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

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

This Internet-Draft will expire on August 8, 2013.

Copyright Notice

Copyright (c) 2013 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents

(<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	3
1.1. Terminology	4
2. Overview	4
3. The Measurement Object (MO)	6
3.1. Format of the base MO	6
3.2. Secure MO	10
4. Originating a Measurement Request	11
4.1. When Measuring A Hop-by-hop Route with a Global RPLInstanceID	12
4.2. When Measuring A Hop-by-hop Route with a Local RPLInstanceID With Route Accumulation Off	13
4.3. When Measuring A Hop-by-hop Route with a Local RPLInstanceID With Route Accumulation On	14
4.4. When Measuring A Source Route	15
5. Processing a Measurement Request at an Intermediate Point	16
5.1. When Measuring A Hop-by-hop Route with a Global RPLInstanceID	17
5.2. When Measuring A Hop-by-hop Route with a Local RPLInstanceID With Route Accumulation Off	18
5.3. When Measuring A Hop-by-hop Route with a Local RPLInstanceID With Route Accumulation On	19
5.4. When Measuring A Source Route	19
5.5. Final Processing	20
6. Processing a Measurement Request at the End Point	20
6.1. Generating the Measurement Reply	21
7. Processing a Measurement Reply at the Start Point	22
8. Security Considerations	22
9. IANA Considerations	23
10. Acknowledgements	24
11. References	24
11.1. Normative References	24
11.2. Informative References	24
Authors' Addresses	25

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 (Expected Number of Transmissions) [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 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. 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 and it will be revised and progressed on the Standards Track based on this feedback.

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.

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 [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

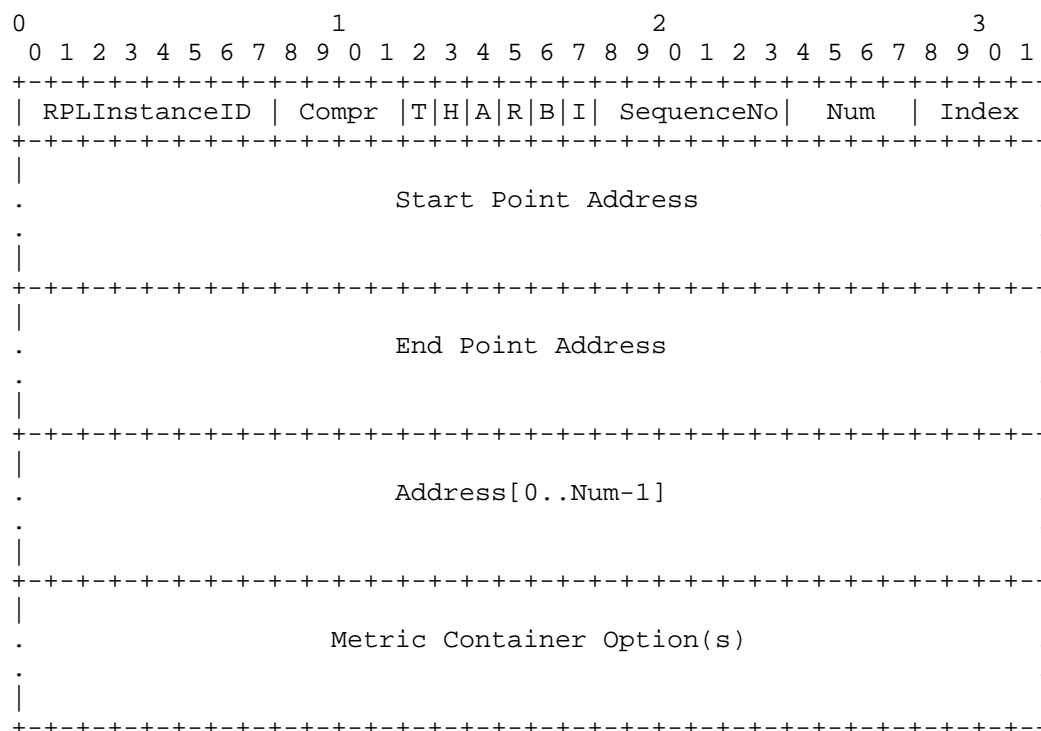


Figure 1: Format of the base Measurement Object (MO)

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

- o 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).
- o 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 same as the DODAGID (the identifier of the destination-oriented DAG 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 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.
- o 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.
- 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, 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.
- o 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.
- o End Point Address: A unicast IPv6 address of the End Point after eliding Compr number of prefix octets.
- o 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 Reverse 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 Reverse 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 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 generate 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 in Figure 7 of [RFC6550], where the base format is the base MO shown in Figure 1.

An LLN deployment MUST support the use of Secure MO messages to have the ability to invoke RPL-provided security mechanisms and prevent misuse of the measurement mechanism by unauthorized routers.

In the following discussion, any reference to MO message is also applicable to 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 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 just transmitted Measurement Request for a life time duration that is large enough to allow the corresponding Measurement Reply to return. This state consists of the RPLInstanceID, the SequenceNo and the End Point Address fields of the Measurement Request. The life time 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 life time 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 unwillingness on part of an Intermediate Point or the End Point to process the Measurement Request. The Start Point should take such possibilities in 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, 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:

- 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 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 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.
- 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:

- 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 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 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 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, 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 Reverse 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 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 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 [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, $A = 1$), 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 [RFC4443] 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 Reverse 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 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 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

The mechanism defined in this document can potentially be used by a compromised router 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.

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 router 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. 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 is required to support Secure MO (Section 3.2) messages to have the ability to invoke RPL-provided security mechanisms and prevent misuse of the measurement mechanism by unauthorized routers.

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.
- o "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-16 (work in progress), February 2013.
- [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.

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.

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

Internet Engineering Task Force
Internet-Draft
Intended status: Experimental
Expires: August 7, 2013

M. Goyal, Ed.
University of Wisconsin
Milwaukee
E. Baccelli
M. Philipp
INRIA
A. Brandt
Sigma Designs
J. Martocci
Johnson Controls
February 3, 2013

Reactive Discovery of Point-to-Point Routes in Low Power and Lossy
Networks
draft-ietf-roll-p2p-rpl-16

Abstract

This document specifies a point-to-point route discovery mechanism, complementary to the RPL core functionality. This mechanism allows an IPv6 router to discover "on demand" routes to one or more IPv6 routers in the LLN such that the discovered routes meet specified metrics constraints.

Status of this Memo

This Internet-Draft is submitted to IETF in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

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

This Internet-Draft will expire on August 7, 2013.

Copyright Notice

Copyright (c) 2013 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal

Provisions Relating to IETF Documents
 (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	3
2. The Use Cases	4
3. Terminology	4
4. Applicability	5
5. Functional Overview	6
6. P2P Route Discovery Mode Of Operation	9
6.1. Setting a P2P Mode DIO	9
7. New RPL Control Message Options	12
7.1. P2P Route Discovery Option (P2P-RDO)	13
7.2. Data Option	16
8. The Discovery Reply Object (DRO)	17
8.1. Secure DRO	19
8.2. Setting a P2P-RDO Carried in a Discovery Reply Object	19
9. P2P-RPL Route Discovery By Creating a Temporary DAG	20
9.1. Joining a Temporary DAG	20
9.2. Trickle Operation For P2P Mode DIOs	20
9.3. Processing a P2P Mode DIO	23
9.4. Additional Processing of a P2P Mode DIO At An Intermediate Router	24
9.5. Additional Processing of a P2P Mode DIO At The Target	25
9.6. Processing a DRO At An Intermediate Router	26
9.7. Processing a DRO At The Origin	28
10. The Discovery Reply Object Acknowledgement (DRO-ACK)	29
11. Packet Forwarding Along a Route Discovered Using P2P-RPL	30
12. Interoperability with Core RPL	30
13. Security Considerations	31
14. IANA Considerations	32
14.1. Additions to Mode of Operation	32
14.2. Additions to RPL Control Message Options	32
14.3. Additions to RPL Control Codes	33
15. Acknowledgements	34
16. References	34
16.1. Normative References	34
16.2. Informative References	35
Authors' Addresses	35

1. Introduction

Targeting Low power and Lossy Networks (LLNs), the IPv6 Routing Protocol for LLNs (RPL) [RFC6550] provides paths along a Directed Acyclic Graph (DAG) rooted at a single router in the network. Establishment and maintenance of a DAG is performed by routers in the LLN using Destination-Oriented DAG (DODAG) Information Object (DIO) messages. When two arbitrary routers (neither of which is the DAG's root) need to communicate, the data packets are restricted to travel only along the links in the DAG. Such point-to-point (P2P) routing functionality may not be sufficient for several Home and Building Automation applications [RFC5826] [RFC5867] due to the following reasons:

- o The need to pre-establish routes: each potential destination in the network must declare itself as such ahead of the time a source needs to reach it.
- o The need to route only along the links in the DAG: A DAG is built to optimize the routing cost to reach the root. Restricting P2P routes to use only the in-DAG links may result in significantly suboptimal routes and severe traffic congestion near the DAG root.

This document describes an extension to core RPL (i.e., the RPL functionality described in [RFC6550]) that enables an IPv6 router in the LLN to discover routes to one or more IPv6 routers in the LLN "on demand". The discovered routes may not be the best available but are guaranteed to meet the specified routing metric constraints. Thus, such routes are considered "good enough" from the application's perspective. This reactive P2P route discovery mechanism is henceforth referred to as P2P-RPL.

A mechanism to measure the end-to-end cost of an existing route is specified in [I-D.ietf-roll-p2p-measurement]. As discussed in Section 4, measuring the end-to-end cost of an existing route may help decide whether to initiate the discovery of a better route using P2P-RPL and the metric constraints to be used for this purpose.

This document is presented as an Experimental specification to facilitate P2P-RPL's deployment in LLN scenarios where reactive P2P route discovery is considered useful or necessary. It is anticipated that, once sufficient operational experience has been gained, this specification will be revised to progress it on to the Standards Track. Experience reports regarding P2P-RPL implementation and deployment are encouraged particularly with respect to:

- o The values in the default DODAG Configuration Option (Section 6.1);

- o The rules governing Trickle operation (Section 9.2);
- o The utility and the implementation complexity of the Data Option (Section 7.2) that provides a facility to piggyback time-critical application data on the routing messages;
- o The utility and the implementation complexity of allowing multiple Target addresses in a P2P-RPL route discovery.

2. The Use Cases

One use case, common in home [RFC5826] and commercial building [RFC5867] environments, involves a device (say a remote control or an airduct controller) that suddenly needs to communicate with another device (say a lamp or a humidity sensor) to which it does not already have a route. In this case, the remote control (or the airduct controller) must be able to discover a route to the lamp (or the humidity sensor) "on demand".

Another use case, common in a commercial building environment, involves a large LLN deployment where P2P communication along a particular DAG among hundreds (or thousands) of routers creates severe traffic congestion near that DAG's root, and thus routes across this DAG are desirable.

Other use cases involve scenarios where energy or latency constraints are not satisfied by the P2P routes along an existing DAG because they involve traversing many more intermediate routers than necessary to reach the destination.

3. 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]. This document introduces the following terms:

Origin : The IPv6 router initiating the P2P-RPL route discovery.

Target : The IPv6 router at the other end point of the P2P route(s) to be discovered. A P2P-RPL route discovery can discover routes to multiple Targets at the same time.

Intermediate Router: An IPv6 router that is neither the Origin nor a Target.

Forward direction: The direction from the Origin to the Target.

Reverse direction: The direction from the Target to the Origin.

Forward Route: A route in the Forward direction.

Reverse Route: A route in the Reverse direction.

Bidirectional Route: A route that can be used in both Forward and Reverse directions.

Source Route: A complete and ordered list of routers that can be used by a packet to travel from a source to a destination node.

Hop-by-hop Route: The route characterized by each router on the route using its routing table to determine the next hop on the route.

4. Applicability

A route discovery using P2P-RPL may be performed by an Origin when no route exists between itself and the Target(s) or when the existing routes do not satisfy the application requirements. P2P-RPL is designed to discover Hop-by-hop or Source Routes to one or more Targets such that the discovered routes meet the specified constraints. In some application contexts, the constraints that the discovered routes must satisfy are intrinsically known or can be specified by the application. For example, an Origin that expects its Targets to be less than 5 hops away may use "hop-count < 5" as the constraint. In other application contexts, the Origin may need to measure the cost of the existing route to a Target to determine the constraints. For example, an Origin that measures the total ETX along its current route to a Target to be 20 may use "ETX < x*20", where x is a fraction that the Origin decides, as the constraint. A mechanism to measure the cost of an existing route between two IPv6 routers is specified in [I-D.ietf-roll-p2p-measurement]. If there is no existing route between the Origin and the Target(s) or the cost measurement for the existing routes fails, the Origin will have to guess the constraints to be used in the initial route discovery. Once, the initial route discovery succeeds or fails, the Origin will have a better estimate for the constraints to be used in the subsequent route discovery.

P2P-RPL may result in discovery of better P2P routes than the ones available along a global DAG designed to optimize routing cost to the

DAG's root. The improvement in route quality depends on a number of factors including the network topology, the "distance" between the Origin and the Target (in terms of the routing metrics in use) and the prevalent conditions in the network. In general, a P2P-RPL route may be better than the one along a global DAG if the Origin and the Target are nearby. Similarly, a P2P-RPL route may not be much better than the one along a global DAG if the Origin and the Target are far apart. Note that, even when P2P-RPL routes are not much better than those along a global DAG, P2P-RPL routes may still be able to avoid congestion that might occur near the root if the routing takes place only along a global DAG. In general, the costs associated with a P2P-RPL route discovery (in terms of the control messages, mostly DIOs, generated) increases with the distance between the Origin and the Target. However, it is possible to limit the cost of route discovery by carefully setting the routing constraints, the Trickle parameters (that govern the DIO generation) and the lifetime of the temporary DAG created for the route discovery. A network designer may take into consideration both the benefits (potentially better routes; no need to maintain routes proactively; avoid congestion near the global DAG's root) and costs when using P2P-RPL. The latency associated with a P2P-RPL route discovery again depends on the distance between the Origin and the Target and the Trickle parameters.

Like core RPL [RFC6550], P2P-RPL operation requires links to have bidirectional reachability. Routers participating in a P2P-RPL route discovery must ensure that

- o Links that do not have bidirectional reachability do not become part of the route being discovered; and
- o IPv6 addresses belonging to egress-only or ingress-only interfaces do not become part of the route being discovered.

5. Functional Overview

This section contains a high level description of P2P-RPL.

A P2P-RPL route discovery takes place by forming a DAG rooted at the Origin. As is the case with core RPL, P2P-RPL uses IPv6 link-local multicast DIO messages to establish a DAG. However, unlike core RPL, this DAG is temporary in nature and routers in the DAG leave once the DAG's life time is over. The sole purpose of DAG creation is to discover routes to the Target(s) and DIOs serve as the route discovery messages. Each router joining the DAG determines a rank for itself in the DAG and ignores the subsequent DIOs received from lower (higher in numerical value) ranked neighbors. Thus, the route

discovery messages propagate away from the Origin rather than return back to it. As in core RPL, DIO generation at a router is controlled by a Trickle timer [RFC6206] that allows a router to avoid generating unnecessary messages while providing protection against packet loss. P2P-RPL also uses the routing metrics [RFC6551], objective functions and packet forwarding framework [RFC6554][RFC6553] developed for core RPL.

An Origin may use P2P-RPL to discover routes to one or more Target(s) identified by one or more unicast/multicast addresses. P2P-RPL allows for the discovery of one Hop-by-hop Route or up to four Source Routes per Target. The discovered routes are guaranteed to meet the specified routing metric constraints but may not be the best available. P2P-RPL may fail to discover any route if the specified routing constraints are overly strict. P2P-RPL allows an Origin to piggyback time-critical application data on the DIO messages for delivery to the Target(s).

The Origin initiates a P2P-RPL route discovery by forming a temporary DAG rooted at itself. The DIOs used to create the temporary DAG are identified by a new Mode of Operation (P2P Route Discovery mode defined in Section 6). The DIOs listing the P2P Route Discovery mode as the Mode of Operation are henceforth referred to as the P2P mode DIOs. A P2P mode DIO always carries exactly one P2P Route Discovery Option (defined in Section 7.1) in which the Origin specifies the following information:

- o The IPv6 address of a Target. This could be a unicast address or a multicast address. Any additional Targets may be specified by including one or more RPL Target Options [RFC6550] inside the DIO.
- o The nature of the route(s) to be discovered: Hop-by-hop or Source Routes. This specification allows for the discovery of one Hop-by-hop Route or up to four Source Routes per Target.
- o The desired number of routes (if Source Routes are being discovered).
- o Whether the Target(s) should send Discovery Reply Object (DRO) messages (defined in Section 8) back to the Origin on receiving a DIO message. A DRO message carries a discovered Source Route back to the Origin or establishes a Hop-by-hop Route between the Origin and the Target. By not allowing the generation of DRO messages, an Origin can use P2P-RPL as purely a mechanism to deliver time-critical application data to the Target(s).

A P2P Route Discovery Option also accumulates a route from the Origin to a Target as the routers join the temporary DAG.

A P2P mode DIO MAY also carry:

- o One or more Metric Container Options to specify:
 - * The relevant routing metrics.
 - * The constraints that the discovered route must satisfy. These constraints also limit how far the DIOs message may travel.
- o One or more RPL Target options to specify additional unicast or multicast Targets.
- o One Data Option (defined in Section 7.2) to carry time-critical application-level data to be delivered to the Target(s).

As the routers join the temporary DAG, they keep track of the best route(s) (so far from the Origin) they have seen and advertise these routes, along with the corresponding routing metrics, in their P2P mode DIOs. A router, including the Target(s), discards a received P2P mode DIO if the aggregated routing metrics on the route advertised by the DIO do not satisfy the listed constraints. These constraints can be used to limit the propagation of P2P mode DIO messages. A router may also discard a received P2P mode DIO if it does not wish to be a part of the discovered route due to limited resources or due to policy reasons.

When a Target receives a P2P mode DIO, it forwards the data in the Data Option, if present, to the higher layer. Since the links in the discovered route have bidirectional reachability (Section 7.1), the Target may remember the discovered route for use as a Source Route to reach the Origin. If the Origin has requested DRO messages to be sent back, the Target may select the route contained in the received DIO for further processing as described next. This document does not specify a particular method for the Target to use to select a route for further processing. Example methods include selecting any route that meets the constraints or selecting the best route(s) discovered over a certain time period.

If one or more Source Route(s) are being discovered, the Target sends the selected Source Route(s) to the Origin via DRO messages with one DRO message carrying one discovered route. On receiving a DRO message, the Origin stores the discovered route in its memory. This specification allows the Origin to discover up to four Source Routes per Target, thereby allowing the Origin to have sufficient ready-to-use alternatives should one or more of these routes fail. If a Hop-by-hop Route is being discovered, the Target sends a DRO message containing the selected route to the Origin. The DRO message travels back to the Origin along the selected route, establishing state for

the Forward Route in the routers on the path. The Target may include a Data Option in a DRO message to deliver any time-critical application data to the Origin.

The Target may request the Origin to acknowledge the receipt of a DRO message by sending back a DRO Acknowledgement (DRO-ACK) message (defined in Section 10). The Origin unicasts a DRO-ACK message to the Target. If the Target does not receive the requested DRO-ACK within a certain time interval of sending a DRO, it resends the DRO message (up to a certain number of times) carrying the same route as before.

The use of trickle timers to delay the propagation of DIO messages may cause some nodes to generate these messages even when the desired routes have already been discovered. In order to preempt the generation of such unnecessary messages, the Target may set a "Stop" flag in the DRO message to let the nodes in the LLN know about the completion of the route discovery process. The routers receiving such a DRO should not generate any more DIOs for this temporary DAG. Neither should they process any received DIOs for this temporary DAG in future. However, such routers must still process the DROs received for this temporary DAG.

6. P2P Route Discovery Mode Of Operation

This section specifies a new RPL Mode of Operation (MOP), P2P Route Discovery Mode (or P2P mode, for short), with value TBD1. A DIO message, listing P2P mode as the MOP, is identified as performing a P2P-RPL route discovery by creating a temporary DAG. A P2P mode DIO MUST carry exactly one P2P Route Discovery Option (specified in Section 7.1).

6.1. Setting a P2P Mode DIO

The Base Object in a P2P mode DIO message MUST be set in the following manner:

- o RPLInstanceID: RPLInstanceID MUST be a local value as described in Section 5.1 of [RFC6550]. The Origin MUST NOT reuse an RPLInstanceID for a route discovery if its previous route discovery using this RPLInstanceID might still be going on. As described in Section 7.1, the Life Time parameter in the P2P Route Discovery Option specifies the time duration the route discovery lasts. So, the Origin MUST NOT reuse an RPLInstanceID in a route discovery until the Life Time of its previous route discovery using this RPLInstanceID is over. When initiating a new route discovery to a particular Target, the Origin MUST NOT reuse the

RPLInstanceID used in a previous route discovery to this Target if the previously discovered routes might still exist. The Default Lifetime and Lifetime Unit parameters in the DODAG Configuration Option specify the lifetime of the state the routers, including the Origin and the Target, maintain for a Hop-by-hop or a Source Route discovered using P2P-RPL. Thus, an Origin can safely reuse an RPLInstanceID to discover a new route to a Target if the lifetime of all previously discovered routes to this Target using this RPLInstanceID is over.

- o Version Number: MUST be set to zero. The temporary DAG used for P2P-RPL route discovery does not exist long enough to have new versions (the Life Time parameter inside the P2P Route Discovery Option, defined in Section 7.1, specifies the life time of the temporary DAG).
- o Grounded (G) Flag: This flag MUST be set to one. Unlike a global RPL instance, the concept of a floating DAG, used to provide connectivity within a sub-DAG detached from a grounded DAG, does not apply to a local RPL instance. Hence, an Origin MUST always set the G flag to one when initiating a P2P-RPL route discovery. Further, clause 3 of Section 8.2.2.2 in [RFC6550] does not apply and a node MUST NOT initiate a new DAG if it does not have any parent left in a P2P-RPL DAG.
- o Mode of Operation (MOP): MUST be set to TBD1, corresponding to P2P Route Discovery mode.
- o DTSN: MUST be set to zero on transmission and ignored on reception.
- o DODAGPreference (Prf): This field MUST be set to zero (least preferred).
- o DODAGID: This field MUST be set to an IPv6 address of the Origin.
- o The other fields in the DIO Base Object can be set in the desired fashion as per the rules described in [RFC6550].

A received P2P mode DIO MUST be discarded if it does not follow the above-listed rules regarding the RPLInstanceID, Version Number, G flag, MOP and Prf fields inside the base object.

The DODAG Configuration Option, inside a P2P mode DIO MUST be set in the following manner:

- o The Origin MUST set the MaxRankIncrease parameter to zero to disable local repair of the temporary DAG. A received P2P mode

DIO MUST be discarded if the MaxRankIncrease parameter inside the DODAG Configuration Option is not zero.

- o The Origin SHOULD set the Trickle parameters (DIOIntervalDoublings, DIOIntervalMin, DIORedundancyConstant) as recommended in Section 9.2.
- o The Origin sets the Default Lifetime and Lifetime Unit parameters to indicate the lifetime of the state the routers, including the Origin and the Target(s), maintain for a Hop-by-hop or a Source Route discovered using P2P-RPL.
- o The Origin sets the other fields in the DODAG Configuration Option, including the OCP identifying the Objective function, in the desired fashion as per the rules described in [RFC6550].
- o An Intermediate Router (or a Target) MUST set various fields in the DODAG Configuration Option in the outgoing P2P mode DIOs to the values they had in the incoming P2P mode DIOs for this DAG.

A default DODAG Configuration Option comes in effect if a P2P mode DIO does not carry an explicit one. The default DODAG Configuration Option has the following parameter values:

- o Authentication Enabled: 0
- o DIOIntervalMin: 6, which translates to 64ms as the value for Imin parameter in Trickle operation. This value is roughly one order of magnitude larger than the typical transmission delay on IEEE 802.15.4 links and corresponds to the recommendation in Section 9.2 for well-connected topologies.
- o DIORedundancyConstant: 1. See the discussion in Section 9.2.
- o MaxRankIncrease: 0 (to disable local repair of the temporary DAG).
- o Default Lifetime: 0xFF, to correspond to infinity.
- o Lifetime Unit: 0xFFFF, to correspond to infinity.
- o Objective Code Point: 0, i.e., OF0 [RFC6552] is the default objective function.
- o The remaining parameters have default values as specified in [RFC6550].

Individual P2P-RPL deployments are encouraged to share their experience with these default values with ROLL working group to help

guide the development of standards track version of the protocol.

The routing metrics and constraints [RFC6551] used in P2P-RPL route discovery are included in one or more Metric Container Options [RFC6550] inside the P2P mode DIO. Note that a DIO need not include a Metric Container if OF0 is the objective function in effect. In that case, a P2P mode DIO may still specify an upper limit on the maximum rank, that a router may have in the temporary DAG, inside the P2P Route Discovery Option (described in Section 7.1).

A P2P mode DIO:

- o MUST carry one (and only one) P2P Route Discovery Option (described in Section 7.1). The P2P Route Discovery Option allows for the specification of one unicast or multicast address for the Target. A received P2P mode DIO MUST be discarded if it does not contain exactly one P2P Route Discovery Option.
- o MAY carry one or more RPL Target Options to specify additional unicast/multicast addresses for the Target.
- o MAY carry one or more Metric Container Options to specify routing metrics and constraints.
- o MAY carry one Data Option (described in Section 7.2) containing time-critical application data to be delivered to the Target(s). A received P2P mode DIO MUST be discarded if it contains multiple Data Options.
- o MAY carry one or more Route Information Options [RFC6550]. In the context of P2P-RPL, a Route Information Option advertizes to the Target(s) the Origin's connectivity to the prefix specified in the option.
- o MAY carry one or more Prefix Information Options subject to the usage and rules specified in Section 6.7.10 in [RFC6550].

7. New RPL Control Message Options

This document defines two new RPL control message options: the P2P Route Discovery Option and the Data Option.

7.1. P2P Route Discovery Option (P2P-RDO)

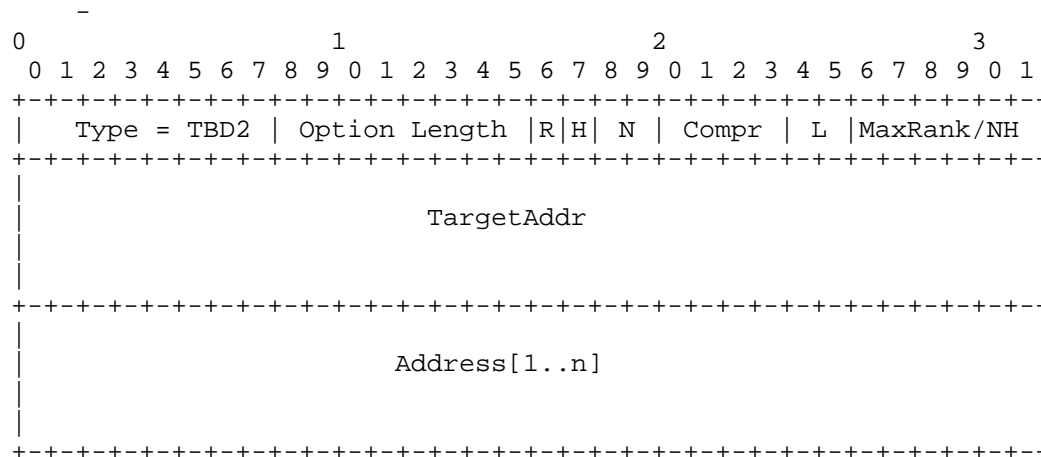


Figure 1: Format of P2P Route Discovery Option (P2P-RDO)

The format of a P2P Route Discovery Option (P2P-RDO) is illustrated in Figure 1. A P2P mode DIO and a DRO (defined in Section 8) message MUST carry exactly one P2P-RDO. A P2P-RDO consists of the following fields:

- o Option Type: TBD2.
- o Option Length: 8-bit unsigned integer, representing the length in octets of the option, not including the Option Type and Option Length fields.
- o Reply (R): The Origin sets this flag to one to allow the Target(s) to send DRO messages back to the Origin. If this flag is zero, a Target MUST NOT generate any DRO message.
- o Hop-by-hop (H): This flag is valid only if the R flag is set to one. The Origin sets this flag to one if it desires Hop-by-hop Routes. The Origin sets this flag to zero if it desires Source Routes. This specification allows for the establishment of one Hop-by-hop route or up to four Source Routes per Target. The Hop-by-hop Route is established in the Forward direction, i.e. from the Origin to the Target. This specification does not allow for the establishment of Hop-by-hop Routes in the Reverse direction.
- o Number of Routes (N): This field is valid only if the R flag is one and H flag is zero, i.e. the Targets are allowed to generate

DRO messages carrying discovered Source Routes back to the Origin. In this case, the value in the N field plus one indicates the number of Source Routes that each Target should convey to the Origin. When Hop-by-hop Routes are being discovered, the N field MUST be set to zero on transmission and ignored on reception.

- o Compr: 4-bit unsigned integer indicating the number of prefix octets that are elided from the Target field and the Address vector. For example, Compr value will be zero if full IPv6 addresses are carried in the Target field and the Address vector.
- o Life Time (L): A 2-bit field that indicates the life time of the temporary DAG, i.e., the exact duration a router joining the temporary DAG MUST maintain its membership in the DAG. The mapping between the values in this field and the life time of the temporary DAG is as follows:

- * 0x00: 1 second;
- * 0x01: 4 seconds;
- * 0x02: 16 seconds;
- * 0x03: 64 seconds;

The Origin sets this field based on its expectation regarding the time required for the route discovery to complete, which includes the time required for the DIOs to reach the Target(s) and the DROs to travel back to the Origin. The time required for the DIOs to reach the Target(s) would in turn depend on the Trickle parameters (Imin and the redundancy constant) as well as the expected distance (in terms of hops and/or ETX) to the Target(s). While deciding the temporary DAG's lifetime, the Origin should also take in account the fact that all routers joining the temporary DAG would need to stay in the DAG for this much time.

- o MaxRank/NH:
 - * When a P2P-RDO is included in a P2P mode DIO, this field indicates the upper limit on the integer portion of the rank (calculated using the DAGRank() macro defined in [RFC6550]) that a router may have in the temporary DAG being created. An Intermediate Router MUST NOT join a temporary DAG being created by a P2P mode DIO if the integer portion of its rank would be equal to or higher (in numerical value) than the MaxRank limit. A Target can join the temporary DAG at a rank whose integer portion is equal to the MaxRank. A router MUST discard a received P2P mode DIO if the integer part of the advertized

rank equals or exceeds the MaxRank limit. A value 0 in this field indicates that the MaxRank is infinity.

- * When a P2P-RDO is included in a DRO message, this field indicates the index of the next hop address inside the Address vector.
- o TargetAddr: An IPv6 address of the Target after eliding Compr number of prefix octets. When the P2P-RDO is included in a P2P mode DIO, this field may contain a unicast address or a multicast address. Any additional Target addresses can be specified by including one or more RPL Target Options [RFC6550] in the DIO. When the P2P-RDO is included in a DRO, this field MUST contain a unicast IPv6 address of the Target generating the DRO.
- o Address[1..n]: A vector of IPv6 addresses representing a complete route so far in the Forward direction:
 - * Each element in the Address vector has size $(16 - \text{Compr})$ octets and MUST contain a valid IPv6 address with first Compr octets elided.
 - * The total number of elements inside the Address vector is given by $n = (\text{Option Length} - 2 - (16 - \text{Compr})) / (16 - \text{Compr})$.
 - * IPv6 addresses of ingress-only (or egress-only) router interfaces MUST NOT be added to the Address vector. This allows the route accumulated in the Address vector to be a Bidirectional Route that can be used by a Target to send a DRO message to the Origin.
 - * The Address vector MUST carry the accumulated route in the Forward direction, i.e., the first element in the Address vector must contain the IPv6 address of the router next to the Origin and so on.
 - * The Origin and Target addresses MUST NOT be included in the Address vector.
 - * A router adding its address to the vector MUST ensure that any of its addresses do not already exist in the vector. A Target specifying a complete route in the Address vector MUST ensure that the vector does not contain any address more than once.
 - * The Address vector MUST NOT contain any multicast addresses.

7.2. Data Option

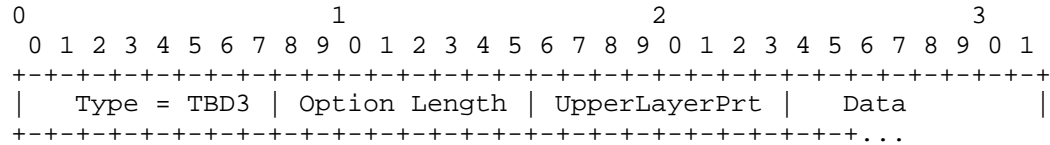


Figure 2: Format of Data Option

The format of a Data Option is illustrated in Figure 2. A P2P mode DIO and a DRO (defined in Section 8) message MAY carry one Data Option. A P2P-RDO consists of the following fields:

- o Option Type: TBD3.
- o Option Length: An 8-bit unsigned integer, representing the length in octets of the option, not including the Option Type and Option Length fields.
- o Upper Layer Protocol: An 8-bit field that identifies the upper layer protocol header with which the information in the Data field starts. The protocol identifiers used in this field are same as those defined in IANA's "Protocol Numbers" registry [PROTOCOL].
- o Data: If the Data Option is contained in a DIO, this field contains application data to be delivered to the Target(s). If the Data Option is contained in a DRO, this field contains application data to be delivered to the Origin.

If the Origin chooses to include a Data Option inside its DIO, it MUST include the same Data Option in all its future DIO transmissions for this temporary DAG. An Intermediate Router MUST NOT modify the Data Option received inside a parent's DIO and MUST include this Data Option in all its future DIO transmissions for this temporary DAG. The same is true for a Target that needs to propagate the DIOs further (required when the route discovery involves multiple Targets). If a Target chooses to include a Data Option inside a DRO, it MUST include the same Data Option in all retransmissions of this DRO message and MUST NOT include a different Data Option in any other DRO messages it generates for this route discovery. Also, an Intermediate Router, which needs to forward a received DRO message further, MUST include in the forwarded message a verbatim copy of the Data Option found inside the received message.

Note that the data inside a Data Option has the same level of security as the DIO/DRO message it is part of. A P2P-RPL deployment

SHOULD take in consideration the security requirements of the data being sent inside the Data Options when deciding the overall security requirements. Further, note that P2P-RPL does not guarantee successful delivery of the data contained in a Data Option.

8. The Discovery Reply Object (DRO)

This section defines two new RPL Control Message types, the Discovery Reply Object (DRO), with code TBD4, and the Secure DRO, with code TBD5. A DRO serves one of the following functions:

- o Carry a discovered Source Route from a Target to the Origin;
- o Establish a Hop-by-hop Route as it travels from a Target to the Origin.

A DRO message MAY serve the function of letting the routers in the LLN know that a P2P-RPL route discovery is complete and no more DIO messages need to be generated for the corresponding temporary DAG. A DRO message MAY also carry time-critical application data from the Target to the Origin in a Data Option. A DRO message MUST carry one (and only one) P2P-RDO whose TargetAddr field MUST contain a unicast IPv6 address of the Target that generates the DRO. A DRO message travels from the Target to the Origin via link-local multicast along the route specified inside the Address vector in the P2P-RDO, as included in the DRO.

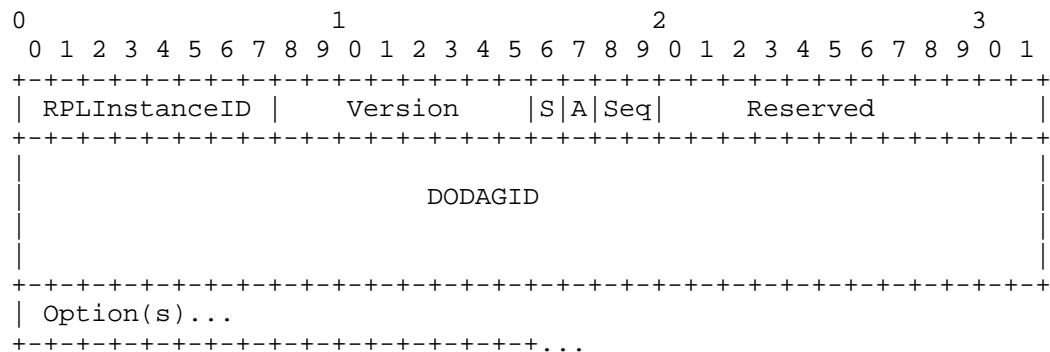


Figure 3: Format of the base Discovery Reply Object (DRO)

The format of the base Discovery Reply Object (DRO) is shown in Figure 3. A base DRO consists of the following fields:

- o RPLInstanceID: The RPLInstanceID of the temporary DAG used for route discovery.
- o Version: The Version of the temporary DAG used for route discovery. Since a temporary DAG always has value zero for the Version, this field MUST always be set to zero.
- o Stop (S): This flag, when set to one by a Target, indicates that the P2P-RPL route discovery is over. All the routers receiving such a DRO, including the ones not listed in the route carried inside P2P-RDO,
 - * SHOULD NOT process any more DIOs received for this temporary DAG;
 - * SHOULD NOT generate any more DIOs for this temporary DAG;
 - * SHOULD cancel any pending DIO transmission for this temporary DAG.

Note that the Stop flag serves to stop further DIO generation/processing for a P2P-RPL route discovery but it does not affect the processing of DRO messages at either the Origin or the Intermediate Routers. In other words, a router (the Origin or an Intermediate Router) MUST continue to process the DRO messages even if an earlier DRO message (with the same RPLInstanceID and DODAGID fields) had the Stop flag set to one. When set to zero, this flag does not imply any thing and MUST be ignored on reception.

- o Ack Required (A): This flag, when set to one by the Target, indicates that the Origin MUST unicast a DRO-ACK message (defined in Section 10) to the Target when it receives the DRO.
- o Sequence Number (Seq): This 2-bit field indicates the sequence number for the DRO. This field is relevant when the A flag is set to one, i.e., the Target requests an acknowledgement from the Origin for a received DRO. The Origin includes the RPLInstanceID, the DODAGID and the Sequence Number of the received DRO inside the DRO-ACK message it sends back to the Target.
- o Reserved: These bits are reserved for future use. These bits MUST be set to zero on transmission and MUST be ignored on reception.
- o DODAGID: The DODAGID of the temporary DAG used for route discovery. The DODAGID also identifies the Origin. The RPLInstanceID, the Version and the DODAGID together uniquely identify the temporary DAG used for route discovery and can be

copied from the DIO message advertizing the temporary DAG.

o Options: The DRO message:

- * MUST carry one (and only one) P2P-RDO that MUST specify a complete route between the Target and the Origin;
- * MAY carry one or more Metric Container Options that contains the aggregated routing metrics values for the route specified in P2P-RDO;
- * MAY carry one Data Option to carry any time-critical application data to the Origin, subject to the following conditions: if a Target chooses to include a Data Option inside a DRO,
 - + it MUST include the same Data Option in all retransmissions of this DRO message and
 - + it MUST NOT include a different Data Option in any other DRO messages it generates for this route discovery.

The Target MAY repeat the same Data Option in multiple DRO messages it generates for a particular route discovery.

A received DRO message MUST be discarded if it does not contain exactly one P2P-RDO or if it contains multiple Data Options.

8.1. Secure DRO

A Secure DRO message follows the format in Figure 7 of [RFC6550], where the base format is the base DRO shown in Figure 3.

8.2. Setting a P2P-RDO Carried in a Discovery Reply Object

A Discovery Reply Object MUST carry one (and only one) P2P-RDO, which MUST be set as defined in Section 7.1. Specifically, the following fields MUST be set as specified next:

- o Reply (R): This flag MUST be set to zero on transmission and ignored on reception.
- o Hop-by-Hop (H): The H flag in the P2P-RDO included in a DRO message MUST have the same value as the H flag in the P2P-RDO inside the corresponding DIO message.
- o Number of Routes (N): This field MUST be set to zero on transmission and ignored on reception.

- o Life Time (L): This field MUST be set to zero on transmission and ignored on reception.
- o MaxRank/NH: This field indicates the index of the next hop address in the Address vector. When a Target generates a DRO message, the NH field is set to $n = (\text{Option Length} - 2 - (16 - \text{Compr})) / (16 - \text{Compr})$.
- o TargetAddr: This field MUST contain a unicast IPv6 address of the Target generating the DRO.
- o Address[1..n]: The Address vector MUST contain a complete route between the Origin and the Target such that the first element in the vector contains the IPv6 address of the router next to the Origin and the last element contains the IPv6 address of the router next to the Target.

9. P2P-RPL Route Discovery By Creating a Temporary DAG

This section details the P2P-RPL route discovery operation.

9.1. Joining a Temporary DAG

All the routers participating in a P2P-RPL route discovery, including the Origin and the Target(s), MUST join the temporary DAG being created for the purpose. When a router joins a temporary DAG advertized by a P2P mode DIO, it MUST maintain its membership in the temporary DAG for the Life Time duration listed in the P2P-RDO. The only purpose of a temporary DAG's existence is to facilitate the P2P-RPL route discovery process. The temporary DAG MUST NOT be used to route packets. A router MUST detach from the temporary DAG once the duration of its membership in the DAG has reached the DAG's life time. After receiving a DRO with the Stop flag set to one, a router SHOULD NOT send or receive any more DIOs for this temporary DAG and SHOULD also cancel any pending DIO transmission.

9.2. Trickle Operation For P2P Mode DIOs

An RPL router uses a Trickle timer [RFC6206] to control DIO transmissions. The Trickle control of DIO transmissions provides quick resolution of any "inconsistency" while avoiding redundant DIO transmissions. The Trickle algorithm also imparts protection against loss of DIOs due to inherent lack of reliability in LLNs. When controlling the transmissions of a P2P mode DIO, a Trickle timer SHOULD follow the following rules:

- o The receipt of a P2P mode DIO, that allows the router to advertise a better route (in terms of the routing metrics and the OF in use) than before, is considered "inconsistent" and hence resets the Trickle timer. Note that the first receipt of a P2P mode DIO advertising a particular temporary DAG is always considered an "inconsistent" event.
- o The receipt of a P2P mode DIO from a parent in the temporary DAG is considered neither "consistent" nor "inconsistent" if it does not allow the router to advertise a better route than before. Thus, the receipt of such DIOs has no impact on the Trickle operation. Note that this document does not impose any requirements on how a router might choose its parents in the temporary DAG.
- o The receipt of a P2P mode DIO is considered "consistent" if the source of the DIO is not a parent in the temporary DAG and either of the following conditions is true:
 - * The DIO advertises a better route than the router but does not allow the router to advertise a better route itself; or
 - * The DIO advertises a route as good as the route (to be) advertised by the router.

Note that the Trickle algorithm's DIO suppression rules are in effect at all times. Hence, a P2P-RPL router may suppress a DIO transmission even if it has not made any DIO transmission yet.

- o The receipt of a P2P mode DIO, that advertises a worse route than what the router advertises (or would advertise when it gets a chance to generate its DIO), is considered neither "consistent" nor "inconsistent", i.e., the receipt of such a DIO has no impact on the Trickle operation.
- o The Imin parameter SHOULD be set taking in account the connectivity within the network. For highly connected networks, a small Imin value (of the order of the typical transmission delay for a DIO) may lead to congestion in the network as a large number of routers reset their Trickle timers in response to the first receipt of a DIO from the Origin. These routers would generate their DIOs within Imin interval and cause additional routers to reset their trickle timers and generate more DIOs. Thus, for highly connected networks, the Imin parameter SHOULD be set to a value at least one order of magnitude larger than the typical transmission delay for a DIO. For sparsely connected networks, the Imin parameter can be set to a value that is a small multiple of the typical transmission delay for a DIO. Note that the Imin

value has a direct impact on the time required for a P2P-RPL route discovery to complete. In general, the time required for a P2P-RPL route discovery would increase approximately linearly with the value of the Imin parameter. Since the route discovery must complete within the lifetime of the temporary DAG created for the purpose, the Origin should set this lifetime to a large enough value taking in account the Imin value as well as the expected distance (in terms of hops and/or ETX) to the Target(s).

- o The Imax parameter SHOULD be set to a large value (several orders of magnitude higher than the Imin value) and is unlikely to be critical for P2P-RPL operation. This is because the first receipt of a P2P mode DIO for a particular temporary DAG is considered an inconsistent event and would lead to resetting of Trickle timer duration to the Imin value. Given the temporary nature of the DAGs used in P2P-RPL, Trickle timer may not get a chance to increase much.
- o The recommended value of redundancy constant "k" is 1. With this value of "k", a DIO transmission will be suppressed if the router receives even a single "consistent" DIO during a timer interval. This setting for the redundancy constant is designed to reduce the number of messages generated during a route discovery process and is suitable for the environments with low or moderate packet loss rates. However, this setting may result in an increase in the time required for the route discovery process to complete. A higher value for the redundancy constant may be more suitable in
 - * Environments with high packet loss rates; or
 - * Deployments where the time required for the route discovery process to complete needs to be as small as possible; or
 - * Deployments where specific destinations are reachable only through specific intermediate routers (and hence these intermediate routers should not suppress their DIOs).

A particular deployment should take in account the above mentioned factors when deciding the value of the redundancy constant.

Individual P2P-RPL deployments are encouraged to share their experience with these rules with ROLL working group to help guide the development of standards track version of the protocol. Applicability Statements that specify the use of P2P-RPL MUST provide guidance for setting Trickle parameters, particularly Imin and the redundancy constant.

9.3. Processing a P2P Mode DIO

The rules for DIO processing and transmission, described in Section 8 of RPL [RFC6550], apply to P2P mode DIOs as well except as modified in this document. In particular, in accordance with Section 8.2.3 of RPL [RFC6550], a received P2P mode DIO MUST be discarded if it is malformed according to the rules specified in this document and in [RFC6550].

The following rules for processing a received P2P mode DIO apply to both Intermediate Routers and the Target.

A router SHOULD discard a received P2P mode DIO with no further processing if it does not have bidirectional reachability with the neighbor that generated the received DIO. Note that bidirectional reachability does not mean that the link must have the same values for a routing metric in both directions. A router SHOULD calculate the values of the link-level routing metrics included in the received DIO taking in account the metric's value in both Forward and Reverse directions. Bidirectional reachability along a discovered route allows the Target to use this route to reach the Origin. In particular, the DRO messages travel from the Target to the Origin along a discovered route.

A router MUST discard a received P2P mode DIO with no further processing:

- o If the DIO advertises INFINITE_RANK as defined in Section 17 of [RFC6550].
- o If the integer part of the rank advertised in the DIO equals or exceeds the MaxRank limit listed in the P2P Route Discovery Option.
- o If the routing metric values do not satisfy one or more of the mandatory route constraints listed in the DIO or if the router cannot evaluate the mandatory route constraints, e.g., if the router does not support the metrics used in the constraints.
- o If the router previously received a DRO message with the same RPLInstanceID and DODAGID as the received DIO and with the Stop flag set to one.

The router MUST check the Target addresses listed in the P2P-RDO and any RPL Target Options included in the received DIO. If one of its IPv6 addresses is listed as a Target address or if it belongs to the multicast group specified as one of the Target addresses, the router considers itself a Target and processes the received DIO as specified

in Section 9.5. Otherwise, the router considers itself an Intermediate Router and processes the received DIO as specified in Section 9.4.

9.4. Additional Processing of a P2P Mode DIO At An Intermediate Router

An Intermediate Router **MUST** discard a received P2P mode DIO with no further processing if the router cannot elide Compr (as specified in the P2P-RDO) prefix octets from its IPv6 address or if adding its IPv6 address to the Address vector (inside the P2P-RDO) would result in the Address vector containing multiple, non-back-to-back addresses belonging to this router.

On receiving a P2P mode DIO, an Intermediate Router **MUST** do the following:

- o The router **MUST** determine whether this DIO advertises a better route than the router itself and whether the receipt of the DIO would allow the router to advertise a better route than before. Accordingly, the router **SHOULD** consider this DIO as consistent/inconsistent from Trickle perspective as described in Section 9.2. Note that the route comparison in a P2P-RPL route discovery is performed using the parent selection rules of the OF in use as specified in Section 14 of RPL [RFC6550]. If the received DIO would allow the router to advertise a better route, the router **MUST** remember the route advertised (inside the P2P-RDO) in the DIO (after adding its own IPv6 address to the route) for inclusion in its future DIOs. When an Intermediate Router adds itself to a route, it **MUST** ensure that the IPv6 address added to the route does not belong to an ingress-only or an egress-only interface. To improve the diversity of the routes being discovered, an Intermediate Router **SHOULD** keep track of multiple routes (as long as all these routes are the best seen so far), one of which **SHOULD** be selected in a uniform random manner for inclusion in the P2P-RDO inside the router's next DIO. Note that the route accumulation in a P2P mode DIO **MUST** take place even if the Origin does not want any DRO messages to be generated (i.e., the R flag inside the P2P-RDO is set to zero). This is because the Target may still be able to use the accumulated route as a source route to reach the Origin.
- o The router **MUST** copy any Data Option (to be included in its future DIO transmissions) if the received DIO comes from a parent and is the first parent-originated DIO received with a Data Option inside.

9.5. Additional Processing of a P2P Mode DIO At The Target

The Target MUST determine if the received DIO contains a Data Option and deliver the data to the specified upper layer protocol unless it has already done so in response to a previously received DIO. If this route discovery involves multiple Targets, the Target MUST remember this Data Option for inclusion in its own DIOs.

The Target MAY store the route contained in the P2P-RDO in the received DIO for use as a Source Route to reach the Origin. The lifetime of this Source Route is specified by the Default Lifetime and Lifetime Unit parameters inside the DODAG Configuration Option currently in effect. This lifetime can be extended (or shortened) appropriately following a hint from an upper-layer protocol.

If the Reply flag inside the P2P-RDO in the received DIO is zero, the Target MUST discard the received DIO with no further processing. Otherwise, the Target MAY select the route contained in the P2P-RDO to send a DRO message back to the Origin. If the H flag inside the P2P-RDO is one, the Target needs to select one route and send a DRO message along this route back to the Origin. If the H flag is zero, the number of routes to be selected (and the number of DRO messages to be sent back) is given by one plus the value of the N field in the P2P-RDO. This document does not prescribe a particular method for the Target to select the routes. Example methods include selecting each route that meets the specified routing constraints until the desired number have been selected or selecting the best routes discovered over a certain time period. If multiple routes are to be selected, the Target SHOULD avoid selecting routes that have large segments in common.

If the Target selects the route contained in the P2P-RDO in the received DIO, it sends a DRO message back to the Origin (identified by the DODAGID field in the DIO). The DRO message MUST include a P2P-RDO that contains the selected route inside the Address vector. Various fields inside the P2P-RDO MUST be set as specified in Section 8.2. The Target MAY set the A flag inside the DRO message to one if it desires the Origin to send back a DRO-ACK message on receiving the DRO. In this case, the Target waits for DRO_ACK_WAIT_TIME duration for the DRO-ACK message to arrive. Failure to receive the DRO-ACK message within this time duration causes the Target to retransmit the DRO message. The Target MAY retransmit the DRO message in this fashion up to MAX_DRO_RETRANSMISSIONS times. Both DRO_ACK_WAIT_TIME and MAX_DRO_RETRANSMISSIONS are configurable parameters to be decided based on the characteristics of individual deployments. Note that all DRO transmissions and retransmissions MUST take place while the Target is still a part of the temporary DAG created for the route

discovery. A Target MUST NOT transmit a DRO if it no longer belongs to this DAG.

The Target MAY set the Stop flag inside the DRO message to one if

- o this router is the only Target specified in the corresponding DIO, i.e., the corresponding DIO specified a unicast address of the router as the TargetAddr inside the P2P-RDO with no additional Targets specified via RPL Target Options; and
- o the Target has already selected the desired number of routes.

The Target MAY include a Metric Container Option in the DRO message. This Metric Container contains the end-to-end routing metric values for the route specified in the P2P-RDO. The Target MAY include one Data Option in the DRO message to carry time-critical application data for the Origin, subject to the conditions listed in Section 8. The Target MUST transmit the DRO message via a link-local multicast.

A Target MUST NOT forward a P2P mode DIO any further if no other Targets are to be discovered, i.e., if a unicast IPv6 address (of this Target) is specified as the TargetAddr inside the P2P-RDO and no additional Targets are specified via RPL Target Options inside the DIOs for this route discovery. Otherwise, the Target MUST generate DIOs for this route discovery as an Intermediate Router would.

9.6. Processing a DRO At An Intermediate Router

If the DODAGID field in the received DRO does not list a router's own IPv6 address, the router considers itself an Intermediate Router and MUST process the received message in the following manner:

- o The router MUST discard the received DRO with no further processing if it does not belong to the temporary DAG identified by the RPLInstanceID and the DODAGID fields in the DRO.
- o If the Stop flag inside the received DRO is set to one, the router SHOULD NOT send or receive any more DIOs for this temporary DAG and SHOULD cancel any pending DIO transmission.
- o The router MUST ignore any Metric Container and Data Options contained in the DRO message.
- o If Address[NH] element inside the P2P-RDO lists the router's own unicast IPv6 address, the router is a part of the route carried in the P2P-RDO. In this case, the router MUST do the following:

- * To prevent loops, the router MUST discard the DRO message with no further processing if the Address vector in the P2P-RDO includes multiple IPv6 addresses assigned to the router's interfaces.
- * If the H flag inside the P2P-RDO is one, the router MUST store the state for the Forward Hop-by-hop route carried inside the P2P-RDO. This state consists of:
 - + The RPLInstanceID and the DODAGID fields of the DRO.
 - + The route's destination, the Target (identified by TargetAddr field inside P2P-RDO).
 - + The IPv6 address of the next hop, Address[NH+1] (unless NH value equals the number of elements in the Address vector, in which case the Target itself is the next hop).

This Hop-by-hop routing state MUST expire at the end of the lifetime specified by the Default Lifetime and Lifetime Unit parameters inside the DODAG Configuration Option used in P2P mode DIOs for this route discovery.

- * If the router already maintains a Hop-by-hop state listing the Target as the destination and carrying same RPLInstanceID and DODAGID fields as the received DRO and the next hop information in the state does not match the next hop indicated in the received DRO, the router MUST discard the DRO message with no further processing. Note that this situation would occur in the following two cases:
 - + When the route listed in the Address vector inside the P2P-RDO contains a previously undetected loop. In this case, the rule above causes the DRO messages to be discarded.
 - + When a Hop-by-hop Route between the Origin and the Target, previously established using the same RPLInstanceID and DODAGID as the route currently being established, still exists and at least partially overlaps the route currently being established.
- * The router MUST decrement the NH field inside the P2P-RDO and send the DRO message further via link-local multicast.

9.7. Processing a DRO At The Origin

When a router receives a DRO message that lists its IPv6 address in the DODAGID field, the router recognizes itself as the Origin for the corresponding P2P-RPL route discovery, notes the Target that originated this message (from the TargetAddr field inside the P2P-RDO) and processes the message in the following manner:

- o The Origin MUST discard the received DRO with no further processing if it no longer belongs to the temporary DAG identified by the RPLInstanceID and the DODAGID fields in the DRO.
- o If the received DRO contains a Data Option and if it has not already done so following the receipt of an earlier DRO from this Target, the Origin MUST deliver the data inside the Data Option to the specified upper layer protocol.
- o If the Stop flag inside the received DRO is set to one, the Origin SHOULD NOT generate any more DIOs for this temporary DAG and SHOULD cancel any pending DIO transmission.
- o If the P2P-RDO inside the DRO has the H flag set to 0, the Address vector inside the P2P-RDO contains a Source Route to this Target and the Origin MUST store this Source Route in its memory. The lifetime of this Source Route is specified by the Default Lifetime and Lifetime Unit parameters inside the DODAG Configuration Option in the P2P mode DIOs used for this route discovery. This lifetime could be extended (or shortened) appropriately following a hint from an upper-layer protocol.
- o If the P2P-RDO inside the DRO has the H flag set to 1, the DRO message is establishing a Hop-by-hop Route to this Target and the Origin MUST store in its memory the state for this Hop-by-hop Route in the manner described in Section 9.6. This Hop-by-hop routing state MUST expire at the end of the lifetime specified by the Default Lifetime and Lifetime Unit parameters inside the DODAG Configuration Option used in P2P mode DIOs for this route discovery. The standards track version of P2P-RPL may consider specifying a signaling mechanism that will allow the Origin to extend (or shorten) the lifetime of a P2P-RPL Hop-by-hop Route following a suitable hint from an upper-layer protocol.
- o If the received DRO message contains one or more Metric Container Options, the Origin MAY store the values of the routing metrics associated with the discovered route in its memory. This information may be useful in formulating the constraints for any future P2P-RPL route discovery to this Target.

- o If the A flag is set to one in the received DRO message, the Origin MUST generate a DRO-ACK message as described in Section 10 and unicast the message to the Target. The Origin MAY use the route just discovered to send the DRO-ACK message to the Target. Section 11 describes how a packet may be forwarded along a Source/Hop-by-hop Route discovered using P2P-RPL.

10. The Discovery Reply Object Acknowledgement (DRO-ACK)

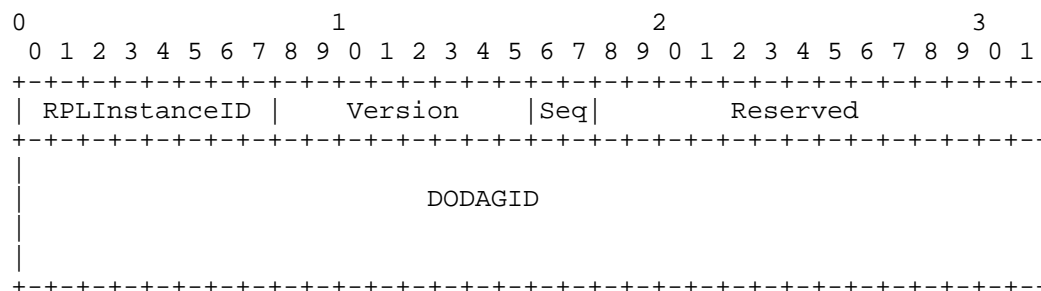


Figure 4: Format of the base Discovery Reply Object Acknowledgement (DRO-ACK)

A DRO message may fail to reach the Origin due to a number of reasons. Unlike the DIO messages that benefit from Trickle-controlled retransmissions, the DRO messages are prone to loss due to unreliable packet transmission in LLNs. Since a DRO message travels via link-local multicast, it cannot use link-level acknowledgements to improve the reliability of its transmission. Also, an Intermediate Router may drop the DRO message (e.g., because of its inability to store the state for the Hop-by-hop Route the DRO is establishing). To protect against the potential failure of a DRO message to reach the Origin, the Target MAY request the Origin to send back a DRO Acknowledgement (DRO-ACK) message on receiving a DRO message. Failure to receive such an acknowledgement within the DRO_ACK_WAIT_TIME interval of sending the DRO message forces the Target to resend the message.

This section defines two new RPL Control Message types: DRO Acknowledgement (DRO-ACK; with code TBD6) and Secure DRO-ACK (with code TBD7). A DRO-ACK message MUST travel as a unicast message from the Origin to the Target. The format of a base DRO-ACK message is shown in Figure 4. Various fields in a DRO-ACK message MUST have the same values as the corresponding fields in the DRO message. The field marked as "Reserved" MUST be set to zero on transmission and MUST be ignored on reception. A Secure DRO-ACK message follows the

format in Figure 7 of [RFC6550], where the base format is same as the base DRO-ACK shown in Figure 4.

11. Packet Forwarding Along a Route Discovered Using P2P-RPL

An Origin MAY use a Source Routing Header (SRH) [RFC6554] to send a packet along a Source Route discovered using P2P-RPL.

Travel along a Hop-by-hop Route, established using P2P-RPL, requires specifying the RPLInstanceID and the DODAGID (of the temporary DAG used for the route discovery) to identify the route. This is because a P2P-RPL route discovery does not use globally unique RPLInstanceID values and hence both the RPLInstanceID (a local value assigned by the Origin) and the DODAGID (an IPv6 address of the Origin) are required to uniquely identify a P2P-RPL Hop-by-hop Route to a particular destination.

An Origin MAY include an RPL option [RFC6553] inside the IPv6 hop-by-hop options header of a packet to send it along a Hop-by-hop Route established using P2P-RPL. For this purpose, the Origin MUST set the DODAGID of the temporary DAG used for the route discovery as the source IPv6 address of the packet. Further, the Origin MUST specify inside the RPL option the RPLInstanceID of the temporary DAG used for the route discovery and set the O flag inside the RPL option to one. On receiving this packet, an Intermediate Router checks the O flag and correctly infer the source IPv6 address of the packet as the DODAGID of the Hop-by-hop Route. The router then uses the DODAGID, the RPLInstanceID and the destination address to identify the routing state to be used to forward the packet further.

12. Interoperability with Core RPL

This section describes how RPL routers that implement P2P-RPL interact with RPL routers that do not. In general, P2P-RPL operation does not affect core RPL operation and vice versa. However, core RPL does allow a router to join a DAG as a leaf node even if it does not understand the Mode of Operation (MOP) used in the DAG. Thus, an RPL router that does not implement P2P-RPL may conceivably join a temporary DAG being created for a P2P-RPL route discovery as a leaf node and maintain its membership even though the DAG no longer exists. This may impose a drain on the router's memory. However, such RPL-only leaf nodes do not interfere with P2P-RPL route discovery since a leaf node may only generate a DIO advertising an INFINITE_RANK and all routers implementing P2P-RPL are required to discard such DIOs. Note that core RPL does not require a router to join a DAG whose MOP it does not understand. Moreover, RPL routers

in a particular deployment may have strict restrictions on the DAGs they may join, thereby mitigating the problem.

The P2P-RPL mechanism described in this document works best when all the RPL routers in the LLN implement P2P-RPL. In general, the ability to discover routes as well as the quality of discovered routes would deteriorate with the fraction of RPL routers that implement P2P-RPL.

13. Security Considerations

A P2P-RPL deployment may be susceptible to denial of service attacks by rogue routers that initiate fake route discoveries. A rogue router could join a temporary DAG and advertise false information in its DIOs in order to include itself in the discovered route(s). It could generate bogus DRO messages carrying bad routes or maliciously modify genuine DRO messages it receives.

In general, the security considerations for the operation of P2P-RPL are similar to the ones for the operation of RPL (as described in Section 19 of [RFC6550]). Section 10 of RPL specification [RFC6550] describes a variety of security mechanisms that provide data confidentiality, authentication, replay protection and delay protection services. Each RPL control message has a secure version that allows the specification of the level of security and the algorithms used to secure the message. The mechanism defined in this document is based on the use of DIOs to form a temporary DAG and discover P2P routes. These DIOs can be used in their secure versions if desired. New RPL control messages defined in this document (DRO and DRO-ACK) have secure versions as well. In addition, a P2P-RPL deployment may use the security features provided by the link layer in use. Thus, a particular P2P-RPL deployment can analyze its security requirements and use the appropriate set of RPL (or link layer) security mechanisms that meet those requirements. Note that the contents of the Data Option, if used, has the same level of security as the DIO/DRO message it is part of. Hence, a P2P-RPL deployment SHOULD take in consideration the security requirements of the data being sent inside the Data Options when deciding the overall security requirements.

Since a DRO message travels along a Source Route specified inside the message, some of the security concerns that led to the deprecation of Type 0 routing header [RFC5095] may apply. To avoid the possibility of a DRO message traveling in a routing loop, this document requires each Intermediate Router to confirm that the Source Route listed inside the message does not contain any routing loop involving itself before the router could forward the message further. As specified in

Section 9.6, this check involves the router making sure that its IPv6 addresses do not appear multiple times inside the Source Route with one or more other IPv6 addresses in between.

14. IANA Considerations

14.1. Additions to Mode of Operation

This document defines a new Mode of Operation, entitled "P2P Route Discovery Mode" (see Section 6), assigned a value TBD1 from the "Mode of Operation" space [to be removed upon publication: <http://www.iana.org/assignments/rpl/rpl.xml#mop>] [RFC6550]. IANA is requested to allocate a suitable value to TBD1. The string TBD1 in this document should be replaced by the allocated value. The previous two sentences should be removed before publication.

Value	Description	Reference
TBD1	P2P Route Discovery Mode of Operation	This document

Mode of Operation

14.2. Additions to RPL Control Message Options

This document defines two new RPL options:

- o "P2P Route Discovery Option" (see Section 7.1), assigned a value TBD2 from the "RPL Control Message Options" space [to be removed upon publication: <http://www.iana.org/assignments/rpl/rpl.xml#control-message-options>] [RFC6550]. IANA is requested to allocate a suitable value to TBD2. The string TBD2 in this document should be replaced by the allocated value. The previous two sentences should be removed before publication.
- o "Data Option" (see Section 7.2), assigned a value TBD3 from the "RPL Control Message Options" space [to be removed upon publication: <http://www.iana.org/assignments/rpl/rpl.xml#control-message-options>] [RFC6550]. IANA is requested to allocate a suitable value to TBD3. The string TBD3 in this document should be replaced by the allocated value. The previous two sentences should be removed before publication.

Value	Meaning	Reference
TBD2	P2P Route Discovery	This document
TBD3	Data	This document

RPL Control Message Options

14.3. Additions to RPL Control Codes

This document defines the following new RPL messages:

- o "Discovery Reply Object" (see Section 8), assigned a value TBD4 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 TBD4 from the range 0x00-0x7F to indicate a message without security enabled. The string TBD4 in this document should be replaced by the allocated value. The previous two sentences should be removed before publication.
- o "Secure Discovery Reply Object" (see Section 8.1), assigned a value TBD5 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 TBD5 from the range 0x80-0xFF to indicate a message with security enabled. The string TBD5 in this document should be replaced by the allocated value. The previous two sentences should be removed before publication.
- o "Discovery Reply Object Acknowledgement" (see Section 10), assigned a value TBD6 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 TBD6 from the range 0x00-0x7F to indicate a message without security enabled. The string TBD6 in this document should be replaced by the allocated value. The previous two sentences should be removed before publication.
- o "Secure Discovery Reply Object Acknowledgement" (see Section 10), assigned a value TBD7 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 TBD7 from the range 0x80-0xFF to indicate a message with security enabled. The string

TBD7 in this document should be replaced by the allocated value.
The previous two sentences should be removed before publication.

Code	Description	Reference
TBD4	Discovery Reply Object	This document
TBD5	Secure Discovery Reply Object	This document
TBD6	Discovery Reply Object Acknowledgement	This document
TBD7	Secure Discovery Reply Object Acknowledgement	This document

RPL Control Codes

15. Acknowledgements

Authors gratefully acknowledge the contributions of the following individuals (in alphabetical order) in the development of this document: Dominique Barthel, Jakob Buron, Cedric Chauvenet, Thomas Clausen, Robert Cragie, Ted Humpal, Richard Kelsey, Phil Levis, Charles Perkins, Joseph Reddy, Michael Richardson, Zach Shelby, Pascal Thubert, Hristo Valev and JP Vasseur.

16. References

16.1. Normative References

- [PROTOCOL] "Protocol Numbers", <<http://www.iana.org/assignments/protocol-numbers/protocol-numbers.xml>>.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 5226, May 2008.
- [RFC6206] Levis, P., Clausen, T., Hui, J., Gnawali, O., and J. Ko, "The Trickle Algorithm", RFC 6206, March 2011.
- [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.

16.2. Informative References

- [I-D.ietf-roll-p2p-measurement]
Goyal, M., Baccelli, E., Brandt, A., and J. Martocci, "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 (work in progress), January 2013.
- [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.
- [RFC6552] Thubert, P., "Objective Function Zero for the Routing Protocol for Low-Power and Lossy Networks (RPL)", RFC 6552, March 2012.
- [RFC6553] Hui, J. and JP. Vasseur, "The Routing Protocol for Low-Power and Lossy Networks (RPL) Option for Carrying RPL Information in Data-Plane Datagrams", RFC 6553, 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.

Authors' Addresses

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

Matthias Philipp
INRIA

Phone: +33-169-335-511
Email: Matthias.Philipp@inria.fr

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 St
Milwaukee, WI 53202
USA

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

Networking Working Group
Internet-Draft
Intended status: Informational
Expires: August 29, 2013

T. Tsao
R. Alexander
Cooper Power Systems
M. Dohler
CTTC
V. Daza
A. Lozano
Universitat Pompeu Fabra
February 25, 2013

A Security Threat Analysis for Routing over Low-Power and Lossy Networks
draft-ietf-roll-security-threats-01

Abstract

This document presents a security threat analysis for routing over low-power and lossy networks (LLN). The development builds upon previous work on routing security and adapts the assessments to the issues and constraints specific to low-power and lossy networks. A systematic approach is used in defining and evaluating the security threats. Applicable countermeasures are application specific and are addressed in relevant applicability statements. These assessments provide the basis of the security recommendations for incorporation into low-power, lossy network routing protocols.

Requirements Language

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 RFC 2119 [RFC2119].

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

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

This Internet-Draft will expire on August 29, 2013.

Copyright Notice

Copyright (c) 2013 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	5
2. Terminology	5
3. Considerations on ROLL Security	5
3.1. Routing Assets and Points of Access	6
3.2. The CIA Security Reference Model	9
3.3. Issues Specific to or Amplified in LLNs	10
3.4. ROLL Security Objectives	12
4. Threats and Attacks	13
4.1. Threats and Attacks on Confidentiality	14
4.1.1. Routing Exchange Exposure	14
4.1.2. Routing Information (Routes and Network Topology) Exposure	15
4.2. Threats and Attacks on Integrity	15
4.2.1. Routing Information Manipulation	15
4.2.2. Node Identity Misappropriation	16
4.3. Threats and Attacks on Availability	16
4.3.1. Routing Exchange Interference or Disruption	17
4.3.2. Network Traffic Forwarding Disruption	17
4.3.3. Communications Resource Disruption	18
4.3.4. Node Resource Exhaustion	19
5. Countermeasures	19
5.1. Confidentiality Attack Countermeasures	20
5.1.1. Countering Deliberate Exposure Attacks	20
5.1.2. Countering Sniffing Attacks	20
5.1.3. Countering Traffic Analysis	21
5.1.4. Countering Physical Device Compromise	22
5.1.5. Countering Remote Device Access Attacks	24
5.2. Integrity Attack Countermeasures	25
5.2.1. Countering Unauthorized Modification Attacks	25
5.2.2. Countering Overclaiming and Misclaiming Attacks	25
5.2.3. Countering Identity (including Sybil) Attacks	26
5.2.4. Countering Routing Information Replay Attacks	26
5.2.5. Countering Byzantine Routing Information Attacks	26
5.3. Availability Attack Countermeasures	27
5.3.1. Countering HELLO Flood Attacks and ACK Spoofing Attacks	28
5.3.2. Countering Overload Attacks	29
5.3.3. Countering Selective Forwarding Attacks	30
5.3.4. Countering Sinkhole Attacks	31
5.3.5. Countering Wormhole Attacks	32
6. ROLL Security Features	32
6.1. Confidentiality Features	33
6.2. Integrity Features	34
6.3. Availability Features	35
6.4. Security Key Management	36
6.5. Consideration on Matching Application Domain Needs	37

6.5.1. Security Architecture	38
6.5.2. Mechanisms and Operations	40
7. IANA Considerations	42
8. Security Considerations	42
9. Acknowledgments	43
10. References	43
10.1. Normative References	43
10.2. Informative References	43
Authors' Addresses	46

1. Introduction

In recent times, networked electronic devices have found an increasing number of applications in various fields. Yet, for reasons ranging from operational application to economics, these wired and wireless devices are often supplied with minimum physical resources; the constraints include those on computational resources (RAM, clock speed, storage), communication resources (duty cycle, packet size, etc.), but also form factors that may rule out user access interfaces (e.g., the housing of a small stick-on switch), or simply safety considerations (e.g., with gas meters). As a consequence, the resulting networks are more prone to loss of traffic and other vulnerabilities. The proliferation of these low-power and lossy networks (LLNs), however, are drawing efforts to examine and address their potential networking challenges. Securing the establishment and maintenance of network connectivity among these deployed devices becomes one of these key challenges.

This document presents a threat analysis for securing Routing Over LLNs (ROLL) through an analysis that starts from the routing basics. The objective is two-fold. First, the analysis will be used to identify pertinent security issues. Second, it will facilitate both the assessment of a protocol's security threats and the identification of the necessary features for development of secure protocols for the ROLL Working Group.

The approach adopted in this effort proceeds in four steps, to examine security issues in ROLL, to analyze threats and attacks, to consider the countermeasures, and then to make recommendations for securing ROLL. The basis is found on identifying the assets and points of access of routing and evaluating their security needs based on the Confidentiality, Integrity, and Availability (CIA) model in the context of LLN.

2. Terminology

This document adopts the terminology defined in [RFC6550] and in [RFC4949], with the following addition:

Node An element of a low-power, lossy network that may be a router or a host.

3. Considerations on ROLL Security

Security, in essence, entails implementing measures to ensure controlled state changes on devices and network elements, both based

on external inputs (received via communications) or internal inputs (physical security of device itself and parameters maintained by the device, including, e.g., clock). State changes would thereby involve not only proper authorization for actions, authentication, and potentially integrity and confidentiality, but also proper order of state changes through timeliness, since seriously delayed state changes, such as commands or updates of routing tables, may negatively impact system operation. A security assessment can therefore begin with a focus on the assets or elements of information that may be the target of the state changes and the access points in terms of interfaces and protocol exchanges through which such changes may occur. In the case of routing security the focus is directed towards the elements associated with the establishment and maintenance of network connectivity.

This section sets the stage for the development of the analysis by applying the systematic approach proposed in [Myagmar2005] to the routing security problem, while also drawing references from other reviews and assessments found in the literature, particularly, [RFC4593] and [Karlof2003]; thus, the work presented herein may find use beyond routing for LLNs. The subsequent subsections begin with a focus on the elements of a generic routing process that is used to establish routing assets and points of access to the routing functionality. Next, the CIA security model is briefly described. Then, consideration is given to issues specific to or amplified in LLNs. This section concludes with the formulation of a set of security objectives for ROLL.

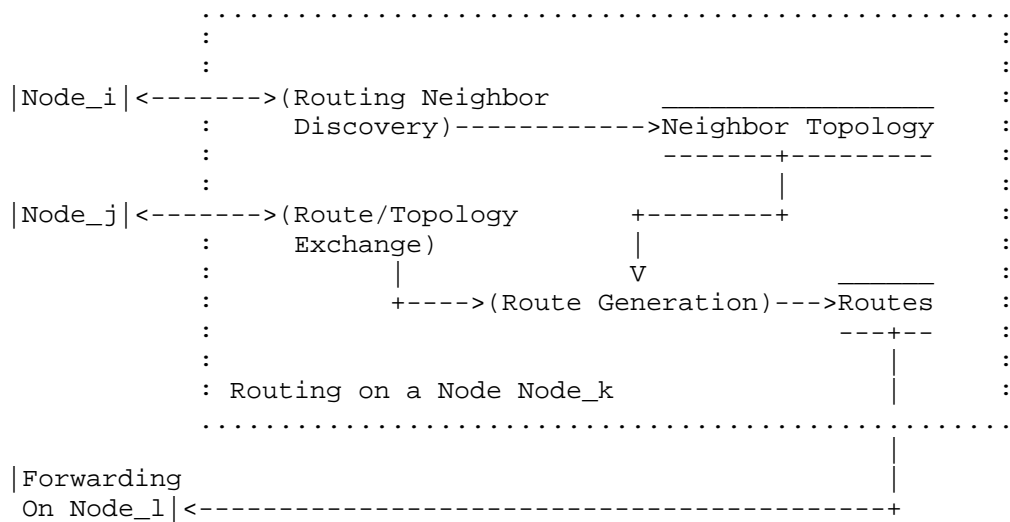
3.1. Routing Assets and Points of Access

An asset implies an important system component (including information, process, or physical resource), the access to, corruption or loss of which adversely affects the system. In the control plane context, an asset is information about the network, processes used to manage and manipulate this data, and the physical devices on which this data is stored and manipulated. The corruption or loss of these assets may adversely impact the control plane of the network. Within the same context, a point of access is an interface or protocol that facilitates interaction between control plane components. Identifying these assets and points of access will provide a basis for enumerating the attack surface of the control plane.

A level-0 data flow diagram [Yourdon1979] is used here to identify the assets and points of access within a generic routing process. The use of a data flow diagram allows for a clear and concise model of the way in which routing nodes interact and process information, and hence provides a context for threats and attacks. The goal of

the model is to be as detailed as possible so that corresponding components and mechanisms in an individual routing protocol can be readily identified, but also to be as general as possible to maximize the relevancy of this effort for the various existing and future protocols. Nevertheless, there may be discrepancies, likely in the form of additional elements, when the model is applied to some protocols. For such cases, the analysis approach laid out in this document should still provide a valid and illustrative path for their security assessment.

Figure 1 shows that nodes participating in the routing process transmit messages to discover neighbors and to exchange routing information; routes are then generated and stored, which may be maintained in the form of the protocol forwarding table. The nodes use the derived routes for making forwarding decisions.



Notation:

(Proc) A process Proc

DataBase A data storage DataBase

|Node_n| An external entity Node_n

-----> Data flow

Figure 1: Data Flow Diagram of a Generic Routing Process

It is seen from Figure 1 that

o Assets include

- * routing and/or topology information;
- * communication channel resources (bandwidth);
- * node resources (computing capacity, memory, and remaining energy);
- * node identifiers (including node identity and ascribed attributes such as relative or absolute node location).

- o Points of access include
 - * neighbor discovery;
 - * route/topology exchange;
 - * node physical interfaces (including access to data storage).

A focus on the above list of assets and points of access enables a more directed assessment of routing security; for example, it is readily understood that some routing attacks are in the form of attempts to misrepresent routing topology. Indeed, the intention of the security threat analysis is to be comprehensive. Hence, some of the discussion which follows is associated with assets and points of access that are not directly related to routing protocol design but nonetheless provided for reference since they do have direct consequences on the security of routing.

3.2. The CIA Security Reference Model

At the conceptual level, security within an information system in general and applied to ROLL in particular is concerned with the primary issues of confidentiality, integrity, and availability. In the context of ROLL:

Confidentiality

Confidentiality involves the protection of routing information as well as routing neighbor maintenance exchanges so that only authorized and intended network entities may view or access it. Because LLNs are most commonly found on a publicly accessible shared medium, e.g., air or wiring in a building, and sometimes formed ad hoc, confidentiality also extends to the neighbor state and database information within the routing device since the deployment of the network creates the potential for unauthorized access to the physical devices themselves.

Integrity

Integrity, as a security principle, entails the protection of routing information and routing neighbor maintenance exchanges, as well as derived information maintained in the database, from unauthorized modification or from misuse. Misuse, for example, may take the form of a delayed or inappropriately replayed message even where confidentiality protection is maintained. Hence, in addition to the data itself, integrity also concerns the authenticity of claimed identity of the origin and destination of a message and its timeliness or freshness. On the other hand, the access to and/or removal of data, execution of the routing process, and use of a device's computing and

energy resources, while relevant to routing security are considered larger system integrity issues [RFC4949] to be addressed beyond the routing protocol.

Availability

Availability ensures that routing information exchanges and forwarding services need to be available when they are required for the functioning of the serving network. Availability will apply to maintaining efficient and correct operation of routing and neighbor discovery exchanges (including needed information) and forwarding services so as not to impair or limit the network's central traffic flow function.

It is recognized that, besides those security issues captured in the CIA model, non-repudiation, that is, the assurance that the transmission and/or reception of a message cannot later be denied, may be a security requirement under certain circumstances. The service of non-repudiation applies after-the-fact and thus relies on the logging or other capture of on-going message exchanges and signatures. Applied to routing, non-repudiation will involve providing some ability to allow traceability or network management review of participants of the routing process including the ability to determine the events and actions leading to a particular routing state. As such, non-repudiation of routing may thus be more useful when interworking with networks of different ownerships. For the LLN application domains as described in [RFC5548], [RFC5673], [RFC5826], and [RFC5867], particularly with regard to routing security, proactive measures are much more critical than retrospective protections. Furthermore, given the significant practical limits to on-going routing transaction logging and storage and individual device signature authentication for each exchange, non-repudiation in the context of routing is not further considered as a ROLL security issue.

It should be emphasized here that for routing security the above CIA requirements must be complemented by the proper security policies and enforcement mechanisms to ensure that security objectives are met by a given routing protocol implementation.

3.3. Issues Specific to or Amplified in LLNs

The work [RFC5548], [RFC5673], [RFC5826], and [RFC5867] have identified specific issues and constraints of routing in LLNs for the urban, industrial, home automation, and building automation application domains, respectively. The following is a list of observations and evaluation of their impact on routing security considerations.

Limited energy, memory, and processing node resources

As a consequence of these constraints, there is an even more critical need than usual for a careful study of trade-offs on which and what level of security services are to be afforded during the system design process. In addition, the choices of security mechanisms are more stringent. Synchronization of security states with sleepy nodes is yet another issue.

Large scale of rolled out network

The possibly numerous nodes to be deployed, e.g., an urban deployment can see several hundreds of thousands of nodes, as well as the generally low level of expertise expected of the installers, make manual on-site configuration unlikely. Prolonged rollout and delayed addition of nodes, which may be from old inventory, over the lifetime of the network, also complicate the operations of key management.

Autonomous operations

Self-forming and self-organizing are commonly prescribed requirements of LLNs. In other words, a routing protocol designed for LLNs needs to contain elements of ad hoc networking and in most cases cannot rely on manual configuration for initialization or local filtering rules. Network topology/ownership changes, partitioning or merging, as well as node replacement, can all contribute to complicating the operations of key management.

Highly directional traffic

Some types of LLNs see a high percentage of their total traffic traverse between the nodes and the LLN Border Routers (LBRs) where the LLNs connect to non-LLNs. The special routing status of and the greater volume of traffic near the LBRs have routing security consequences. In fact, when Point-to-MultiPoint (P2MP) and MultiPoint-to-Point (MP2P) traffic represents a majority of the traffic, routing attacks consisting of advertising untruthfully preferred routes may cause serious damage.

Unattended locations and limited physical security

Many applications have the nodes deployed in unattended or remote locations; furthermore, the nodes themselves are often built with minimal physical protection. These constraints lower the barrier of accessing the data or security material stored on the nodes through physical means.

Support for mobility

On the one hand, only a number of applications require the support of mobile nodes, e.g., a home LLN that includes nodes on wearable health care devices or an industry LLN that includes nodes on cranes and vehicles. On the other hand, if a routing protocol is indeed used in such applications, it will clearly need to have corresponding security mechanisms.

Support for multicast and anycast

Support for multicast and anycast is called out chiefly for large-scale networks. Since application of these routing mechanisms in autonomous operations of many nodes is new, the consequence on security requires careful consideration.

The above list considers how an LLN's physical constraints, size, operations, and varieties of application areas may impact security. However, it is the combinations of these factors that particularly stress the security concerns. For instance, securing routing for a large number of autonomous devices that are left in unattended locations with limited physical security presents challenges that are not found in the common circumstance of administered networked routers. The following subsection sets up the security objectives for the routing protocol designed by the ROLL WG.

3.4. ROLL Security Objectives

This subsection applies the CIA model to the routing assets and access points, taking into account the LLN issues, to develop a set of ROLL security objectives.

Since the fundamental function of a routing protocol is to build routes for forwarding packets, it is essential to ensure that

- o routing/topology information is not tampered during transfer and in storage;
- o routing/topology information is not misappropriated;
- o routing/topology information is available when needed.

In conjunction, it is necessary to be assured of

- o the authenticity and legitimacy of the participants of the routing neighbor discovery process;
- o the routing/topology information received was faithfully generated according to the protocol design.

However, when trust cannot be fully vested through authentication of the principals alone, i.e., concerns of insider attack, assurance of the truthfulness and timeliness of the received routing/topology information is necessary. With regard to confidentiality, protecting the routing/topology information from eavesdropping or unauthorized exposure may be desirable in certain cases but is in itself less pertinent in general to the routing function.

One of the main problems of synchronizing security states of sleepy nodes, as listed in the last subsection, lies in difficulties in authentication; these nodes may not have received in time the most recent update of security material. Similarly, the issues of minimal manual configuration, prolonged rollout and delayed addition of nodes, and network topology changes also complicate key management. Hence, routing in LLNs needs to bootstrap the authentication process and allow for flexible expiration scheme of authentication credentials.

The vulnerability brought forth by some special-function nodes, e.g., LBRs, requires the assurance, particularly in a security context,

- o of the availability of communication channels and node resources;
- o that the neighbor discovery process operates without undermining routing availability.

There are other factors which are not part of a ROLL protocol but directly affecting its function. These factors include weaker barrier of accessing the data or security material stored on the nodes through physical means; therefore, the internal and external interfaces of a node need to be adequate for guarding the integrity, and possibly the confidentiality, of stored information, as well as the integrity of routing and route generation processes.

Each individual system's use and environment will dictate how the above objectives are applied, including the choices of security services as well as the strengths of the mechanisms that must be implemented. The next two sections take a closer look at how the ROLL security objectives may be compromised and how those potential compromises can be countered.

4. Threats and Attacks

This section outlines general categories of threats under the CIA model and highlights the specific attacks in each of these categories for ROLL. As defined in [RFC4949], a threat is "a potential for violation of security, which exists when there is a circumstance,

capability, action, or event that could breach security and cause harm." An attack is "an assault on system security that derives from an intelligent threat, i.e., an intelligent act that is a deliberate attempt (especially in the sense of a method or technique) to evade security services and violate the security policy of a system."

The subsequent subsections consider the threats and their realizing attacks that can cause security breaches under the CIA model to the routing assets and via the routing points of access identified in Section 3.1. The assessment steps through the security concerns of each routing asset and looks at the attacks that can exploit routing points of access. The threats and attacks identified are based on the routing model analysis and associated review of the existing literature. The manifestation of the attacks is assumed to be from either inside or outside attackers, whose capabilities may be limited to node-equivalent or more sophisticated computing platforms.

4.1. Threats and Attacks on Confidentiality

The assessment in Section 3.2 indicates that routing information assets are exposed to confidentiality threats from all points of access. The confidentiality threat space is thus defined by the access to routing information achievable through the communication exchanges between routing nodes together with the direct access to information maintained within the nodes.

4.1.1. Routing Exchange Exposure

Routing exchanges include both routing information as well as information associated with the establishment and maintenance of neighbor state information. As indicated in Section 3.1, the associated routing information assets may also include device specific resource information, such as memory, remaining power, etc., that may be metrics of the routing protocol.

The exposure of routing information exchanged will allow unauthorized sources to gain access to the content of the exchanges between communicating nodes. The exposure of neighbor state information will allow unauthorized sources to gain knowledge of communication links between routing nodes that are necessary to maintain routing information exchanges.

The forms of attack that allow unauthorized access or exposure of routing exchange information include

- o Deliberate exposure (where one party to the routing exchange is able to independently provide unauthorized access);

- o Sniffing (passive reading of transmitted data content);
- o Traffic analysis (evaluation of the network routing header information).

4.1.2. Routing Information (Routes and Network Topology) Exposure

Routes (which may be maintained in the form of the protocol forwarding table) and neighbor topology information are the products of the routing process that are stored within the node device databases.

The exposure of this information will allow unauthorized sources to gain direct access to the configuration and connectivity of the network thereby exposing routing to targeted attacks on key nodes or links. Since routes and neighbor topology information is stored within the node device, threats or attacks on the confidentiality of the information will apply to the physical device including specified and unspecified internal and external interfaces.

The forms of attack that allow unauthorized access or exposure of the routing information (other than occurring through explicit node exchanges) will include

- o Physical device compromise;
- o Remote device access attacks (including those occurring through remote network management or software/field upgrade interfaces).

More detailed descriptions of the exposure attacks on routing exchange and information will be given in Section 5 together with the corresponding countermeasures.

4.2. Threats and Attacks on Integrity

The assessment in Section 3.2 indicates that information and identity assets are exposed to integrity threats from all points of access. In other words, the integrity threat space is defined by the potential for exploitation introduced by access to assets available through routing exchanges and the on-device storage.

4.2.1. Routing Information Manipulation

Manipulation of routing information that range from neighbor states to derived routes will allow unauthorized sources to influence the operation and convergence of the routing protocols and ultimately impact the forwarding decisions made in the network. Manipulation of topology and reachability information will allow unauthorized sources

to influence the nodes with which routing information is exchanged and updated. The consequence of manipulating routing exchanges can thus lead to sub-optimality and fragmentation or partitioning of the network by restricting the universe of routers with which associations can be established and maintained. For example, being able to attract network traffic can make a blackhole attack more damaging.

The forms of attack that allow manipulation to compromise the content and validity of routing information include

- o Falsification, including overclaiming and misclaiming;
- o Routing information replay;
- o Byzantine (internal) attacks that permit corruption of routing information in the node even where the node continues to be a validated entity within the network (see, for example, [RFC4593] for further discussions on Byzantine attacks);
- o Physical device compromise or remote device access attacks.

4.2.2. Node Identity Misappropriation

Falsification or misappropriation of node identity between routing participants opens the door for other attacks; it can also cause incorrect routing relationships to form and/or topologies to emerge. Routing attacks may also be mounted through less sophisticated node identity misappropriation in which the valid information broadcast or exchanged by a node is replayed without modification. The receipt of seemingly valid information that is however no longer current can result in routing disruption, and instability (including failure to converge). Without measures to authenticate the routing participants and to ensure the freshness and validity of the received information the protocol operation can be compromised. The forms of attack that misuse node identity include

- o Identity attacks, including Sybil attacks in which a malicious node illegitimately assumes multiple identities;
- o Routing information replay.

4.3. Threats and Attacks on Availability

The assessment in Section 3.2 indicates that the process and resources assets are exposed to availability threats; attacks of this category may exploit directly or indirectly information exchange or forwarding (see [RFC4732] for a general discussion).

4.3.1. Routing Exchange Interference or Disruption

Interference or disruption of routing information exchanges will allow unauthorized sources to influence the operation and convergence of the routing protocols by impeding the regularity of routing information exchange.

The forms of attack that allow interference or disruption of routing exchange include

- o Routing information replay;
- o HELLO flood attacks and ACK spoofing;
- o Overload attacks.

In addition, attacks may also be directly conducted at the physical layer in the form of jamming or interfering.

4.3.2. Network Traffic Forwarding Disruption

The disruption of the network traffic forwarding capability of the network will undermine the central function of network routers and the ability to handle user traffic. This threat and the associated attacks affect the availability of the network because of the potential to impair the primary capability of the network.

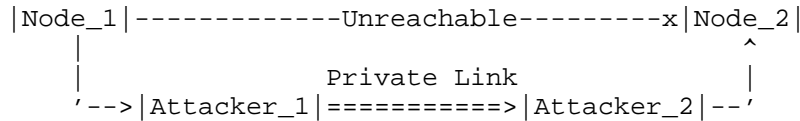
In addition to physical layer obstructions, the forms of attack that allows disruption of network traffic forwarding include [Karlof2003]

- o Selective forwarding attacks;
- o Wormhole attacks;
- o Sinkhole attacks.

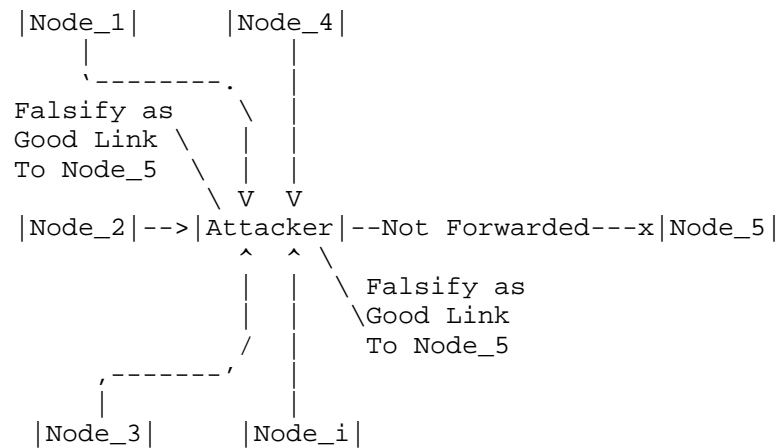
For reference, Figure 2 depicts the above listed three types of attacks.


```
|Node_1| -- (msg1 | msg2 | msg3) --> |Attacker| -- (msg1 | msg3) --> |Node_2|
```

(a) Selective Forwarding



(b) Wormhole



(c) Sinkhole

Figure 2: Selective Forwarding, Wormhole, and Sinkhole Attacks

4.3.3. Communications Resource Disruption

Attacks mounted against the communication channel resource assets needed by the routing protocol can be used as a means of disrupting its operation. However, while various forms of Denial of Service (DoS) attacks on the underlying transport subsystem will affect routing protocol exchanges and operation (for example physical layer RF jamming in a wireless network or link layer attacks), these attacks cannot be countered by the routing protocol. As such, the threats to the underlying transport network that supports routing is considered beyond the scope of the current document. Nonetheless, attacks on the subsystem will affect routing operation and so must be

directly addressed within the underlying subsystem and its implemented protocol layers.

4.3.4. Node Resource Exhaustion

A potential security threat to routing can arise from attempts to exhaust the node resource asset by initiating exchanges that can lead to the undue utilization or exhaustion of processing, memory, or energy resources. The establishment and maintenance of routing neighbors opens the routing process to engagement and potential acceptance of multiple neighboring peers. Association information must be stored for each peer entity and for the wireless network operation provisions made to periodically update and reassess the associations. An introduced proliferation of apparent routing peers can therefore have a negative impact on node resources.

Node resources may also be unduly consumed by the attackers attempting uncontrolled topology peering or routing exchanges, routing replays, or the generating of other data traffic floods. Beyond the disruption of communications channel resources, these threats may be able to exhaust node resources only where the engagements are able to proceed with the peer routing entities. Routing operation and network forwarding functions can thus be adversely impacted by node resources exhaustion that stems from attacks that include

- o Identity (including Sybil) attacks;
- o Routing information replay attacks;
- o HELLO flood attacks and ACK spoofing;
- o Overload attacks.

5. Countermeasures

By recognizing the characteristics of LLNs that may impact routing and identifying potential countermeasures, this analysis provides the basis for developing capabilities within ROLL protocols to deter the identified attacks and mitigate the threats. The following subsections consider such countermeasures by grouping the attacks according to the classification of the CIA model so that associations with the necessary security services are more readily visible. However, the considerations here are more systematic than confined to means available only within routing; the next section will then distill and make recommendations appropriate for a secured ROLL protocol.

5.1. Confidentiality Attack Countermeasures

Attacks on confidentiality may be mounted at the level of the routing information assets, at the points of access associated with routing exchanges between nodes, or through device interface access. To gain access to routing/topology information, the attacker may rely on a compromised node that deliberately exposes the information during the routing exchange process, may rely on passive sniffing or analysis of routing traffic, or may attempt access through a component or device interface of a tampered routing node.

5.1.1. Countering Deliberate Exposure Attacks

A deliberate exposure attack is one in which an entity that is party to the routing process or topology exchange allows the routing/topology information or generated route information to be exposed to an unauthorized entity during the exchange.

A prerequisite to countering this type of confidentiality attacks associated with the routing/topology exchange is to ensure that the communicating nodes are authenticated prior to data encryption applied in the routing exchange. Authentication ensures that the nodes are who they claim to be even though it does not provide an indication of whether the node has been compromised.

To prevent deliberate exposure, the process that communicating nodes use for establishing communication session keys must be peer-to-peer, between the routing initiating and responding nodes, so that neither node can independently weaken the confidentiality of the exchange without the knowledge of its communicating peer. A deliberate exposure attack will therefore require more overt and independent action on the part of the offending node.

Note that the same measures which apply to securing routing/topology exchanges between operational nodes must also extend to field tools and other devices used in a deployed network where such devices can be configured to participate in routing exchanges.

5.1.2. Countering Sniffing Attacks

A sniffing attack seeks to breach routing confidentiality through passive, direct analysis and processing of the information exchanges between nodes. A sniffing attack in an LLN that is not based on a physical device compromise will rely on the attacker attempting to directly derive information from the over-the-shared-medium routing/topology communication exchange (neighbor discovery exchanges may of necessity be conducted in the clear thus limiting the extent to which the information can be kept confidential).

Sniffing attacks can be directly countered through the use of data encryption for all routing exchanges. Only when a validated and authenticated node association is completed will routing exchange be allowed to proceed using established session confidentiality keys and an agreed confidentiality algorithm. The level of security applied in providing confidentiality will determine the minimum requirement for an attacker mounting this passive security attack. The possibility of incorporating options for security level and algorithms is further considered in Section 6.5. Because of the resource constraints of LLN devices, symmetric (private) key session security will provide the best trade-off in terms of node and channel resource overhead and the level of security achieved. This will of course not preclude the use of asymmetric (public) key encryption during the session key establishment phase.

As with the key establishment process, data encryption must include an authentication prerequisite to ensure that each node is implementing a level of security that prevents deliberate or inadvertent exposure. The authenticated key establishment will ensure that confidentiality is not compromised by providing the information to an unauthorized entity (see also [Huang2003]).

Based on the current state of the art, a minimum 128-bit key length should be applied where robust confidentiality is demanded for routing protection. This session key shall be applied in conjunction with an encryption algorithm that has been publicly vetted and where applicable approved for the level of security desired. Algorithms such as the Advanced Encryption Standard (AES) [FIPS197], adopted by the U.S. government, or Kasumi-Misty [Kasumi3gpp], adopted by the 3GPP 3rd generation wireless mobile consortium, are examples of symmetric-key algorithms capable of ensuring robust confidentiality for routing exchanges. The key length, algorithm and mode of operation will be selected as part of the overall security trade-off that also achieves a balance with the level of confidentiality afforded by the physical device in protecting the routing assets (see Section 5.1.4 below).

As with any encryption algorithm, the use of ciphering synchronization parameters and limitations to the usage duration of established keys should be part of the security specification to reduce the potential for brute force analysis.

5.1.3. Countering Traffic Analysis

Traffic analysis provides an indirect means of subverting confidentiality and gaining access to routing information by allowing an attacker to indirectly map the connectivity or flow patterns (including link-load) of the network from which other attacks can be

mounted. The traffic analysis attack on an LLN, especially one founded on shared medium, may be passive and relying on the ability to read the immutable source/destination routing information that must remain unencrypted to permit network routing. Alternatively, attacks can be active through the injection of unauthorized discovery traffic into the network. By implementing authentication measures between communicating nodes, active traffic analysis attacks can be prevented within the LLN thereby reducing confidentiality vulnerabilities to those associated with passive analysis.

One way in which passive traffic analysis attacks can be muted is through the support of load balancing that allows traffic to a given destination to be sent along diverse routing paths. Where the routing protocol supports load balancing along multiple links at each node, the number of routing permutations in a wide area network surges thus increasing the cost of traffic analysis. Network analysis through this passive attack will require a wider array of analysis points and additional processing on the part of the attacker. Note however that where network traffic is dispersed as a countermeasure there may be implications beyond routing with regard to general traffic confidentiality. Another approach to countering passive traffic analysis could be for nodes to maintain constant amount of traffic to different destinations through the generation of arbitrary traffic flows; the drawback of course would be the consequent overhead. In LLNs, the diverse radio connectivity and dynamic links (including potential frequency hopping), or a complex wiring system hidden from sight, will help to further mitigate traffic analysis attacks when load balancing is also implemented.

The only means of fully countering a traffic analysis attack is through the use of tunneling (encapsulation) where encryption is applied across the entirety of the original packet source/destination addresses. With tunneling there is a further requirement that the encapsulating intermediate nodes apply an additional layer of routing so that traffic arrives at the destination through dynamic routes. For some LLNs, memory and processing constraints as well as the limitations of the communication channel will preclude both the additional routing traffic overhead and the node implementation required for tunneling countermeasures to traffic analysis.

5.1.4. Countering Physical Device Compromise

Section 4 identified that many threats to the routing functionality may involve compromised devices. For the sake of completeness, this subsection examines how to counter physical device compromise, without restricting the consideration to only those methods and apparatuses available to an LLN routing protocol.

Given the distributed nature of LLNs and the varying environment of deployed devices, confidentiality of routing assets and points of access may rely heavily on the security of the routing devices. One means of precluding attacks on the physical device is to prevent physical access to the node through other external security means. However, given the environment in which many LLNs operate, preventing unauthorized access to the physical device cannot be assured. Countermeasures must therefore be employed at the device and component level so that routing/topology or neighbor information and stored route information cannot be accessed even if physical access to the node is obtained.

With the physical device in the possession of an attacker, unauthorized information access can be attempted by probing internal interfaces or device components. Device security must therefore move to preventing the reading of device processor code or memory locations without the appropriate security keys and in preventing the access to any information exchanges occurring between individual components. Information access will then be restricted to external interfaces in which confidentiality, integrity, and authentication measures can be applied.

To prevent component information access, deployed routing devices must ensure that their implementation avoids address or data buses being connected to external general purpose input/output (GPIO) pins. Beyond this measure, an important component interface to be protected against attack is the Joint Test Action Group (JTAG) [IEEE1149.1] interface used for component and populated circuit board testing after manufacture. To provide security on the routing devices, components should be employed that allow fuses on the JTAG interfaces to be blown to disable access. This will raise the bar on unauthorized component information access within a captured device.

At the device level a key component information exchange is between the microprocessor and its associated external memory. While encryption can be implemented to secure data bus exchanges, the use of integrated physical packaging which avoids inter-component exchanges (other than secure external device exchanges) will increase routing security against a physical device interface attack. With an integrated package and disabled internal component interfaces, the level of physical device security can be controlled by managing the degree to which the device packaging is protected against expert physical decomposition and analysis.

The device package should be hardened such that attempts to remove the integrated components will result in damage to access interfaces, ports or pins that prevent retrieval of code or stored information. The degree of Very Large Scale Integration (VLSI) or Printed Circuit

Board (PCB) package security through manufacture can be selected as a trade-off or desired security consistent with the level of security achieved by measures applied for other routing assets and points of access. With package hardening and restricted component access countermeasures, the security level will be raised to that provided by measures employed at the external communications interfaces.

Another area of node interface vulnerability is that associated with interfaces provided for remote software or firmware upgrades. This may impact both routing information and routing/topology exchange security where it leads to unauthorized upgrade or change to the routing protocol running on a given node as this type of attack can allow for the execution of compromised or intentionally malicious routing code on multiple nodes. Countermeasures to this device interface confidentiality attack needs to be addressed in the larger context of node remote access security. This will ensure not only the authenticity of the provided code (including routing protocol) but that the process is initiated by an authorized (authenticated) entity. For example, digital signing of firmware by an authorized entity will provide an appropriate countermeasure.

The above identified countermeasures against attacks on routing information confidentiality through internal device interface compromise must be part of the larger LLN system security as they cannot be addressed within the routing protocol itself. Similarly, the use of field tools or other devices that allow explicit access to node information must implement security mechanisms to ensure that routing information can be protected against unauthorized access. These protections will also be external to the routing protocol and hence not part of ROLL.

5.1.5. Countering Remote Device Access Attacks

Where LLN nodes are deployed in the field, measures are introduced to allow for remote retrieval of routing data and for software or field upgrades. These paths create the potential for a device to be remotely accessed across the network or through a provided field tool. In the case of network management a node can be directly requested to provide routing tables and neighbor information.

To ensure confidentiality of the node routing information against attacks through remote access, any local or remote device requesting routing information must be authenticated to ensure authorized access. Since remote access is not invoked as part of a routing protocol security of routing information stored on the node against remote access will not be addressable as part of the routing protocol.

5.2. Integrity Attack Countermeasures

Integrity attack countermeasures address routing information manipulation, as well as node identity and routing information misuse. Manipulation can occur in the form of falsification attack and physical compromise. To be effective, the following development considers the two aspects of falsification, namely, the unauthorized modifications and the overclaiming and misclaiming content. The countering of physical compromise was considered in the previous section and is not repeated here. With regard to misuse, there are two types of attacks to be deterred, identity attacks and replay attacks.

5.2.1. Countering Unauthorized Modification Attacks

Unauthorized modifications may occur in the form of altering the message being transferred or the data stored. Therefore, it is necessary to ensure that only authorized nodes can change the portion of the information that is allowed to be mutable, while the integrity of the rest of the information is protected, e.g., through well-studied cryptographic mechanisms.

Unauthorized modifications may also occur in the form of insertion or deletion of messages during protocol changes. Therefore, the protocol needs to ensure the integrity of the sequence of the exchange sequence.

The countermeasure to unauthorized modifications needs to

- o implement access control on storage;
- o provide data integrity service to transferred messages and stored data;
- o include sequence number under integrity protection.

5.2.2. Countering Overclaiming and Misclaiming Attacks

Both overclaiming and misclaiming aim to introduce false routes or topology that would not be generated by the network otherwise, while there are not necessarily unauthorized modifications to the routing messages or information. The requisite for a counter is the capability to determine unreasonable routes or topology.

The counter to overclaiming and misclaiming may employ

- o comparison with historical routing/topology data;

- o designs which restrict realizable network topologies.

5.2.3. Countering Identity (including Sybil) Attacks

Identity attacks, sometimes simply called spoofing, seek to gain or damage assets whose access is controlled through identity. In routing, an identity attacker can illegitimately participate in routing exchanges, distribute false routing information, or cause an invalid outcome of a routing process.

A perpetrator of Sybil attacks assumes multiple identities. The result is not only an amplification of the damage to routing, but extension to new areas, e.g., where geographic distribution is explicitly or implicitly an asset to an application running on the LLN, for example, the LBR in a P2MP or MP2P LLN.

The countering of identity attacks need to ensure the authenticity and liveness of the parties of a message exchange. The means may be through the use of shared key- or public key-based authentication scheme. On the one hand, the large-scale nature of the LLNs makes the network-wide shared key scheme undesirable from a security perspective; on the other hand, public-key based approaches generally require more computational resources. Each system will need to make trade-off decisions based on its security requirements. As an example, [Wander2005] compared the energy consumption between two public-key algorithms on a low-power microcontroller, with reference to a symmetric-key algorithm and a hash algorithm.

5.2.4. Countering Routing Information Replay Attacks

In routing, message replay can result in false topology and/or routes. The counter of replay attacks needs to ensure the freshness of the message. On the one hand, there are a number of mechanisms commonly used for countering replay, e.g., with a counter. On the other hand, the choice should take into account how a particular mechanism is made available in an LLN. For example, many LLNs have a central source of time and have it distributed by relaying, such that secured time distribution becomes a prerequisite of using timestamping to counter replay.

5.2.5. Countering Byzantine Routing Information Attacks

Where a node is captured or compromised but continues to operate for a period with valid network security credentials, the potential exists for routing information to be manipulated. This compromise of the routing information could thus exist in spite of security countermeasures that operate between the peer routing devices.

Consistent with the end-to-end principle of communications, such an attack can only be fully addressed through measures operating directly between the routing entities themselves or by means of external entities able to access and independently analyze the routing information. Verification of the authenticity and liveness of the routing entities can therefore only provide a limited counter against internal (Byzantine) node attacks.

For link state routing protocols where information is flooded with, for example, areas (OSPF [RFC2328]) or levels (ISIS [RFC1142]), countermeasures can be directly applied by the routing entities through the processing and comparison of link state information received from different peers. By comparing the link information from multiple sources decisions can be made by a routing node or external entity with regard to routing information validity; see Chapter 2 of [Perlman1988] for a discussion on flooding attacks.

For distance vector protocols where information is aggregated at each routing node it is not possible for nodes to directly detect Byzantine information manipulation attacks from the routing information exchange. In such cases, the routing protocol must include and support indirect communications exchanges between non-adjacent routing peers to provide a secondary channel for performing routing information validation. S-RIP [Wan2004] is an example of the implementation of this type of dedicated routing protocol security where the correctness of aggregate distance vector information can only be validated by initiating confirmation exchanges directly between nodes that are not routing neighbors.

Alternatively, an entity external to the routing protocol would be required to collect and audit routing information exchanges to detect the Byzantine attack. In the context of the current security analysis, any protection against Byzantine routing information attacks will need to be directly included within the mechanisms of the ROLL routing protocol. This can be implemented where such an attack is considered relevant even within the physical device protections discussed in Section 5.1.4.

5.3. Availability Attack Countermeasures

As alluded to before, availability requires that routing information exchanges and forwarding mechanisms be available when needed so as to guarantee proper functioning of the network. This may, e.g., include the correct operation of routing information and neighbor state information exchanges, among others. We will highlight the key features of the security threats along with typical countermeasures to prevent or at least mitigate them. We will also note that an availability attack may be facilitated by an identity attack as well

as a replay attack, as was addressed in Section 5.2.3 and Section 5.2.4, respectively.

5.3.1. Countering HELLO Flood Attacks and ACK Spoofing Attacks

HELLO Flood [Karlof2003],[I-D.suhopark-hello-wsn] and ACK Spoofing attacks are different but highly related forms of attacking an LLN. They essentially lead nodes to believe that suitable routes are available even though they are not and hence constitute a serious availability attack.

The origin of facilitating a HELLO flood attack lies in the fact that many routing protocols require nodes to send HELLO packets either upon joining or in regular intervals so as to announce or confirm their existence to the network. Those nodes that receive the HELLO packet assume that they are indeed neighbors.

With this in mind, a malicious node can send or replay HELLO packets using, e.g., a higher transmission power. That creates the false illusion of being a neighbor to an increased number of nodes in the network, thereby effectively increasing its unidirectional neighborhood cardinality. The high quality of the falsely advertised link may coerce nodes to route data via the malicious node. However, those affected nodes, for which the malicious node is in fact unreachable, never succeed in their delivery and the packets are effectively dropped. The symptoms are hence similar to those of a sinkhole, wormhole and selective forwarding attack.

A malicious HELLO flood attack clearly distorts the network topology. It thus affects protocols building and maintaining the network topology as well as routing protocols as such, since the attack is primarily targeted on protocols that require sharing of information for topology maintenance or flow control.

To counter HELLO flood attacks, several mutually non-exclusive methods are feasible:

- o restricting neighborhood cardinality;
- o facilitating multipath routing;
- o verifying bidirectionality.

Restricting the neighborhood cardinality prevents malicious nodes from having an extended set of neighbors beyond some tolerated threshold and thereby preventing topologies to be built where malicious nodes have a false neighborhood set. Furthermore, as shown in [I-D.suhopark-hello-wsn], if the routing protocol supports

multiple paths from a sensing node towards several LBRs then HELLO flood attacks can also be diminished; however, the energy-efficiency of such approach is clearly sub-optimal. Finally, verifying that the link is truly bidirectional by means of, e.g., an ACK handshake and appropriate security measures ensures that a communication link is only established if not only the affected node is within range of the malicious node but also vice versa. Whilst this does not really eliminate the problem of HELLO flooding, it greatly reduces the number of affected nodes and the probability of such an attack succeeding.

As for the latter, the adversary may spoof the ACK messages to convince the affected node that the link is truly bidirectional and thereupon drop, tunnel or selectively forward messages. Such ACK spoofing attack is possible if the malicious node has a receiver which is significantly more sensitive than that of a normal node, thereby effectively extending its range. Since an ACK spoofing attack facilitates a HELLO flood attack, similar countermeasure are applicable here. Viable counter and security measures for both attacks have been exposed in [I-D.suhopark-hello-wsn]

5.3.2. Countering Overload Attacks

Overload attacks are a form of DoS attack in that a malicious node overloads the network with irrelevant traffic, thereby draining the nodes' energy store quicker, when the nodes rely on batteries or energy scavenging. It thus significantly shortens the lifetime of networks of energy-constrained nodes and constitutes another serious availability attack.

With energy being one of the most precious assets of LLNs, targeting its availability is a fairly obvious attack. Another way of depleting the energy of an LLN node is to have the malicious node overload the network with irrelevant traffic. This impacts availability since certain routes get congested which

- o renders them useless for affected nodes and data can hence not be delivered;
- o makes routes longer as shortest path algorithms work with the congested network;
- o depletes battery and energy scavenging nodes quicker and thus shortens the network's availability at large.

Overload attacks can be countered by deploying a series of mutually non-exclusive security measures:

- o introduce quotas on the traffic rate each node is allowed to send;
- o isolate nodes which send traffic above a certain threshold based on system operation characteristics;
- o allow only trusted data to be received and forwarded.

As for the first one, a simple approach to minimize the harmful impact of an overload attack is to introduce traffic quotas. This prevents a malicious node from injecting a large amount of traffic into the network, even though it does not prevent said node from injecting irrelevant traffic at all. Another method is to isolate nodes from the network at the network layer once it has been detected that more traffic is injected into the network than allowed by a prior set or dynamically adjusted threshold. Finally, if communication is sufficiently secured, only trusted nodes can receive and forward traffic which also lowers the risk of an overload attack.

Receiving nodes that validate signatures and sending nodes that encrypt messages need to be cautious of cryptographic processing usage when validating signatures and encrypting messages. Where feasible, certificates should be validated prior to use of the associated keys to counter potential resource overloading attacks. The associated design decision needs to also consider that the validation process requires resources and thus itself could be exploited for attacks. Alternatively, resource management limits can be placed on routing security processing events (see the comment in Section 6, paragraph 4, of [RFC5751]).

5.3.3. Countering Selective Forwarding Attacks

Selective forwarding attacks are another form of DoS attack which impacts the routing path availability.

An insider malicious node basically blends neatly in with the network but then may decide to forward and/or manipulate certain packets. If all packets are dropped, then this attacker is also often referred to as a "black hole". Such a form of attack is particularly dangerous if coupled with sinkhole attacks since inherently a large amount of traffic is attracted to the malicious node and thereby causing significant damage. In a shared medium, an outside malicious node would selectively jam overheard data flows, where the thus caused collisions incur selective forwarding.

Selective Forwarding attacks can be countered by deploying a series of mutually non-exclusive security measures:

- o multipath routing of the same message over disjoint paths;
- o dynamically selecting the next hop from a set of candidates.

The first measure basically guarantees that if a message gets lost on a particular routing path due to a malicious selective forwarding attack, there will be another route which successfully delivers the data. Such a method is inherently suboptimal from an energy consumption point of view; it is also suboptimal from a network utilization perspective. The second method basically involves a constantly changing routing topology in that next-hop routers are chosen from a dynamic set in the hope that the number of malicious nodes in this set is negligible. A routing protocol that allows for disjoint routing paths may also be useful.

5.3.4. Countering Sinkhole Attacks

In sinkhole attacks, the malicious node manages to attract a lot of traffic mainly by advertising the availability of high-quality links even though there are none [Karlof2003]. It hence constitutes a serious attack on availability.

The malicious node creates a sinkhole by attracting a large amount of, if not all, traffic from surrounding neighbors by advertising in and outwards links of superior quality. Affected nodes hence eagerly route their traffic via the malicious node which, if coupled with other attacks such as selective forwarding, may lead to serious availability and security breaches. Such an attack can only be executed by an inside malicious node and is generally very difficult to detect. An ongoing attack has a profound impact on the network topology and essentially becomes a problem of flow control.

Sinkhole attacks can be countered by deploying a series of mutually non-exclusive security measures:

- o use geographical insights for flow control;
- o isolate nodes which receive traffic above a certain threshold;
- o dynamically pick up next hop from set of candidates;
- o allow only trusted data to be received and forwarded.

Whilst most of these countermeasures have been discussed before, the use of geographical information deserves further attention. Essentially, if geographic positions of nodes are available, then the network can assure that data is actually routed towards the intended destination and not elsewhere. On the other hand, geographic

position is a sensitive information that has security and/or privacy consequences (see Section 6.1).

5.3.5. Countering Wormhole Attacks

In wormhole attacks at least two malicious nodes shortcut or divert the usual routing path by means of a low-latency out-of-band channel [Karlof2003]. This changes the availability of certain routing paths and hence constitutes a serious security breach.

Essentially, two malicious insider nodes use another, more powerful, transmitter to communicate with each other and thereby distort the would-be-agreed routing path. This distortion could involve shortcutting and hence paralyzing a large part of the network; it could also involve tunneling the information to another region of the network where there are, e.g., more malicious nodes available to aid the intrusion or where messages are replayed, etc. In conjunction with selective forwarding, wormhole attacks can create race conditions which impact topology maintenance, routing protocols as well as any security suits built on "time of check" and "time of use".

Wormhole attacks are very difficult to detect in general but can be mitigated using similar strategies as already outlined above in the context of sinkhole attacks.

6. ROLL Security Features

The assessments and analysis in Section 4 examined all areas of threats and attacks that could impact routing, and the countermeasures presented in Section 5 were reached without confining the consideration to means only available to routing. This section puts the results into perspective and provides a framework for addressing the derived set of security objectives that must be met by the routing protocol(s) specified by the ROLL Working Group. It bears emphasizing that the target here is a generic, universal form of the protocol(s) specified and the normative keywords are mainly to convey the relative level of importance or urgency of the features specified.

In this view, 'MUST' is used to define the requirements that are specific to the routing protocol and that are essential for an LLN routing protocol to ensure that routing operation can be maintained. Adherence to MUST requirements is needed to directly counter attacks that can affect the routing operation (such as those that can impact maintained or derived routing/forwarding tables). 'SHOULD' is used to define requirements that counter indirect routing attacks where

such attacks do not of themselves affect routing but can assist an attacker in focusing its attack resources to impact network operation (such as DoS targeting of key forwarding nodes). 'MAY' covers optional requirements that can further enhance security by increasing the space over which an attacker must operate or the resources that must be applied. While in support of routing security, where appropriate, these requirements may also be addressed beyond the network routing protocol at other system communications layers.

The first part of this section, Section 6.1 to Section 6.3, is a prescription of ROLL security features of measures that can be addressed as part of the routing protocol itself. As routing is one component of an LLN system, the actual strength of the security services afforded to it should be made to conform to each system's security policy; how a design may address the needs of the urban, industrial, home automation, and building automation application domains also needs to be considered. The second part of this section, Section 6.4 and Section 6.5, discusses system security aspects that may impact routing but that also require considerations beyond the routing protocol, as well as potential approaches.

If an LLN employs multicast and/or anycast, these alternative communications modes MUST be secured with the same routing security services specified in this section. Furthermore, irrespective of the modes of communication, nodes MUST provide adequate physical tamper resistance commensurate with the particular application domain environment to ensure the confidentiality, integrity, and availability of stored routing information.

6.1. Confidentiality Features

With regard to confidentiality, protecting the routing/topology information from eavesdropping or unauthorized exposure is not directly essential to maintaining the routing function. Breaches of confidentiality may lead to other attacks or the focusing of an attacker's resources (see Section 4.1) but does not of itself directly undermine the operation of the routing function. However, to protect against, and improve vulnerability against other more direct attacks, routing information confidentiality should be protected. Thus, a secured ROLL protocol

- o MUST implement payload encryption;
- o MUST provide privacy when geographic information is used (see, e.g., [RFC3693]);
- o MAY provide tunneling;

- o MAY provide load balancing.

Where confidentiality is incorporated into the routing exchanges, encryption algorithms and key lengths need to be specified in accordance with the level of protection dictated by the routing protocol and the associated application domain transport network. In terms of the life time of the keys, the opportunity to periodically change the encryption key increases the offered level of security for any given implementation. However, where strong cryptography is employed, physical, procedural, and logical data access protection considerations may have more significant impact on cryptoperiod selection than algorithm and key size factors. Nevertheless, in general, shorter cryptoperiods, during which a single key is applied, will enhance security.

Given the mandatory protocol requirement to implement routing node authentication as part of routing integrity (see Section 6.2), key exchanges may be coordinated as part of the integrity verification process. This provides an opportunity to increase the frequency of key exchange and shorten the cryptoperiod as a complement to the key length and encryption algorithm required for a given application domain. For LLNs, the coordination of confidentiality key management with the implementation of node device authentication can thus reduce the overhead associated with supporting data confidentiality. If a new ciphering key is concurrently generated or updated in conjunction with the mandatory authentication exchange occurring with each routing peer association, signaling exchange overhead can be reduced.

6.2. Integrity Features

The integrity of routing information provides the basis for ensuring that the function of the routing protocol is achieved and maintained. To protect integrity, a secured ROLL protocol

- o MUST provide and verify message integrity (including integrity of the encrypted message when confidentiality is applied);
- o MUST verify the authenticity and liveness of both principals of a connection (independent of the device interface over which the information is received or accessed);
- o MUST verify message sequence;
- o SHOULD incorporate protocol-specific parameter validity range checks, change increments, and message event frequency checks, etc. as a means of countering intentional or unintentional Byzantine threats;

- o MAY incorporate external consistency checking and auditing of routing information to protect against intentional or unintentional Byzantine-induced network anomalies.

In conjunction with the integrity protection requirements, a secured ROLL protocol SHOULD log, against the offending node, any security failure that occurs after a valid integrity check. The record of such failures (as may result, for example, from incorrect security policy configuration) can provide the basis for nodes to avoid initiating routing access to the offender or be used for further system countermeasures in the case of potential insider attacks. All integrity security failures SHOULD be logged, where feasible, but cannot be reliably considered as countering against the offending source(s).

Depending on the nature of the routing protocol, e.g., distance vector or link state, additional measures may be necessary when the validity of the routing information is of concern. In the most basic form, verification of routing peer authenticity and liveness can be used to build a "chain of trust" along the path the routing information flows, such that network-wide information is validated through the concatenation of trust established at each individual routing peer exchange. This is particularly important in the case of distance vector-based routing protocols, where information is updated at intermediate nodes. In such cases, there are no direct means within routing for a receiver to verify the validity of the routing information beyond the current exchange; as such, nodes would need to be able to communicate and request information from non-adjacent peers (see [Wan2004]) to provide information integrity assurances. With link state-based protocols, on the other hand, routing information can be signed at the source thus providing a means for validating information that originates beyond a routing peer. Therefore, where necessary, a secured ROLL protocol MAY use security auditing mechanisms that are external to routing to verify the validity of the routing information content exchanged among routing peers.

6.3. Availability Features

Availability of routing information is linked to system and network availability which in the case of LLNs require a broader security view beyond the requirements of the routing entities (see Section 6.5). Where availability of the network is compromised, routing information availability will be accordingly affected. However, to specifically assist in protecting routing availability

- o MAY restrict neighborhood cardinality;

- o MAY use multiple paths;
- o MAY use multiple destinations;
- o MAY choose randomly if multiple paths are available;
- o MAY set quotas to limit transmit or receive volume;
- o MAY use geographic information for flow control.

6.4. Security Key Management

The functioning of the routing security services requires keys and credentials. Therefore, even though not directly a ROLL security requirement, an LLN MUST have a process for initial key and credential configuration, as well as secure storage within the associated devices (including use of trusted platform modules where feasible and appropriate to the operating environment). Beyond initial credential configuration, an LLN is also encouraged to have automatic procedures for the long-term revocation and replacement of the maintained security credentials.

Individual routing peer associations and signaling exchanges will require the generation and use of keys that may be derived from secret or public key exchanges or directly obtained through device configuration means. The routing protocol specification MUST include mechanisms for identifying and synchronizing the keys used for securing exchanges between the routing entities. The keys used to protect the communications between the routing entities MAY be implicit, configured keys or may be explicitly generated as part of the routing signaling exchange.

For the keys used to protect routing associations, the routing protocol(s) specified by the ROLL Working Group SHOULD employ key management mechanisms consistent with the guidelines given in [RFC4107]. Based on that RFC's recommendations, many LLNs, particularly given the intended scale and ad hoc device associations, will meet the requirement for supporting automated key management in conjunction with the routing protocol operation. These short-term, automated routing session keys may be derived from pre-stored security credentials or can be generated through key management mechanisms that are defined as part of the routing protocol exchange. Beyond the automated short-term keys, a long-term key management mechanism SHOULD also be defined for changing or updating the credentials from which short-term routing association key material is derived.

The use of a public key infrastructure (PKI), where feasible, can be

used to support authenticated short-term key management as well as the distribution of long-term routing security keying material. Note that where the option for a PKI is supported for security of the routing protocol itself, the routing protocol MUST include provisions for public key certificates to be included or referenced within routing messages to allow a node's public key to be shared with communicating peers. Even if the certificate itself is not distributed by the node, there needs to be a mechanism to inform the receiving node where to find the certificate and obtain associated validation information; see [RFC3029] for an example of the kind of localized PKI support that may be applied in a given LLN environment. Where PKI systems are not feasible, the key management system MUST support means for secure configuration, device authentication, and adherence to secure key wrapping principles for the secure distribution and update of device keys.

LLN routing protocols SHOULD be designed to allow the use of existing and validated key management schemes. As part of the LLN optimization, these schemes may be independent of the routing protocol and part of the broader LLN system security specifications. Where the long-term key management is defined separately from the routing protocol security, LLN application domains can appropriately employ IETF-standard key management specifications. Established key management solutions such as IKEv2 [RFC5996] or MIKEY [RFC3830], which supports several alternative private, public, or Diffie-Hellman key distribution methods (see [RFC5197]), can thus be adapted for use in LLNs. For example, see [I-D.alexander-roll-mikey-lln-key-mgmt]. Group key management and distribution methods may also be developed based on the architecture principles defined in MSEC [RFC4046].

6.5. Consideration on Matching Application Domain Needs

Providing security within an LLN requires considerations that extend beyond routing security to the broader LLN application domain security implementation. In other words, as routing is one component of an LLN system, the actual strength of the implemented security algorithms for the routing protocol MUST be made to conform to the system's target level of security. The development so far takes into account collectively the impacts of the issues gathered from [RFC5548], [RFC5673], [RFC5826], and [RFC5867]. The following two subsections first consider from an architectural perspective how the security design of a ROLL protocol may be made to adapt to the four application domains, and then examine mechanisms and protocol operations issues.

6.5.1. Security Architecture

The first challenge for a ROLL protocol security design is to have an architecture that can adequately address a set of very diversified needs. It is mainly a consequence of the fact that there are both common and non-overlapping requirements from the four application domains, while, conceivably, each individual application will present yet its own unique constraints.

For a ROLL protocol, the security requirements defined in Section 6.1 to Section 6.4 can be addressed at two levels: 1) through measures implemented directly within the routing protocol itself and initiated and controlled by the routing protocol entities; or 2) through measures invoked on behalf of the routing protocol entities but implemented within the part of the network over which the protocol exchanges occur.

Where security is directly implemented as part of the routing protocol the security requirements configured by the user (system administrator) will operate independently of the lower layers. OSPFv2 [RFC2328] is an example of such an approach in which security parameters are exchanged and assessed within the routing protocol messages. In this case, the mechanism may be, e.g., a header containing security material of configurable security primitives in the fashion of OSPFv2 or RIPv2 [RFC2453]. Where IPsec [RFC4301] is employed to secure the network, the included protocol-specific (OSPF or RIP) security elements are in addition to and independent of those at the network layer. In the case of LLNs or other networks where system security mandates protective mechanisms at other lower layers of the network, security measures implemented as part of the routing protocol will be redundant to security measures implemented elsewhere as part of the protocol stack.

Security mechanisms built into the routing protocol can ensure that all desired countermeasures can be directly addressed by the protocol all the way to the endpoint of the routing exchange. In particular, routing protocol Byzantine attacks by a compromised node that retains valid network security credentials can only be detected at the level of the information exchanged within the routing protocol. Such attacks aimed at the manipulation of the routing information can only be fully addressed through measures operating directly between the routing entities themselves or external entities able to access and analyze the routing information (see discussion in Section 5.2.5).

On the other hand, it is more desirable from an LLN device perspective that the ROLL protocol is integrated into the framework of an overall system architecture where the security facility may be shared by different applications and/or across layers for efficiency,

and where security policy and configurations can be consistently specified. See, for example, considerations made in RIPng [RFC2080] or the approach presented in [Messerges2003].

Where the routing protocol is able to rely on security measures configured within other layers of the protocol stack, greater system efficiency can be realized by avoiding potentially redundant security. Relying on an open trust model [Messerges2003], the security requirements of the routing protocol can be more flexibly met at different layers of the transport network; measures that must be applied to protect the communications network are concurrently able to provide the needed routing protocol protection.

For example, where a given security encryption scheme is deemed the appropriate standard for network confidentiality of data exchanges at the link layer, that level of security is directly provided to routing protocol exchanges across the local link. Similarly, where a given authentication procedure is stipulated as part of the standard required for authenticating network traffic, that security provision can then meet the requirement needed for authentication of routing exchanges. In addition, in the context of the different LLN application domains, the level of security specified for routing can and should be consistent with that considered appropriate for protecting the network within the given environment.

A ROLL protocol MUST be made flexible by a design that offers the configuration facility so that the user (network administrator) can choose the security settings that match the application's needs. Furthermore, in the case of LLNs, that flexibility SHOULD extend to allowing the routing protocol security requirements to be met by measures applied at different protocol layers, provided the identified requirements are collectively met.

Since Byzantine attacks that can affect the validity of the information content exchanged between routing entities can only be directly countered at the routing protocol level, the ROLL protocol MAY support mechanisms for verifying routing data validity that extend beyond the chain of trust created through device authentication. This protocol-specific security mechanism SHOULD be made optional within the protocol allowing it to be invoked according to the given routing protocol and application domain and as selected by the system user. All other ROLL security mechanisms needed to meet the above identified routing security requirements can be flexibly implemented within the transport network (at the IP network layer or higher or lower protocol layers(s)) according to the particular application domain and user network configuration.

Based on device capabilities and the spectrum of operating

environments it would be difficult for a single fixed security design to be applied to address the diversified needs of the urban, industrial, home, and building ROLL application domains, and foreseeable others, without forcing a very low common denominator set of requirements. On the other hand, providing four individual domain designs that attempt to a priori match each individual domain is also very unlikely to provide a suitable answer given the degree of network variability even within a given domain; furthermore, the type of link layers in use within each domain also influences the overall security.

Instead, the framework implementation approach recommended is for optional, routing protocol-specific measures that can be applied separately from, or together with, flexible transport network mechanisms. Protocol-specific measures include the specification of valid parameter ranges, increments and/or event frequencies that can be verified by individual routing devices. In addition to deliberate attacks this allows basic protocol sanity checks against unintentional mis-configuration. Transport network mechanisms would include out-of-band communications that may be defined to allow an external entity to request and process individual device information as a means to effecting an external verification of the derived network routing information to identify the existence of intentional or unintentional network anomalies.

This approach allows countermeasures against internal attacks to be applied in environments where applicable threats exist. At the same time, it allows routing protocol security to be supported through measures implemented within the transport network that are consistent with available system resources and commensurate and consistent with the security level and strength applied in the particular application domain networks.

6.5.2. Mechanisms and Operations

With an architecture allowing different configurations to meet the application domain needs, the task is then to find suitable mechanisms. For example, one of the main problems of synchronizing security states of sleepy nodes lies in difficulties in authentication; these nodes may not have received in time the most recent update of security material. Similarly, the issues of minimal manual configuration, prolonged rollout and delayed addition of nodes, and network topology changes also complicate security management. In many cases the ROLL protocol may need to bootstrap the authentication process and allow for a flexible expiration scheme of authentication credentials. This exemplifies the need for the coordination and interoperation between the requirements of the ROLL routing protocol and that of the system security elements.

Similarly, the vulnerability brought forth by some special-function nodes, e.g., LBRs requires the assurance, particularly, of the availability of communication channels and node resources, or that the neighbor discovery process operates without undermining routing availability.

There are other factors which are not part of a ROLL routing protocol but which can still affect its operation. These include elements such as weaker barrier to accessing the data or security material stored on the nodes through physical means; therefore, the internal and external interfaces of a node need to be adequate for guarding the integrity, and possibly the confidentiality, of stored information, as well as the integrity of routing and route generation processes.

Figure 3 provides an overview of the larger context of system security and the relationship between ROLL requirements and measures and those that relate to the LLN system.

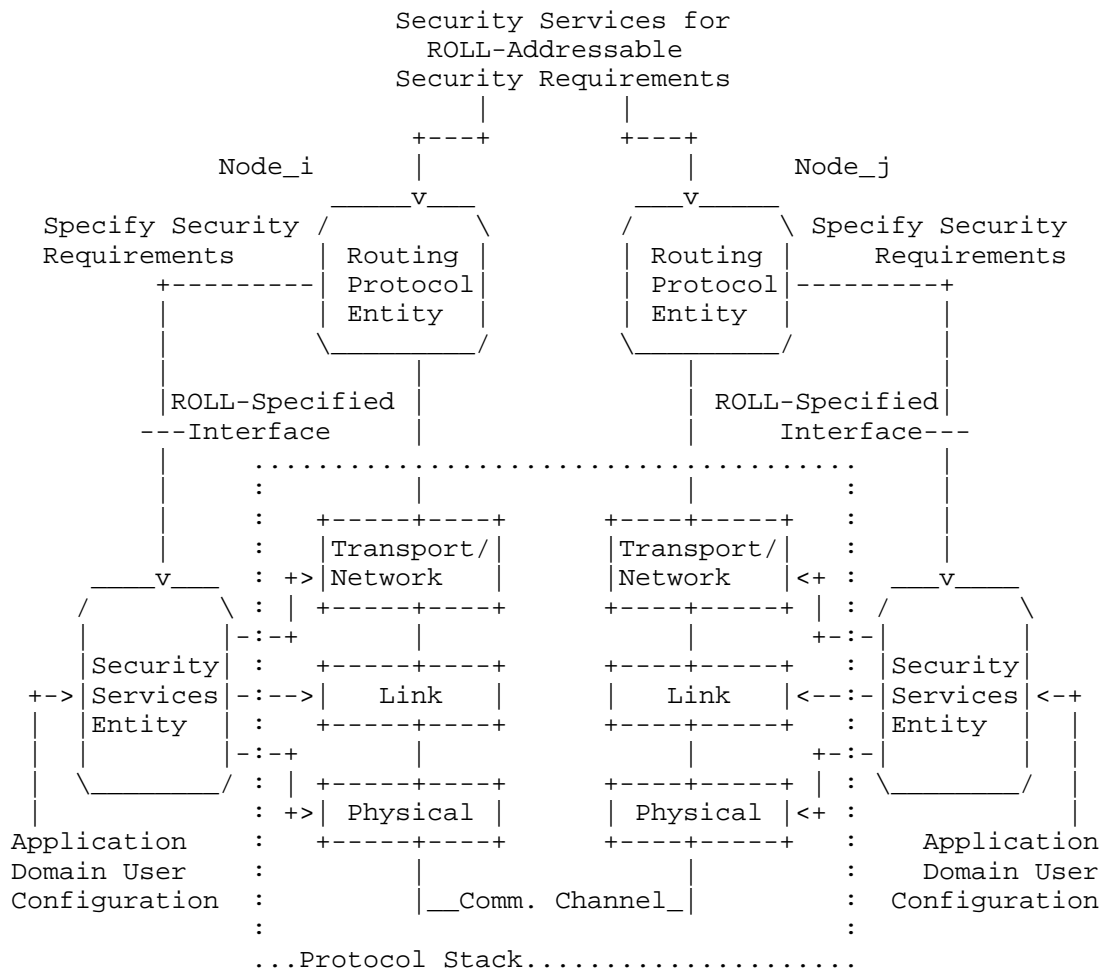


Figure 3: LLN Device Security Model

7. IANA Considerations

This memo includes no request to IANA.

8. Security Considerations

The analysis presented in this document provides security analysis and design guidelines with a scope limited to ROLL. Security services are identified as requirements for securing ROLL. The

specific mechanisms to be used to deal with each threat is specified in link-layer and deployment specific applicability statements.

9. Acknowledgments

The authors would like to acknowledge the review and comments from Rene Struik and JP Vasseur. The authors would also like to acknowledge the guidance and input provided by the ROLL Chairs, David Culler, and JP Vasseur, and the Area Director Adrian Farrel.

This document started out as a combined threat and solutions document, but was split up by ROLL co-Chair Michael Richardson as it went through the IETF publication process.

10. References

10.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC4107] Bellovin, S. and R. Housley, "Guidelines for Cryptographic Key Management", BCP 107, RFC 4107, June 2005.
- [RFC4301] Kent, S. and K. Seo, "Security Architecture for the Internet Protocol", RFC 4301, December 2005.
- [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.

10.2. Informative References

- [FIPS197] "Federal Information Processing Standards Publication 197: Advanced Encryption Standard (AES)", US National Institute of Standards and Technology, Nov. 26 2001.
- [Huang2003] Huang, Q., Cukier, J., Kobayashi, H., Liu, B., and J. Zhang, "Fast Authenticated Key Establishment Protocols for Self-Organizing Sensor Networks", in Proceedings of the 2nd ACM International Conference on Wireless Sensor Networks and Applications, San Diego, CA, USA, pp. 141-150, Sept. 19 2003.

- [I-D.alexander-roll-mikey-lln-key-mgmt]
Alexander, R. and T. Tsao, "Adapted Multimedia Internet KEYing (AMIKEY): An extension of Multimedia Internet KEYing (MIKEY) Methods for Generic LLN Environments", draft-alexander-roll-mikey-lln-key-mgmt-04 (work in progress), September 2012.
- [I-D.suhopark-hello-wsn]
Park, S., "Routing Security in Sensor Network: HELLO Flood Attack and Defense", draft-suhopark-hello-wsn-00 (work in progress), December 2005.
- [IEEE1149.1]
"IEEE Standard Test Access Port and Boundary Scan Architecture", IEEE-SA Standards Board, Jun. 14 2001.
- [Karlof2003]
Karlof, C. and D. Wagner, "Secure routing in wireless sensor networks: attacks and countermeasures", Elsevier AdHoc Networks Journal, Special Issue on Sensor Network Applications and Protocols, 1(2):293-315, September 2003.
- [Kasumi3gpp]
"3GPP TS 35.202 Specification of the 3GPP confidentiality and integrity algorithms; Document 2: Kasumi specification", 3GPP TSG SA3, 2009.
- [Messerges2003]
Messerges, T., Cukier, J., Kevenaar, T., Puhl, L., Struik, R., and E. Callaway, "Low-Power Security for Wireless Sensor Networks", in Proceedings of the 1st ACM Workshop on Security of Ad Hoc and Sensor Networks, Fairfax, VA, USA, pp. 1-11, Oct. 31 2003.
- [Myagmar2005]
Myagmar, S., Lee, A.J., and W. Yurcik, "Threat Modeling as a Basis for Security Requirements", in Proceedings of the Symposium on Requirements Engineering for Information Security (SREIS'05), Paris, France, pp. 94-102, Aug 29, 2005.
- [Perlman1988]
Perlman, N., "Network Layer Protocols with Byzantine Robustness", MIT LCS Tech Report, 429, 1988.
- [RFC1142]
Oran, D., "OSI IS-IS Intra-domain Routing Protocol", RFC 1142, February 1990.

- [RFC2080] Malkin, G. and R. Minnear, "RIPng for IPv6", RFC 2080, January 1997.
- [RFC2328] Moy, J., "OSPF Version 2", STD 54, RFC 2328, April 1998.
- [RFC2453] Malkin, G., "RIP Version 2", STD 56, RFC 2453, November 1998.
- [RFC3029] Adams, C., Sylvester, P., Zolotarev, M., and R. Zuccherato, "Internet X.509 Public Key Infrastructure Data Validation and Certification Server Protocols", RFC 3029, February 2001.
- [RFC3693] Cuellar, J., Morris, J., Mulligan, D., Peterson, J., and J. Polk, "Geopriv Requirements", RFC 3693, February 2004.
- [RFC3830] Arkko, J., Carrara, E., Lindholm, F., Naslund, M., and K. Norrman, "MIKEY: Multimedia Internet KEYing", RFC 3830, August 2004.
- [RFC4046] Baugher, M., Canetti, R., Dondeti, L., and F. Lindholm, "Multicast Security (MSEC) Group Key Management Architecture", RFC 4046, April 2005.
- [RFC4593] Barbir, A., Murphy, S., and Y. Yang, "Generic Threats to Routing Protocols", RFC 4593, October 2006.
- [RFC4732] Handley, M., Rescorla, E., and IAB, "Internet Denial-of-Service Considerations", RFC 4732, December 2006.
- [RFC4949] Shirey, R., "Internet Security Glossary, Version 2", RFC 4949, August 2007.
- [RFC5197] Fries, S. and D. Ignjatic, "On the Applicability of Various Multimedia Internet KEYing (MIKEY) Modes and Extensions", RFC 5197, June 2008.
- [RFC5548] Dohler, M., Watteyne, T., Winter, T., and D. Barthel, "Routing Requirements for Urban Low-Power and Lossy Networks", RFC 5548, May 2009.
- [RFC5673] Pister, K., Thubert, P., Dwars, S., and T. Phinney, "Industrial Routing Requirements in Low-Power and Lossy Networks", RFC 5673, October 2009.
- [RFC5751] Ramsdell, B. and S. Turner, "Secure/Multipurpose Internet Mail Extensions (S/MIME) Version 3.2 Message Specification", RFC 5751, January 2010.

- [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. Vermeulen, "Building Automation Routing Requirements in Low-Power and Lossy Networks", RFC 5867, June 2010.
- [RFC5996] Kaufman, C., Hoffman, P., Nir, Y., and P. Eronen, "Internet Key Exchange Protocol Version 2 (IKEv2)", RFC 5996, September 2010.
- [Wan2004] Wan, T., Kranakis, E., and P.C. van Oorschot, "S-RIP: A Secure Distance Vector Routing Protocol", in Proceedings of the 2nd International Conference on Applied Cryptography and Network Security, Yellow Mountain, China, pp. 103-119, Jun. 8-11 2004.
- [Wander2005] Wander, A., Gura, N., Eberle, H., Gupta, V., and S. Shantz, "Energy analysis of public-key cryptography for wireless sensor network", in the Proceedings of the Third IEEE International Conference on Pervasive Computing and Communications pp. 324-328, March 8-12 2005.
- [Yourdon1979] Yourdon, E. and L. Constantine, "Structured Design", Yourdon Press, New York, Chapter 10, pp. 187-222, 1979.

Authors' Addresses

Tzeta Tsao
Cooper Power Systems
910 Clopper Rd. Suite 201S
Gaithersburg, Maryland 20878
USA

Email: tzeta.tsao@cooperindustries.com

Roger K. Alexander
Cooper Power Systems
910 Clopper Rd. Suite 201S
Gaithersburg, Maryland 20878
USA

Email: roger.alexander@cooperindustries.com

Mischa Dohler
CTTC
Parc Mediterrani de la Tecnologia, Av. Canal Olímpic S/N
Castelldefels, Barcelona 08860
Spain

Email: mischa.dohler@cttc.es

Vanesa Daza
Universitat Pompeu Fabra
P/ Circumval.lacio 8, Oficina 308
Barcelona 08003
Spain

Email: vanesa.daza@upf.edu

Angel Lozano
Universitat Pompeu Fabra
P/ Circumval.lacio 8, Oficina 309
Barcelona 08003
Spain

Email: angel.lozano@upf.edu

Networking Working Group
Internet-Draft
Intended status: Informational
Expires: August 20, 2013

JP. Vasseur
Cisco Systems, Inc
February 16, 2013

Terminology in Low power And Lossy Networks
draft-ietf-roll-terminology-11.txt

Abstract

The documents defines a terminology for discussing routing requirements and solutions for networks referred to as Low power and Lossy Networks (LLN). A LLN is typically composed of many embedded devices with limited power, memory, and processing resources interconnected by a variety of links. There is a wide scope of application areas for LLNs, including industrial monitoring, building automation (e.g. Heating, Ventilating, Air Conditioning, lighting, access control, fire), connected home, healthcare, environmental monitoring, urban sensor networks, energy management, assets tracking, refrigeration.

Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

Status of this Memo

This Internet-Draft is submitted to IETF in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

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

This Internet-Draft will expire on August 20, 2013.

Copyright Notice

Copyright (c) 2013 IETF Trust and the persons identified as the

document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	3
2. Terminology	3
3. IANA Considerations	7
4. Security Considerations	7
5. Acknowledgements	7
6. References	7
6.1. Normative References	7
6.2. Informative References	7
Author's Address	8

1. Introduction

This document defines a terminology for discussing routing requirements and solutions for networks referred to as Low power and Lossy Networks (LLN).

Low power and Lossy networks (LLNs) are typically composed of many embedded devices with limited power, memory, and processing resources interconnected by a variety of links, such as IEEE 802.15.4, Low Power WiFi. There is a wide scope of application areas for LLNs, including industrial monitoring, building automation (HVAC, lighting, access control, fire), connected home, healthcare, environmental monitoring, urban sensor networks, energy management, assets tracking and refrigeration.

Since these applications are usually highly specific (for example Industrial Automation, Building Automation, ...), it is not uncommon to see a number of disparate terms to describe the same device or functionality. Thus in order to avoid confusion or discrepancies, this document specifies the common terminology to be used in all ROLL Working Group documents. The terms defined in this document are used in [RFC5548], [RFC5673], [RFC5826] and [RFC5867].

Terminology specific to a particular application are out of the scope of this document.

It is expected that all routing requirements documents defining requirements or specifying routing solutions for LLN will use the common terminology specified in this document. This document should be listed as an informative reference.

2. Terminology

Actuator: a field device that controls a set of equipment. For example, an actuator might control and/or modulates the flow of a gas or liquid, control electricity distribution, perform a mechanical operation, ...

AMI: Advanced Metering Infrastructure that makes use of Smart Grid technologies. A canonical Smart Grid application is smart-metering.

Channel: Radio frequency sub-band used to transmit a modulated signal carrying packets.

Channel Hopping: A procedure by which field devices synchronously change channels during operation.

Commissioning Tool: Any physical or logical device temporarily added to the network for the expressed purpose of setting up the network and device operational parameters. The commissioning tool can also be temporarily added to the LLN for scheduled or unscheduled maintenance.

Closed Loop Control: A procedure whereby a device controller controls an actuator based on input information sensed by one or more field devices.

Controller: A field device that can receive sensor input and automatically change the environment in the facility by manipulating digital or analog actuators.

DA: Distribution Automation, part of Smart Grid. Encompasses technologies for maintenance and management of electrical distribution systems.

Data sink: A device that collects data from nodes in a LLN.

Downstream: Data direction traveling from outside of the LLN (e.g. traffic coming from a LAN, WAN or the Internet) via a LBR.

Field Device: A field device is a physical device placed in the network's operating environment (e.g. plant, urban or home). Field devices include sensors, actuators as well as routers and Low power and Lossy Network Border Router (including LBR). A field device is usually (but not always) a device with constrained CPU, memory footprint, storage capacity, bandwidth and sometimes power (battery operated). At the time of writing, for the sake of illustration, a typical sensor or actuator would have a few KBytes of RAM, a few dozens of KBytes of ROM/Flash memory, a 8/16/32 bit microcontroller and communication capabilities ranging from a few Kbits/s to a few hundreds of Kbits/s. Although it is expected to see continuous improvements of hardware and software technologies, such devices will likely continue to be seen as resource constrained devices compared to computers and routers used in the Internet.

Flash memory: non-volatile memory that can be re-programmed.

FMS: Facility Management System. A global term applied across all the vertical designations within a building including, Heating, Ventilating, and Air Conditioning also referred to as HVAC, Fire, Security, Lighting and Elevator control.

HART: "Highway Addressable Remote Transducer", a group of specifications for industrial process and control devices administered by the HART Foundation (see [HART]). The latest version

for the specifications is HART7 which includes the additions for WirelessHART.

HVAC: Heating, Ventilation and Air Conditioning. A term applied to the comfort level of an internal space.

ISA: "International Society of Automation". ISA is an ANSI accredited standards-making society. ISA100 is an ISA committee whose charter includes defining a family of standards for industrial automation. [ISA100.11a] is a working group within ISA100 that is working on a standard for monitoring and non-critical process control applications.

LAN: Local Area Network.

LBR: Low power and lossy network Border Router. The LBR is a device that connects the Low power and Lossy Network to another routing domain such as a Local Area Network (LAN), Wide Area Network (WAN) or the Internet where a possibly different routing protocol is in operation. The LBR acts as a routing device and may possibly host other functions such as data collector or aggregator.

LLN: Low power and Lossy networks (LLNs) are typically composed of many embedded devices with limited power, memory, and processing resources interconnected by a variety of links, such as IEEE 802.15.4 or Low Power WiFi. There is a wide scope of application areas for LLNs, including industrial monitoring, building automation (HVAC, lighting, access control, fire), connected home, healthcare, environmental monitoring, urban sensor networks, energy management, assets tracking and refrigeration..

MP2P: Multipoint-to-Point is used to describe a particular traffic pattern (e.g. MP2P flows collecting information from many nodes flowing inwards towards a collecting sink or an LBR).

MAC: Medium Access Control. Refers to algorithms and procedures used by the data link layer to coordinate use of the physical layer.

Non-sleepy Node: A non-sleepy node is a node that always remains in a fully powered on state (i.e. always awake) where it has the capability to perform RPL protocol communication.

Open Loop Control: A process whereby a plant operator manually manipulates an actuator over the network where the decision is influenced by information sensed by field devices.

PER: Packet Error Rate. A ratio of the number of unusable packets (not received at all, or received in error- even after any applicable

error correction has been applied) to the total number of packets that would have been received in the absence of errors.

P2P: Point To Point. This refers to traffic exchanged between two nodes (regardless of the number of hops between the two nodes).

P2MP: Point-to-Multipoint traffic refers to traffic between one node and a set of nodes. This is similar to the P2MP concept in Multicast or MPLS Traffic Engineering ([RFC4461]and [RFC4875]). A common RPL use case involves P2MP flows from or through a DAG root outward towards other nodes contained in the DAG.

RAM: Random Access Memory. The RAM is a volatile memory.

RFID: Radio Frequency IDentification.

ROM: Read Only Memory.

ROLL: Routing Over Low power and Lossy networks.

RPL Domain: A RPL routing domain is a collection of RPL routers under the control of a single administration. The boundaries of routing domains are defined by network management by setting some links to be exterior, or inter-domain, links.

Schedule: An agreed execution, wake-up, transmission, reception, etc., time-table between two or more field devices.

Sensor: A sensor is a device that measures a physical quantity and converts it to a analog or digital signal that can be read by a program or a user. Sensed data can be of many types: electromagnetic (e.g. current, voltage, power, resistance, ...) , mechanical (e.g. pressure, flow, liquid density, humidity, ...), chemical (e.g. oxygen, carbon monoxide, ...), acoustic (e.g. noise, ultrasound), ...

Sleepy Node: A sleepy node is a node that may sometimes go into a sleep mode (i.e. go into a low power state to conserve power) and temporarily suspend protocol communication. A sleepy node may also sometimes remain in a fully powered on state where it has the capability to perform RPL protocol communication.

Smart Grid: A Smart Grid is a broad class of applications to network and automate utility infrastructure.

Timeslot: A Timeslot is a fixed time interval that may be used for the transmission or reception of a packet between two field devices. A timeslot used for communications is associated with a slotted-link

Upstream: Data direction traveling from the LLN via the LBR to outside of the LLN (LAN, WAN, Internet).

WAN: Wide Area Network.

3. IANA Considerations

This document includes no request for IANA action.

4. Security Considerations

Since this document specifies terminology and does not specify new procedure or protocols, it raises no new security issue.

5. Acknowledgements

The authors would like to thank Christian Jacquenet, Tim Winter, Pieter De Mil, David Meyer, Mukul Goyal and Abdussalam Baryun for their valuable feed-back.

6. References

6.1. Normative References

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

6.2. Informative References

- [HART] HART Communication Foundation (<http://www.hartcomm.org>)
- [RFC4461] Yasukawa, S., "Signaling Requirements for Point-to-Multipoint Traffic-Engineered MPLS Label Switched Paths (LSPs)", RFC 4461, April 2006.
- [RFC4875] Aggarwal, R., Papadimitriou, D., and S. Yasukawa, "Extensions to Resource Reservation Protocol - Traffic Engineering (RSVP-TE) for Point-to-Multipoint TE Label Switched Paths (LSPs)", RFC 4875, May 2007.
- [RFC5548] Dohler, M., Watteyne, T., Winter, T., and D. Barthel, "Routing Requirements for Urban Low-Power and Lossy Networks", RFC 5548, May 2009.
- [RFC5673] Pister, K., Thubert, P., Dwars, S., and T. Phinney, "Industrial Routing Requirements in Low-Power and Lossy

Networks", RFC 5673, October 2009.

[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.

Author's Address

JP Vasseur
Cisco Systems, Inc
1414 Massachusetts Avenue
Boxborough, MA 01719
USA

Email: jpv@cisco.com

ROLL
Internet-Draft
Intended status: Standards Track
Expires: August 29, 2013

J. Hui
Cisco
R. Kelsey
Silicon Labs
February 25, 2013

Multicast Protocol for Low power and Lossy Networks (MPL)
draft-ietf-roll-trickle-mcast-04

Abstract

This document specifies the Multicast Protocol for Low power and Lossy Networks (MPL) that provides IPv6 multicast forwarding in constrained networks. MPL avoids the need to construct or maintain any multicast forwarding topology, disseminating messages to all MPL Forwarders in an MPL Domain. MPL uses the Trickle algorithm to manage message transmissions for both control and data-plane messages. Different Trickle parameter configurations allow MPL to trade between dissemination latency and transmission efficiency.

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

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

This Internet-Draft will expire on August 29, 2013.

Copyright Notice

Copyright (c) 2013 IETF Trust and the persons identified as the document authors. All rights reserved.

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

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

Table of Contents

1. Introduction	3
2. Terminology	4
3. Applicability Statement	6
4. Protocol Overview	7
4.1. Information Base Overview	7
4.2. Overview	7
4.3. Signaling Overview	9
5. MPL Parameters and Constants	10
5.1. MPL Multicast Addresses	10
5.2. MPL Message Types	10
5.3. MPL Seed Identifiers	10
5.4. MPL Forwarder Parameters	10
5.5. MPL Trickle Parameters	11
6. Protocol Message Formats	13
6.1. MPL Option	13
6.2. MPL Control Message	14
6.3. MPL Seed Info	15
7. Information Base	17
7.1. Local Interface Set	17
7.2. Domain Set	17
7.3. Seed Set	17
7.4. Buffered Message Set	17
8. MPL Domains	19
9. MPL Seed Sequence Numbers	20
10. MPL Data Messages	21
10.1. MPL Data Message Generation	21
10.2. MPL Data Message Transmission	21
10.3. MPL Data Message Processing	22
11. MPL Control Messages	24
11.1. MPL Control Message Generation	24
11.2. MPL Control Message Transmission	24
11.3. MPL Control Message Processing	25
12. Acknowledgements	27
13. IANA Considerations	28
13.1. MPL Option Type	28
13.2. MPL ICMPv6 Type	28
13.3. Well-known Multicast Addresses	28
14. Security Considerations	29
15. Normative References	30
Authors' Addresses	31

1. Introduction

Low power and Lossy Networks typically operate with strict resource constraints in communication, computation, memory, and energy. Such resource constraints may preclude the use of existing IPv6 multicast routing and forwarding mechanisms. Traditional IP multicast delivery typically relies on topology maintenance mechanisms to discover and maintain routes to all subscribers of a multicast group. However, maintaining such topologies in LLNs is costly and may not be feasible given the available resources.

Memory constraints may limit devices to maintaining links/routes to one or a few neighbors. For this reason, the Routing Protocol for LLNs (RPL) specifies both storing and non-storing modes [RFC6550]. The latter allows RPL routers to maintain only one or a few default routes towards a LLN Border Router (LBR) and use source routing to forward messages away from the LBR. For the same reasons, a LLN device may not be able to maintain a multicast routing topology when operating with limited memory.

Furthermore, the dynamic properties of wireless networks can make the cost of maintaining a multicast routing topology prohibitively expensive. In wireless environments, topology maintenance may involve selecting a connected dominating set used to forward multicast messages to all nodes in an administrative domain. However, existing mechanisms often require two-hop topology information and the cost of maintaining such information grows polynomially with network density.

This document specifies the Multicast Protocol for Low power and Lossy Networks (MPL), which provides IPv6 multicast forwarding in constrained networks. MPL avoids the need to construct or maintain any multicast routing topology, disseminating multicast messages to all MPL Forwarders in an MPL Domain. By using the Trickle algorithm [RFC6206], MPL requires only small, constant state for each MPL device that initiates disseminations. The Trickle algorithm also allows MPL to be density-aware, allowing the communication rate to scale logarithmically with density.

2. Terminology

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

The following terms are used throughout this document:

MPL Forwarder	- A router that implements this protocol. An MPL Forwarder is equipped with at least one MPL Interface.
MPL Interface	- An MPL Forwarder's attachment to a communications medium, over which it transmits and receives MPL Data Messages and MPL Control Messages according to this specification. An MPL Interface is assigned one or more unicast addresses and is subscribed to one or more MPL Domain Addresses.
MPL Domain Address	- A multicast address that identifies the set of MPL Interfaces within an MPL Domain. MPL Data Messages disseminated in an MPL Domain have the associated MPL Domain Address as their destination address.
MPL Domain	- A scope zone, as defined in [RFC4007], in which MPL Interfaces subscribe to the same MPL Domain Address and participate in disseminating MPL Data Messages.
MPL Data Message	- A multicast message that is used to communicate a multicast payload between MPL Forwarders within an MPL domain. An MPL Data Message contains an MPL Option in the IPv6 header and has as its destination address the MPL Domain Address corresponding to the MPL Domain.
MPL Control Message	- A link-local multicast message that is used to communicate information about recently received MPL Data Messages to neighboring MPL Forwarders.
MPL Seed	- An MPL Forwarder that generates MPL Data Messages and serves as an entry point into an MPL Domain.

MPL Seed Identifier - An unsigned integer that uniquely identifies an MPL Seed within an MPL Domain.

3. Applicability Statement

This protocol is an IPv6 multicast forwarding protocol for Low-Power and Lossy Networks. By implementing a controlled dissemination using the Trickle algorithm, this protocol is designed for networks that communicate using low-power and lossy links with widely varying topologies in both the space and time dimensions.

4. Protocol Overview

The goal of MPL is to deliver multicast messages to all interfaces that subscribe to the multicast messages' destination address within an MPL Domain.

4.1. Information Base Overview

A node records necessary protocol state in the following information sets:

- o The Local Interface Set records the set of local MPL Interfaces and the unicast addresses assigned to those MPL Interfaces.
- o The Domain Set records the set of MPL Domain Addresses and the local MPL Interfaces that subscribe to those addresses.
- o A Seed Set records information about received MPL Data Messages received from an MPL Seed within an MPL Domain. Each MPL Domain has an associated Seed Set. A Seed Set maintains the minimum sequence number for MPL Data Messages that the MPL Forwarder is willing to receive or has buffered in its Buffered Message Set from an MPL Seed. MPL uses Seed Sets and Buffered Message Sets to determine when to accept an MPL Data Message, process its payload, and retransmit it.
- o A Buffered Message Set records recently received MPL Data Messages from an MPL Seed within an MPL Domain. Each MPL Domain has an associated Buffered Message Set. MPL Data Messages resident in a Buffered Message Set have sequence numbers that are greater than or equal to the minimum threshold maintained in the corresponding Seed Set. MPL uses Buffered Message Sets to store MPL Data Messages that may be transmitted by the MPL Forwarder for forwarding.

4.2. Overview

MPL achieves its goal by implementing a controlled flood that attempts to disseminate the multicast data message to all interfaces within an MPL Domain. MPL performs the following tasks to disseminate a multicast message:

- o When having a multicast message to forward into an MPL Domain, the MPL Seed generates an MPL Data Message that includes the MPL Domain Address as the IPv6 Destination Address, the MPL Seed Identifier, a newly generated sequence number, and the multicast message. If the multicast destination address is not the MPL Domain Address, IP-in-IP [RFC2473] is used to encapsulate the

multicast message in an MPL Data Message, preserving the original IPv6 Destination Address.

- o Upon receiving an MPL Data Message, the MPL Forwarder extracts the MPL Seed and sequence number and determines whether or not the MPL Data Message was previously received using the MPL Domain's Seed Set and Buffered Message Set.
 - * If the sequence number is less than the lower-bound sequence number maintained in the Seed Set or a message with the same sequence number exists within the Buffered Message Set, the MPL Forwarder marks the MPL Data Message as old.
 - * Otherwise, the MPL Forwarder marks the MPL Data Message as new.
- o For each newly received MPL Data Message, an MPL Forwarder updates the Seed Set, adds the MPL Data Message into the Buffered Message Set, processes its payload, and multicasts the MPL Data Message a number of times on all MPL Interfaces participating in the same MPL Domain to forward the message.
- o Each MPL Forwarder may periodically link-local multicast MPL Control Messages on MPL Interfaces to communicate information contained in an MPL Domain's Seed Set and Buffered Message Set.
- o Upon receiving an MPL Control Message, an MPL Forwarder determines whether there are any new MPL Data Messages that have yet to be received by the MPL Control Message's source and multicasts those MPL Data Messages.

MPL's configuration parameters allow two forwarding strategies for disseminating MPL Data Messages.

Proactive Forwarding - With proactive forwarding, an MPL Forwarder schedules transmissions of MPL Data Messages using the Trickle algorithm, without any prior indication that neighboring nodes have yet to receive the message. After transmitting the MPL Data Message a limited number of times, the MPL Forwarder may terminate proactive forwarding for the MPL Data Message message.

Reactive Forwarding - With reactive forwarding, an MPL Forwarder link-local multicasts MPL Control Messages using the Trickle algorithm [RFC6206]. MPL Forwarders use MPL Control Messages to discover new MPL Data Messages that have not yet been received. When discovering that a neighboring MPL Forwarder has not yet received an MPL Data Message, the MPL Forwarder schedules those MPL Data Messages for transmission using the Trickle algorithm.

4.3. Signaling Overview

This protocol generates and processes the following messages:

MPL Data Message - Generated by an MPL Seed to deliver a multicast message across an MPL Domain. The MPL Data Message's source is an address in the Local Interface Set of the MPL Seed that generated the message and is valid within the MPL Domain. The MPL Data Message's destination is the MPL Domain Address corresponding to the MPL Domain. An MPL Data Message contains:

- * The Seed Identifier of the MPL Seed that generated the MPL Data Message.
- * The sequence number of the MPL Seed that generated the MPL Data Message.
- * The original multicast message.

MPL Control Message - Generated by an MPL Forwarder to communicate information contained in an MPL Domain's Seed Set and Buffered Message Set to neighboring MPL Forwarders. An MPL Control Message contains a list of tuples for each entry in the Seed Set. Each tuple contains:

- * The minimum sequence number maintained in the Seed Set for the MPL Seed.
- * A bit-vector indicating the sequence numbers of MPL Data Messages resident in the Buffered Message Set for the MPL Seed, where the first bit represents a sequence number equal to the minimum threshold maintained in the Seed Set.
- * The length of the bit-vector.

5. MPL Parameters and Constants

This section describes various program and networking parameters and constants used by MPL.

5.1. MPL Multicast Addresses

MPL makes use of MPL Domain Addresses to identify MPL Interfaces of an MPL Domain. By default, MPL Forwarders subscribe to the `ALL_MPL_FORWARDERS` multicast address with a scope value of 3 (subnet-local).

For each MPL Domain Address that an MPL Interface subscribes to, the MPL Interface **MUST** also subscribe to the MPL Domain Address with a scope value of 2 (link-local) when reactive forwarding is in use. MPL Forwarders use the link-scoped MPL Domain Address to communicate MPL Control Messages to neighboring (i.e. on-link) MPL Forwarders.

5.2. MPL Message Types

MPL defines an IPv6 Option for carrying an MPL Seed Identifier and a sequence number within an MPL Data Message. The IPv6 Option Type has value `MPL_OPT_TYPE`.

MPL defines an ICMPv6 Message (MPL Control Message) for communicating information contained in an MPL Domain's Seed Set and Buffered Message Set to neighboring MPL Forwarders. The MPL Control Message has ICMPv6 Type `MPL_ICMP_TYPE`.

5.3. MPL Seed Identifiers

MPL uses MPL Seed Identifiers to uniquely identify MPL Seeds within an MPL Domain. For each MPL Domain that the MPL Forwarder serves as an MPL Seed, the MPL Forwarder **MUST** have an associated MPL Seed Identifier. An MPL Forwarder **MAY** use the same MPL Seed Identifier across multiple MPL Domains, but the MPL Seed Identifier **MUST** be unique within each MPL Domain. The mechanism for assigning and verifying uniqueness of MPL Seed Identifiers is not specified in this document.

5.4. MPL Forwarder Parameters

`PROACTIVE_FORWARDING` A boolean value that indicates whether the MPL Forwarder should schedule MPL Data Message transmissions after receiving them for the first time. `PROACTIVE_FORWARDING` has a default value of `TRUE`.

SEED_SET_ENTRY_LIFETIME The minimum lifetime for an entry in the Seed Set. **SEED_SET_ENTRY_LIFETIME** has a default value of 30 minutes.

It is RECOMMENDED that all MPL Forwarders use the same values for the MPL Forwarder Parameters above for a given MPL Domain. The mechanism for setting the MPL Forwarder Parameters is not specified within this document.

5.5. MPL Trickle Parameters

As specified in [RFC6206], a Trickle timer runs for a defined interval and has three configuration parameters: the minimum interval size *Imin*, the maximum interval size *Imax*, and a redundancy constant *k*.

This specification defines a fourth Trickle configuration parameter, *TimerExpirations*, which indicates the number of Trickle timer expiration events that occur before terminating the Trickle algorithm.

Each MPL Forwarder uses the following Trickle parameters for MPL Data Message and MPL Control Message transmissions.

DATA_MESSAGE_IMIN The minimum Trickle timer interval, as defined in [RFC6206], for MPL Data Message transmissions. **DATA_MESSAGE_IMIN** has a default value of 10 times the worst-case link-layer latency.

DATA_MESSAGE_IMAX The maximum Trickle timer interval, as defined in [RFC6206], for MPL Data Message transmissions. **DATA_MESSAGE_IMAX** has a default value equal to **DATA_MESSAGE_IMIN**.

DATA_MESSAGE_K The redundancy constant, as defined in [RFC6206], for MPL Data Message transmissions. **DATA_MESSAGE_K** has a default value of 5.

DATA_MESSAGE_TIMER_EXPIRATIONS The number of Trickle timer expirations that occur before terminating the Trickle algorithm for MPL Data Message transmissions. **DATA_MESSAGE_TIMER_EXPIRATIONS** has a default value of 3.

CONTROL_MESSAGE_IMIN The minimum Trickle timer interval, as defined in [RFC6206], for MPL Control Message transmissions. **CONTROL_MESSAGE_IMIN** has a default value of 10 times the worst-case link-layer latency.

CONTROL_MESSAGE_IMAX The maximum Trickle timer interval, as defined in [RFC6206], for MPL Control Message transmissions. CONTROL_MESSAGE_IMAX has a default value of 5 minutes.

CONTROL_MESSAGE_K The redundancy constant, as defined in [RFC6206], for MPL Control Message transmissions. CONTROL_MESSAGE_K has a default value of 1.

CONTROL_MESSAGE_TIMER_EXPIRATIONS The number of Trickle timer expirations that occur before terminating the Trickle algorithm for MPL Control Message transmissions. CONTROL_MESSAGE_TIMER_EXPIRATIONS has a default value of 10.

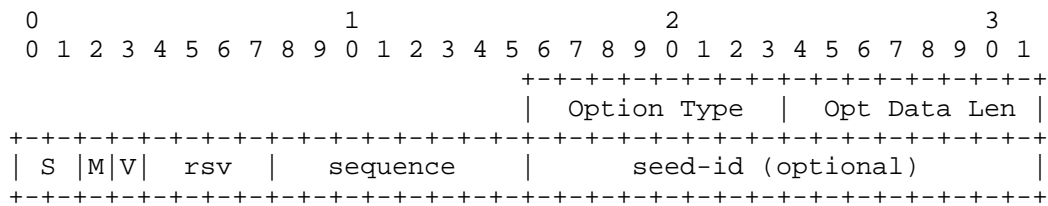
Following [RFC6206], it is RECOMMENDED that all MPL Forwarders use the same values for the Trickle Parameters above for a given MPL Domain. The mechanism for setting the Trickle Parameters is not specified within this document.

6. Protocol Message Formats

The protocol messages generated and processed by an MPL Forwarder are described in this section.

6.1. MPL Option

The MPL Option is carried in MPL Data Messages in an IPv6 Hop-by-Hop Options header, immediately following the IPv6 header. The MPL Option has the following format:



Option Type	MPL_OPT_TYPE
Opt Data Len	Length of the Option Data field in octets.
S	2-bit unsigned integer. Identifies the length of seed-id. 0 indicates that the seed-id is the IPv6 Source Address and not included in the MPL Option. 1 indicates that the seed-id is a 16-bit unsigned integer. 2 indicates that the seed-id is a 64-bit unsigned integer. 3 indicates that the seed-id is a 128-bit unsigned integer.
M	1-bit flag. 1 indicates that the value in sequence is known to be the largest sequence number that was received from the MPL Seed.
V	1-bit flag. 0 indicates that the MPL Option conforms to this specification. MPL Data Messages with an MPL Option in which this flag is 1 MUST be dropped.
rsv	4-bit reserved field. MUST be set to 0 on transmission and ignored on reception.
sequence	8-bit unsigned integer. Identifies relative ordering of MPL Data Messages from the MPL Seed identified by seed-id.

seed-id Uniquely identifies the MPL Seed that initiated dissemination of the MPL Data Message. The size of seed-id is indicated by the S field.

The Option Data (in particular the M flag) of the MPL Option is updated by MPL Forwarders as the MPL Data Message is forwarded. Nodes that do not understand the MPL Option MUST discard the MPL Data Message. Thus, according to [RFC2460] the three high order bits of the Option Type are set to '011'. The Option Data length is variable.

The seed-id uniquely identifies an MPL Seed. When seed-id is 128 bits (S=3), the MPL seed MAY use an IPv6 address assigned to one of its interfaces that is unique within the MPL Domain. Managing MPL Seed Identifiers is not within scope of this document.

The sequence field establishes a total ordering of MPL Data Messages generated by an MPL Seed for an MPL Domain. The MPL Seed MUST increment the sequence field's value on each new MPL Data Message that it generates for an MPL Domain. Implementations MUST follow the Serial Number Arithmetic as defined in [RFC1982] when incrementing a sequence value or comparing two sequence values.

Future updates to this specification may define additional fields following the seed-id field.

6.2. MPL Control Message

An MPL Forwarder uses ICMPv6 messages to communicate information contained in an MPL Domain's Seed Set and Buffered Message Set to neighboring MPL Forwarders. The MPL Control Message has the following format:

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								
Type										Code										Checksum																			
MPL Seed Info[1..n]																																							

IP Fields:

Source Address An IPv6 address in the AddressSet of the corresponding MPL Interface and MUST be valid within the MPL Domain.

Destination Address The link-scoped MPL Domain Address corresponding to the MPL Domain.

Hop Limit 255

ICMPv6 Fields:

Type MPL_ICMP_TYPE

Code 0

Checksum The ICMP checksum. See [RFC4443].

MPL Seed Info[0..n] List of zero or more MPL Seed Info entries.

The MPL Control Message indicates the sequence numbers of MPL Data Messages that are within the MPL Domain's Buffered Message Set. The MPL Control Message also indicates the sequence numbers of MPL Data Messages that an MPL Forwarder is willing to receive. The MPL Control Message allows neighboring MPL Forwarders to determine whether there are any new MPL Data Messages to exchange.

6.3. MPL Seed Info

An MPL Seed Info encodes the minimum sequence number for an MPL Seed maintained in the MPL Domain's Seed Set. The MPL Seed Info also indicates the sequence numbers of MPL Data Messages generated by the MPL Seed that are stored within the MPL Domain's Buffered Message Set. The MPL Seed Info has the following format:

```

      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
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| min-seqno | bm-len | S | seed-id (0/2/8/16 octets) |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|
| buffered-mpl-messages (variable length)
|
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

min-seqno	8-bit unsigned integer. The lower-bound sequence number for the MPL Seed.
bm-len	6-bit unsigned integer. The size of buffered-mpl-messages in octets.
S	2-bit unsigned integer. Identifies the length of seed-id. 0 indicates that the seed-id value is the IPv6 Source Address and not included in the MPL Seed Info. 1 indicates that the seed-id value is a 16-bit unsigned integer. 2 indicates that the seed-id value is a 64-bit unsigned integer. 3 indicates that the seed-id is a 128-bit unsigned integer.
seed-id	Variable-length unsigned integer. Indicates the MPL Seed associated with this MPL Seed Info.
buffered-mpl-messages	Variable-length bit vector. Identifies the sequence numbers of MPL Data Messages maintained in the corresponding Buffered Message Set for the MPL Seed. The i'th bit represents a sequence number of min-seqno + i. '0' indicates that the corresponding MPL Data Message does not exist in the Buffered Message Set. '1' indicates that the corresponding MPL Data Message does exist in the Buffered Message Set.

The MPL Seed Info does not have any octet alignment requirement.

7. Information Base

7.1. Local Interface Set

The Local Interface Set records the local MPL Interfaces of an MPL Forwarder. The Local Interface Set consists of Local Interface Tuples, one per MPL Interface: (AddressSet).

AddressSet - a set of unicast addresses assigned to the MPL Interface.

7.2. Domain Set

The Domain Set records the MPL Interfaces that subscribe to each MPL Domain Address. The Domain Set consists of MPL Domain Tuples, one per MPL Domain: (MPLInterfaceSet).

MPLInterfaceSet - a set of MPL Interfaces that subscribe to the MPL Domain Address that identifies the MPL Domain.

7.3. Seed Set

A Seed Set records a sliding window used to determine the sequence numbers of MPL Data Messages that an MPL Forwarder is willing to accept generated by the MPL Seed. An MPL Forwarder maintains a Seed Set for each MPL Domain that it participates in. A Seed Set consists of MPL Seed Tuples: (SeedID, MinSequence, Lifetime).

SeedID - the identifier for the MPL Seed.

MinSequence - a lower-bound sequence number that represents the sequence number of the oldest MPL Data Message the MPL Forwarder is willing to receive or transmit. An MPL Forwarder MUST ignore any MPL Data Message that has sequence value less than than MinSequence.

Lifetime - indicates the minimum remaining lifetime of the Seed Set entry. An MPL Forwarder MUST NOT free a Seed Set entry before the remaining lifetime expires.

7.4. Buffered Message Set

A Buffered Message Set records recently received MPL Data Messages from an MPL Seed within an MPL Domain. An MPL Forwarder uses a Buffered Message Set to buffer MPL Data Messages while the MPL Forwarder is forwarding the MPL Data Messages. An MPL Forwarder maintains a Buffered Message Set for each MPL Domain that it participates in. A Buffered Message Set consists of Buffered Message

Tuples: (SeedID, SequenceNumber, DataMessage).

SeedID - the identifier for the MPL Seed that generated the MPL Data Message.

SequenceNumber - the sequence number for the MPL Data Message.

DataMessage - the MPL Data Message.

All MPL Data Messages within a Buffered Message Set MUST have a sequence number greater than or equal to MinSequence for the corresponding SeedID. When increasing MinSequence for an MPL Seed, the MPL Forwarder MUST delete any MPL Data Messages from the corresponding Buffered Message Set that have sequence numbers less than MinSequence.

8. MPL Domains

An MPL Domain is a scope zone, as defined in [RFC4007], in which MPL Interfaces subscribe to the same MPL Domain Address and participate in disseminating MPL Data Messages.

By default, an MPL Forwarder SHOULD participate in an MPL Domain identified by the ALL_MPL_FORWARDERS multicast address with a scope value of 3 (subnet-local).

An MPL Forwarder MAY participate in additional MPL Domains identified by other multicast addresses. An MPL Interface MUST subscribe to the MPL Domain Addresses for the MPL Domains that it participates in. The assignment of other multicast addresses is out of scope.

For each MPL Domain Address that an MPL Interface subscribes to, the MPL Interface MUST also subscribe to the same MPL Domain Address with a scope value of 2 (link-local) when reactive forwarding is in use (i.e. when communicating MPL Control Messages).

9. MPL Seed Sequence Numbers

Each MPL Seed maintains a sequence number for each MPL Domain that it serves. The sequence numbers are included in MPL Data Messages generated by the MPL Seed. The MPL Seed MUST increment the sequence number for each MPL Data Message that it generates for an MPL Domain. Implementations MUST follow the Serial Number Arithmetic as defined in [RFC1982] when incrementing a sequence value or comparing two sequence values. This sequence number is used to establish a total ordering of MPL Data Messages generated by an MPL Seed for an MPL Domain.

10. MPL Data Messages

10.1. MPL Data Message Generation

MPL Data Messages are generated by MPL Seeds when these messages enter the MPL Domain. All MPL Data messages have the following properties:

- o The IPv6 Source Address MUST be an address in the AddressSet of a corresponding MPL Interface and MUST be valid within the MPL Domain.
- o The IPv6 Destination Address MUST be set to the MPL Domain Address corresponding to the MPL Domain.
- o An MPL Data Message MUST contain an MPL Option in its IPv6 Header to identify the MPL Seed that generated the message and the ordering relative to other MPL Data Messages generated by the MPL Seed.

When the source address is in the AddressList of an MPL Interface corresponding to the MPL Domain Address and the destination address is the MPL Domain Address, the application message and the MPL Data Message MAY be identical. In other words, the MPL Data Message may contain a single IPv6 header that includes the MPL Option.

Otherwise, IPv6-in-IPv6 encapsulation MUST be used to satisfy the MPL Data Message requirements listed above [RFC2473]. The complete IPv6-in-IPv6 message forms an MPL Data Message. The outer IPv6 header conforms to the MPL Data Message requirements listed above. The encapsulated IPv6 datagram encodes the multicast data message that is communicated beyond the MPL Domain.

10.2. MPL Data Message Transmission

An MPL Forwarder manages transmission of MPL Data Messages in its Buffered Message Sets using the Trickle algorithm [RFC6206]. An MPL Forwarder MUST use a separate Trickle timer for each MPL Data Message that it is actively forwarding. In accordance with Section 5 of RFC 6206 [RFC6206], this document defines the following:

- o This document defines a "consistent" transmission as receiving an MPL Data Message that has the same MPL Domain Address, seed-id, and sequence value as the MPL Data Message managed by the Trickle timer.
- o This document defines an "inconsistent" transmission as receiving an MPL Data Message that has the same MPL Domain Address, seed-id

value, and the M flag set, but has a sequence value less than MPL Data Message managed by the Trickle timer.

- o This document does not define any external "events".
- o This document defines MPL Data Messages as Trickle messages.
- o The actions outside the Trickle algorithm that the protocol takes involve managing the MPL Domain's Seed Set and Buffered Message Set.

As specified in [RFC6206], a Trickle timer has three variables: the current interval size I, a time within the current interval t, and a counter c. MPL defines a fourth variable, e, which counts the number of Trickle timer expiration events since the Trickle timer was last reset.

After DATA_MESSAGE_TIMER_EXPIRATIONS Trickle timer events, the MPL Forwarder MUST disable the Trickle timer. When a buffered MPL Data Message does not have an associated Trickle timer, the MPL Forwarder MAY delete the message from the Buffered Message Set by advancing MinSequence of the corresponding MPL Seed in the Seed Set. When the MPL Forwarder no longer buffers any messages for an MPL Seed, the MPL Forwarder MUST NOT increment MinSequence for that MPL Seed.

When transmitting an MPL Data Message, the MPL Forwarder MUST either set the M flag to zero or set it to a level that indicates whether or not the message's sequence number is the largest value that has been received from the MPL Seed.

10.3. MPL Data Message Processing

Upon receiving an MPL Data Message, the MPL Forwarder first processes the MPL Option and updates the Trickle timer associated with the MPL Data Message if one exists.

Upon receiving an MPL Data Message, an MPL Forwarder MUST perform one of the following actions:

- o Accept the message and enter the MPL Data Message in the MPL Domain's Buffered Message Set.
- o Accept the message and update the corresponding MinSequence in the MPL Domain's Seed Set to 1 greater than the message's sequence number.
- o Discard the message without any change to the MPL Information Base.

If a Seed Set entry exists for the MPL Seed, the MPL Forwarder MUST discard the MPL Data Message if its sequence number is less than MinSequence or exists in the Buffered Message Set.

If a Seed Set entry does not exist for the MPL Seed, the MPL Forwarder MUST create a new entry for the MPL Seed before accepting the MPL Data Message.

If memory is limited, an MPL Forwarder SHOULD reclaim memory resources by:

- o Incrementing MinSequence entries in a Seed Set and deleting MPL Data Messages in the corresponding Buffered Message Set that fall below the MinSequence value.
- o Deleting other Seed Set entries that have expired and the corresponding MPL Data Messages in the Buffered Message Set.

If the MPL Forwarder accepts the MPL Data Message, the MPL Forwarder MUST perform the following actions:

- o Reset the Lifetime of the corresponding Seed Set entry to SEED_SET_ENTRY_LIFETIME.
- o If PROACTIVE_FORWARDING is true, the MPL Forwarder MUST initialize and start a Trickle timer for the MPL Data Message.
- o If the MPL Control Message Trickle timer is not running and CONTROL_MESSAGE_TIMER_EXPIRATIONS is non-zero, the MPL Forwarder MUST initialize and start the MPL Control Message Trickle timer.
- o If the MPL Control Message Trickle timer is running, the MPL Forwarder MUST reset the MPL Control Message Trickle timer.

11. MPL Control Messages

11.1. MPL Control Message Generation

An MPL Forwarder generates MPL Control Messages to communicate an MPL Domain's Seed Set and Buffered Message Set to neighboring MPL Forwarders. Each MPL Control Message is generated according to Section 6.2, with an MPL Seed Info for each entry in the MPL Domain's Seed Set. Each MPL Seed Info entry has the following content:

- o S set to the size of the seed-id field in the MPL Seed Info entry.
- o min-seqno set to MinSequence of the MPL Seed.
- o bm-len set to the size of buffered-mpl-messages in octets.
- o seed-id set to the MPL seed identifier.
- o buffered-mpl-messages with each bit representing whether or not an MPL Data Message with the corresponding sequence number exists in the Buffered Message Set. The i'th bit represents a sequence number of min-seqno + i. '0' indicates that the corresponding MPL Data Message does not exist in the Buffered Message Set. '1' indicates that the corresponding MPL Data Message does exist in the Buffered Message Set.

11.2. MPL Control Message Transmission

An MPL Forwarder transmits MPL Control Messages using the Trickle algorithm. An MPL Forwarder maintains a single Trickle timer for each MPL Domain. When CONTROL_MESSAGE_TIMER_EXPIRATIONS is 0, the MPL Forwarder does not execute the Trickle algorithm and does not transmit MPL Control Messages. In accordance with Section 5 of RFC 6206 [RFC6206], this document defines the following:

- o This document defines a "consistent" transmission as receiving an MPL Control Message that indicates neither the receiving nor transmitting node has any new MPL Data Messages to offer.
- o This document defines an "inconsistent" transmission as receiving an MPL Control Message that indicates either the receiving or transmitting node has at least one new MPL Data Message to offer.
- o This document defines an "event" as increasing MinSequence of any entry in the corresponding Seed Set or adding a message to the corresponding Buffered Message Set.

- o This document defines an MPL Control Message as a Trickle message.

As specified in [RFC6206], a Trickle timer has three variables: the current interval size I , a time within the current interval t , and a counter c . MPL defines a fourth variable, e , which counts the number of Trickle timer expiration events since the Trickle timer was last reset. After `CONTROL_MESSAGE_TIMER_EXPIRATIONS` Trickle timer events, the MPL Forwarder MUST disable the Trickle timer.

11.3. MPL Control Message Processing

An MPL Forwarder processes each MPL Control Message that it receives to determine if it has any new MPL Data Messages to receive or offer.

An MPL Forwarder determines if a new MPL Data Message has not been received from a neighboring node if any of the following conditions hold true:

- o The MPL Control Message includes an MPL Seed that does not exist in the MPL Domain's Seed Set.
- o The MPL Control Message indicates that the neighbor has an MPL Data Message in its Buffered Message Set with sequence number greater than `MinSequence` (i.e. the i -th bit is set to 1 and `min-seqno + i > MinSequence`) and is not included in the MPL Domain's Buffered Message Set.

When an MPL Forwarder determines that it has not yet received an MPL Data Message buffered by a neighboring device, the MPL Forwarder MUST reset its Trickle timer associated with MPL Control Message transmissions. If an MPL Control Message Trickle timer is not running, the MPL Forwarder MUST initialize and start a new Trickle timer.

An MPL Forwarder determines if an MPL Data Message in the Buffered Message Set has not yet been received by a neighboring MPL Forwarder if any of the following conditions hold true:

- o The MPL Control Message does not include an MPL Seed for the MPL Data Message.
- o The MPL Data Message's sequence number is greater than or equal to `min-seqno` and not included in the neighbor's corresponding Buffered Message Set (i.e. the MPL Data Message's sequence number does not have a corresponding bit in `buffered-mpl-messages` set to 1).

When an MPL Forwarder determines that it has at least one MPL Data

Message in its corresponding Buffered Message Set that has not yet been received by a neighbor, the MPL Forwarder MUST reset the MPL Control Message Trickle timer. Additionally, for each of those entries in the Buffered Message Set, the MPL Forwarder MUST reset the Trickle timer and reset *e* to 0. If a Trickle timer is not associated with the MPL Data Message, the MPL Forwarder MUST initialize and start a new Trickle timer.

12. Acknowledgements

The authors would like to acknowledge the helpful comments of Robert Cragie, Esko Dijk, Ralph Droms, Paul Duffy, Ulrich Herberg, Owen Kirby, Joseph Reddy, Don Sturek, Dario Tedeschi, and Peter van der Stok, which greatly improved the document.

13. IANA Considerations

This document defines one IPv6 Option, a type that must be allocated from the IPv6 "Destination Options and Hop-by-Hop Options" registry of [RFC2780].

This document defines one ICMPv6 Message, a type that must be allocated from the "ICMPv6 "type" Numbers" registry of [RFC4443].

This document registers two well-known multicast addresses from the IPv6 multicast address space.

13.1. MPL Option Type

IANA is requested to allocate an IPv6 Option Type from the IPv6 "Destination Options and Hop-by-Hop Options" registry of [RFC2780], as specified in Table 1 below:

Mnemonic	act	chg	rest	Description	Reference
MPL_OPT_TYPE	01	1	TBD (suggested value 01101)	MPL Option	This Document

Table 1: IPv6 Option Type Allocation

13.2. MPL ICMPv6 Type

IANA is requested to allocate an ICMPv6 Type from the "ICMPv6 "type" Numbers" registry of [RFC4443], as specified in Table 2 below:

Mnemonic	Type	Name	Reference
MPL_ICMP_TYPE	TBD	MPL Control Message	This Document

Table 2: IPv6 Option Type Allocation

13.3. Well-known Multicast Addresses

IANA is requested to allocate an IPv6 multicast address, with Group ID in the range [0x01,0xFF] for 6LoWPAN compression [RFC6282], "ALL_MPL_FORWARDERS" from the "Variable Scope Multicast Addresses" sub-registry of the "INTERNET PROTOCOL VERSION 6 MULTICAST ADDRESSES" registry.

14. Security Considerations

MPL uses sequence numbers to maintain a total ordering of MPL Data Messages from an MPL Seed. The use of sequence numbers allows a denial-of-service attack where an attacker can spoof a message with a sufficiently large sequence number to: (i) flush messages from the Buffered Message List and (ii) increase the MinSequence value for an MPL Seed in the corresponding Seed Set. The former side effect allows an attacker to halt the forwarding process of any MPL Data Messages being disseminated. The latter side effect allows an attacker to prevent MPL Forwarders from accepting new MPL Data Messages that an MPL Seed generates while the sequence number is less than MinSequence.

More generally, the basic ability to inject messages into a Low-power and Lossy Network can be used as a denial-of-service attack regardless of what forwarding protocol is used. For these reasons, Low-power and Lossy Networks typically employ link-layer security mechanisms to disable an attacker's ability to inject messages.

To prevent attackers from injecting packets through an MPL Forwarder, the MPL Forwarder MUST NOT accept or forward MPL Data Messages from a communication interface that does not subscribe to the MPL Domain Address identified in message's destination address.

MPL uses the Trickle algorithm to manage message transmissions and the security considerations described in [RFC6206] apply.

15. Normative References

- [RFC1982] Elz, R. and R. Bush, "Serial Number Arithmetic", RFC 1982, August 1996.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC2460] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, December 1998.
- [RFC2473] Conta, A. and S. Deering, "Generic Packet Tunneling in IPv6 Specification", RFC 2473, December 1998.
- [RFC2780] Bradner, S. and V. Paxson, "IANA Allocation Guidelines For Values In the Internet Protocol and Related Headers", BCP 37, RFC 2780, March 2000.
- [RFC4007] Deering, S., Haberman, B., Jinmei, T., Nordmark, E., and B. Zill, "IPv6 Scoped Address Architecture", RFC 4007, March 2005.
- [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.
- [RFC6206] Levis, P., Clausen, T., Hui, J., Gnawali, O., and J. Ko, "The Trickle Algorithm", RFC 6206, March 2011.
- [RFC6282] Hui, J. and P. Thubert, "Compression Format for IPv6 Datagrams over IEEE 802.15.4-Based Networks", RFC 6282, September 2011.
- [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.

Authors' Addresses

Jonathan W. Hui
Cisco
170 West Tasman Drive
San Jose, California 95134
USA

Phone: +408 424 1547
Email: jonhui@cisco.com

Richard Kelsey
Silicon Labs
25 Thomson Place
Boston, Massachusetts 02210
USA

Phone: +617 951 1225
Email: richard.kelsey@silabs.com

