

ROLL	J. Guo
Internet-Draft	P. Orlik
Intended status: Standards Track	G. Bhatti
Expires: October 5, 2013	Mitsubishi Electric Research Laboratories
	April 3, 2013

Loop Free DODAG Local Repair  
draft-guo-roll-loop-free-dodag-repair-01

Abstract

IETF has been developing IPv6 based standards for Low-power and Lossy Networks (LLNs) to meet requirements of constrained applications, such as field monitoring, inventory control and so on. IPv6 Routing Protocol for LLNs (RPL) has been published in [RFC6550]. Based on routing metrics and constraints [RFC6551], RPL builds Directed Acyclic Graph (DAG) topology to establish bidirectional routes for LLNs for traffic types of multipoint-to-point, point-to-multipoint, and point-to-point. RPL routes are optimized for traffic to or from one or more roots that act as sinks. As a result, a DAG is partitioned into one or more Destination Oriented DAGs (DODAGs), one DODAG per sink. RPL is widely considered as a feasible routing protocol for LLNs. However, DODAG loops caused by local DODAG repair mechanism is an issues to be addressed. This draft introduces a loop free local DODAG repair mechanism. This draft also introduces a piggybacked data option for transferring delay sensitive data during route repair process. The piggybacked data can be included in DODAG Repair Request (DRQ) message or DODAG Information Solicitation (DIS) message.

## 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 April 4, 2013.

## Copyright Notice

Copyright (c) 2012 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 . . . . .	4
2. Terminology . . . . .	5
3. ICMPv6 RPL Control Message Extension . . . . .	6
3.1. DODAG Repair Request (DRQ) . . . . .	6
3.1.1. Format of the DRQ Base Object . . . . .	6
3.1.2. Secure DRQ . . . . .	7
3.1.3. DRQ Options . . . . .	8
3.2. DODAG Repair Reply (DRP). . . . .	8
3.2.1. Format of the DRP Base Object . . . . .	8
3.2.2. Secure DRP . . . . .	9
3.2.3. DRP Options . . . . .	9
3.3. Format of the Path Option . . . . .	9
4. DODAG Local Repair . . . . .	10
4.1. DODAG Local Repair in Storing Mode . . . . .	11
4.1.1. DRQ Message Processing . . . . .	11
4.1.2. DRP Message Processing . . . . .	12
4.2. DODAG Local Repair in Non-Storing Mode . . . . .	13
4.2.1. DRQ Message Processing . . . . .	14
4.2.2. DRP Message Processing . . . . .	14
5. DIS Message with Piggybacked Data . . . . .	15
5.1. Format of the Modified DIS Base Object . . . . .	17
5.2. Modified DIS Options . . . . .	17
5.3. Process of the Modified DIS message . . . . .	17
6. Security Considerations . . . . .	17
7. IANA Considerations . . . . .	18
8. References . . . . .	18
8.1. Normative References . . . . .	18
8.2. Informative References . . . . .	19
Authors' Addresses . . . . .	19

## 1. Introduction

Low-power and Lossy Networks (LLNs) are a class of networks in which nodes and their communication links are constrained. LLN nodes typically operate with constraints on processing power, physical size, memory, power consumption, lifetime, and rate of activity. Their communication links are characterized by high loss rate, low data rate, instability, low transmission power, and short transmission range. There can be from a few dozen up to thousands of nodes within a LLN. Routing in LLNs is different from routing in mobile ad-hoc networks. IETF has developed an IPv6 Routing Protocol for LLNs (RPL) in [RFC6550]. RPL supports multipoint-to-point traffic and point-to-multipoint traffic. The support for point-to-point traffic is also available.

RPL builds Directed Acyclic Graph (DAG) topology, which is partitioned into one or more Destination Oriented DAGs (DODAGs). DODAG is basic logical structure in RPL. RPL nodes construct and maintain DODAG through the DODAG Information Object (DIO) message which is transmitted via link-local multicasting by using the Trickle timer [RFC6206]. The sink in a DODAG is called the DODAG root. RPL defines rules to transmit the DIO messages within a DODAG. The DODAG root configures the DODAG parameters including RPLInstanceID, DODAGVersionNumber, DODAGID, Rank, etc. and advertises the DODAG parameters in the DIO messages. To join a DODAG, a node selects a set of DODAG parents, on the routes towards the DODAG root, and a preferred DODAG parent as the preferred next hop node for upward traffic. Once a node joins a DODAG, it transmits DIO messages to advertise the DODAG parameters.

The traffic inside a LLN flows along the edges of the DODAG, either upward or downward. In RPL, upward routes, having the DODAG root as destination, are provided by the DODAG construction mechanism using DIO messages. Downward routes, from the DODAG root to any other destination, are provided by these destinations transmitting the Destination Advertisement Object (DAO) messages.

Three different modes of operation (MOP) for downward routes are specified in [RFC6550]:

- 1) No downward routes maintained by RPL.
- 2) Storing mode of operation in which each router stores downward routing tables for its sub-DODAG. In Storing mode, the DAO message is sent to DAO parents. A node unicasts the DAO messages to the selected parent(s). Transmission of the DAO messages propagates from the nodes towards the DODAG root, where each intermediate router adds its downward routing stack to the DAO messages. In Storing mode, downward traffic is sent by using the downward

routing tables.

- 3) Non-Storing mode of operation in which only the DODAG root stores routes to all nodes in the network. In Non-Storing mode, the DAO message is sent to the DODAG root. A node unicasts the DAO messages to the DODAG root, which then calculates routes to all destinations by piecing together the information collected from the DAO messages. In Non-Storing mode, downward traffic is sent by way of source routing.

An RPL node may act as a leaf node or as a router. RPL defines operation rules for both leaf node and router in [RFC6550]. For example, a leaf node does not extend DODAG connectivity. An RPL router needs to implement Trickle [RFC6206]. An RPL router implementation needs to support the MOP in use by the DODAG, that is, support for upward routes only or support for upward routes and downward routes in Storing mode or support for upward routes and downward routes in Non-Storing mode.

RPL has been implemented and tested. It has been shown that DODAG loops occur quite often [83rd IETF Meeting Presentation]. The cause of DODAG loops comes from rank increase by DODAG local repair mechanism. This draft introduces a method for repairing DODAG locally without causing any DODAG loops. The DODAG local repair method applies to both Storing and Non-Storing modes of operation in RPL.

## 2. Terminology

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 [RFC2119].

Additionally, this draft employs terminologies defined in [RFC6550], and extends following terminologies:

DIO: DODAG Information Object in which the rank is represented by two integers.

Up: Up refers to the direction from leaf node or router node towards the DODAG root.

Down: Down refers to the direction from the DODAG root towards leaf node or router node.

This draft introduces the following new terminologies:

DRQ: DODAG Repair Request

DRP: DODAG Repair Reply

Rank\_DRQ: The rank of the node generating the DRQ message.

Rank\_DRP: The rank of the node transmitting the DRP message.

DRQID: IPv6 address of the node generating DRQ message.

DRSN: Sequence number of the DRQ message of the node generating DRQ message.

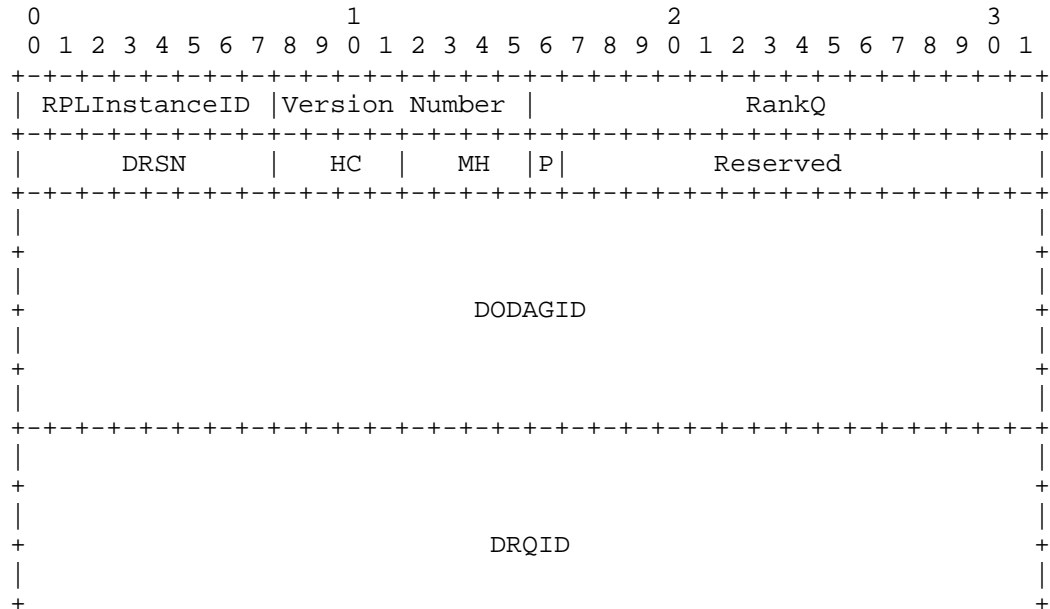
### 3. ICMPv6 RPL Control Message Extension

This draft uses RPL control messages defined in in Figure 6 and Figure 7 of [RFC6550]. In addition, to repair DODAG locally, two new RPL control messages, DODAG Repair Request (DRQ) message and DODAG Repair Reply (DRP) message, are introduced. The code field for the DRQ and DRP messages needs to be assigned by IANA. The message base for DRQ and DRP are defined as follows.

#### 3.1. DODAG Repair Request (DRQ)

The DRQ message is used by a node to repair a DODAG locally if a parent becomes unreachable. A node may also use the DRQ message to discover additional parents if it is necessary.

##### 3.1.1. Format of the DRQ Base Object



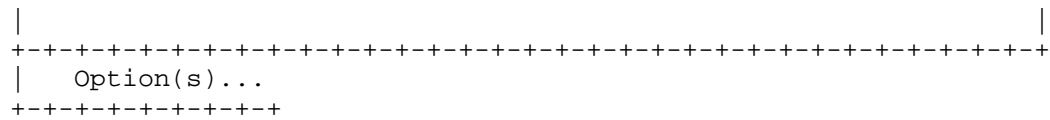


Figure 1: The DRQ Base Object

RPLInstanceID: 8-bit unsigned field as described in [RFC6550] to indicate which RPL Instance the DODAG is a part.

Version Number: 8-bit unsigned integer as described in [RFC6550] to indicate the DODAGVersionNumber.

RankQ: 16-bit unsigned integer indicating rank of the node generating the DRQ message.

DRSN: 8-bit field indicating sequence number of the DRQ message at the node generating the DRQ message.

HC: 4-bit field to indicate the number of hops traveled by a DRQ message.

MH: 4-bit field to indicate the maximum number of hops a DRQ message can travel. If a DRQ message reaches the maximum number of hops, it must be ignored.

P: 1-bit flag to indicate if the sender has included the piggybacked data. If P = 1, piggybacked data is present and a receiver MAY consider to forward piggybacked data towards the DODAG root. If P = 0, piggybacked is not present. A node sets P flag to 1 only if its parent set is empty and it has a delay sensitive packet to transmit. For example, if a building monitoring sensor detects fire, it must send a notification as soon as possible.

Reserved: 15 bits of the DRQ Base are reserved. They must be set to zero on transmission and must be ignored on reception.

DODAGID: 128-bit field as defined in [RFC6550]. A DODAGID is the identifier of a DODAG root. The DODAGID is unique within the scope of a RPL Instance in the LLN. The DODAGID must be a routable IPv6 address belonging to the DODAG root.

DRQID: 128-bit IPv6 address of the node generating the DRQ message.

### 3.1.2. Secure DRQ

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

### 3.1.3. DRQ Options

The DRQ message may carry valid options.

This draft allows for the DRQ message to carry the following options:

0x00 Pad1

0x01 PadN

Path: This option field is present only if MOP is Non-Storing. The Path field contains IPv6 addresses of traversed nodes by the DRQ message along the path.

PiggyData: This option field is present only if flag P = 1.

PiggyData field contains data to be forward to the root..

### 3.2. DODAG Repair Reply (DRP)

Upon receiving a DRQ message, a router with lower rank and non-empty DODAG parent set may generate a DRP message in responding to a received DRQ message.

#### 3.2.1. Format of the DRP Base Object

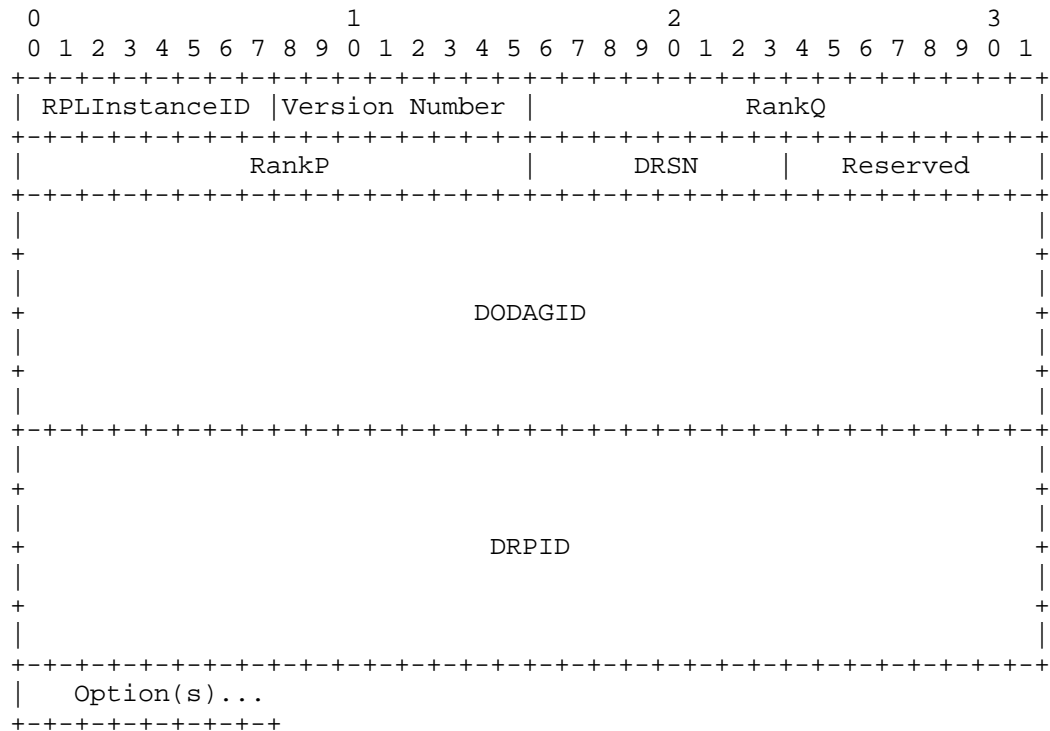


Figure 2: The DRP Base Object

RPLInstanceID: 8-bit unsigned field as described in [RFC6550] to indicate which RPL Instance the DODAG is a part.

Version Number: 8-bit unsigned integer as described in [RFC6550] to indicate the DODAGVersionNumber.

RankQ: 16-bit unsigned integer indicating rank of the node generating the DRQ message.

RankP: 16-bit unsigned integer indicating rank of the node transmitting the DRP message.

DRSN: 8-bit field indicating the sequence number of DRQ message at the node generating the DRQ message.

Reserved: 8 bits of the DRP Base are reserved. They must be set to zero on transmission and must be ignored on reception.

DODAGID: 128-bit field as defined in [RFC6550]. A DODAGID is the identifier of a DODAG root. The DODAGID is unique within the scope of a RPL Instance in the LLN. The DODAGID MUST be a routable IPv6 address belonging to the DODAG root.

DRPID: 128-bit IPv6 address of the node that is destination of the DRP message.

### 3.2.2. Secure DRP

A Secure DRP message follows the format in Figure 7 of [RFC6550], where the base format is the DRP message shown in Figure 2.

### 3.2.3. DRP Options

The DRP message may carry valid options.

This draft allows for the DRP message to carry the following options:

0x00 Pad1

0x01 PadN

Path: This option is present only if MOP is Non-Storing. The Path field contains IPv6 addresses of traversed nodes by the DRQ message along the path.

### 3.3. RPL Control Message Options

The formats of option Pad1 and PadN are described in Figure 20 and

Figure 21 of [RFC6550], respectively.

The format of the Path option is as follows:

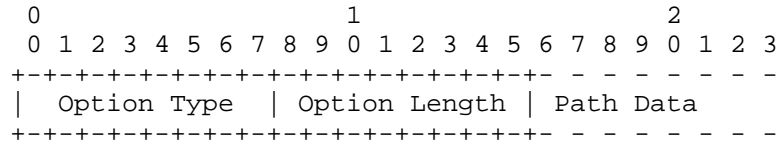


Figure 3: Format of the Path Option

Option Type: 8-bit identifier of the type of option. The Option Type value needs to be assigned by IANA.

Option Length: 8-bit unsigned integer, representing the length in octets of the option Path Data field, not including the Option Type and Option Length fields.

Path Data: A variable length field that contains a list of IPv6 addresses.

The format of the PiggyData option is as follows:

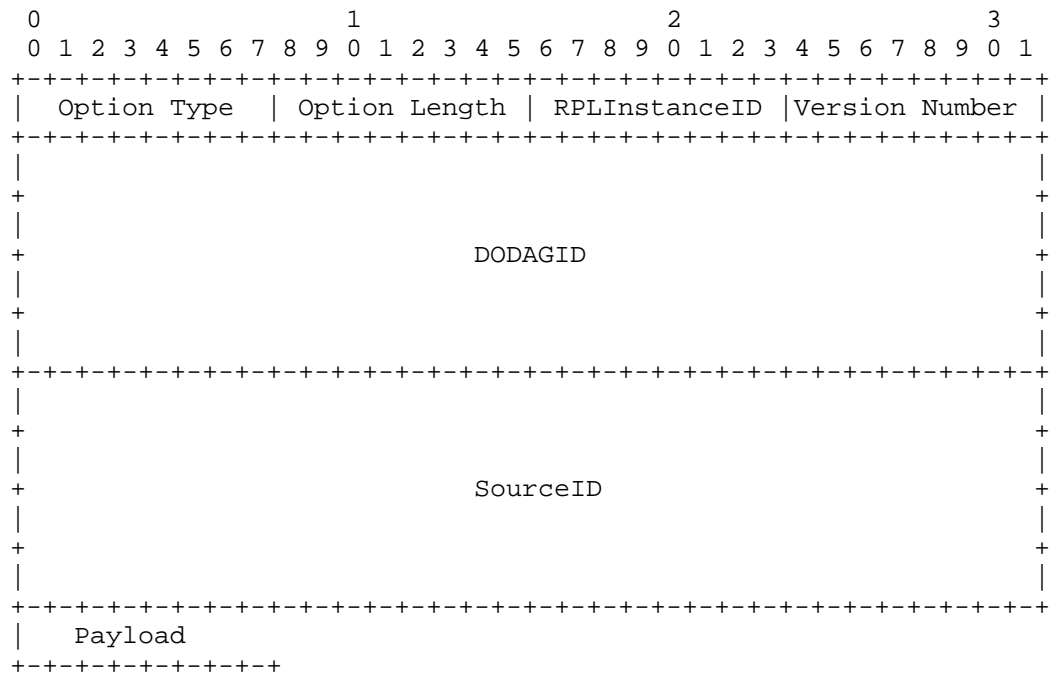


Figure 4: Format of the PiggyData Option

Option Type: 8-bit identifier of the type of option. The Option Type value needs to be assigned by IANA.

Option Length: 8-bit unsigned integer, representing the length in octets of the option PiggyData field, not including the Option Type and Option Length fields.

RPLInstanceID, Version Number and DODAGID are described in RFC 6550. SourceID is same as DRQID.

PiggyData: A variable length field that contains payload of piggybacked data.

#### 4. Loop Free DODAG Local Repair

In RPL, the rank plays very important role in the DODAG construction and maintenance. The rank of a node defines a position of the node relative to other nodes with respect to the DODAG root. Each node maintains its own rank. The DODAG root has the lowest rank. Nodes maintain their ranks based on parent-child relationship such that a child must have a rank strictly greater than ranks of all its DODAG parents. The DODAG root has no parent. The acyclic structure of the DODAG is guaranteed as long as the rank of any node is strictly greater than ranks of its DODAG parents. It is safe for a node to decrease its rank, as long as its rank remains greater than the ranks of its DODAG parents. However, rank increase can cause DODAG loops.

The DODAG local repair methods provided in this draft does not increase the rank, and therefore, is loop free.

When a DODAG parent becomes unreachable, a node may switch to another DODAG parent for upward traffic. DODAG may be locally repaired by the node transmitting a DRQ message. The DRQ message is transmitted by the DRQ message generator via link-local multicasting to all-RPL-nodes.

Upon receiving a DRQ message, a link-local neighboring router which is not the DODAG root discards the DRQ message if it does not have any DODAG parent. If the link-local neighboring router is the DODAG root, it accepts the DRQ message and generates a DRP message. If the link-local neighboring router is not the DODAG root and has a non-empty DODAG parent set and its rank is lower than RankQ, it accepts the DRQ message and generates a DRP message. If the link-local neighboring router is not the DODAG root and has a non-empty DODAG parent set and its rank is greater than or equal to RankQ, it forwards the DRQ message to its preferred DODAG parent. This

forwarding process continues until the DRQ message is discarded by a router that has an empty DODAG parent set or the DRQ message reaches a node which is either the DODAG root or a router that has a non-empty DODAG parent set and a rank lower than RankQ, a DRP message is then generated.

The DRP message is unicasted. In Storing mode, the DRP message generator transmits the DRP message to the DRQ message generator by using a downward routing table. In Non-Storing mode, the DRP message generator transmits the DRP message to the DRQ message generator by way of source routing via the Path option.

#### 4.1. DODAG Local Repair in Storing Mode

In Storing mode, if a DODAG parent becomes unreachable, a node removes that DODAG parent from its DODAG parent set.

If the updated DODAG parent set becomes empty, the node shall transmit a DRQ message to discover new DODAG parents.

If the updated DODAG parent set is not empty, the node checks if the removed DODAG parent is its preferred DODAG parent. If yes, the node shall select a new preferred DODAG parent. Whether or not the removed DODAG parent is the preferred DODAG parent, the node may transmit a DRQ message to discover additional parents. The node may also schedule a No-Path DAO message transmission if the removed DODAG parent is its DAO parent.

To transmit a DRQ message in Storing mode, the node generates a DRQ message. It sets RPLInstanceID, DODAGVersionNumber and DODAGID by using the maintained DODAG parameters. It sets RankQ to its rank. The node increases its DRSN by 1 and sets HC = 0, and MH to an appropriate value. There is no Path option field in Storing mode.

##### 4.1.1. DRQ Message Processing

When a router receives a DRQ message, it discards the DRQ message if its DODAG parent set is empty. Otherwise, the router performs the following filtering process and discards the DRQ message if any of following conditions is true:

- i) RPLInstanceID or DODAGVersionNumber or DODAGID in the DRQ message is not equal to the respective value maintained by the router.
- ii) The DRQ message was received already by comparing DRQID and DRSN.
- iii) HC is equal to MH.
- iv) The DRQ message is transmitted by router's DODAG parent.

v) The DRQ message is generated by router itself or by its DODAG parent.

If the DRQ message passes filtering process, the receiving router processes the DRQ message further.

If the receiving router is the DODAG root, it accepts the DRQ message and generates a DRP message. To generate a DRP message, the DODAG root copies RPLInstanceID, DODAGVersionNumber, DRSN, and DODAGID from the DRQ message. It sets DRPID to DRQID, RankQ to the RankQ in DRQ message, and RankP to its rank. The DODAG root forwards the DRP message to the node from which it received the DRQ message.

If the receiving router is not the DODAG root and its rank is lower than RankQ, the router accepts the DRQ message and generates a DRP message as the root does. The router forwards the DRP message to the node from which it received the DRQ message.

If the receiving router is not the DODAG root and its rank is greater than or equal to RankQ, it adds a route entry to node DRQID into its downward routing table, increases value of HC field by 1 and forwards the DRQ message to its preferred DODAG parent.

If P = 1 and the receiver is the DODAG root, the root processes piggybacked data. The root MAY also mark downward routes to sender as invalid and waits for new DAO messages to reestablish downward routes.

If P = 1 and the receiver is not the DODAG root, besides generating DRP message, the receiver MAY forward the piggybacked data to its parent.

#### 4.1.2. DRP Message Processing

When a node receives a DRP message, it first performs filtering process and discards the DRP message if any of following conditions is true:

- i) RPLInstanceID or DODAGVersionNumber or DODAGID in the DRP message is not equal to the respective value maintained by the receiving node.
- ii) The DRP message was received already by comparing DRPID and DRSN.
- iii) The receiving node is leaf node and is not the DRQ message generator.

If DRP message passes filtering process, the receiving node processes the DRP message further.

The receiving node can be a leaf node or a router.

If the receiving node is the DRQ message generator and the DRP message sender is not in its DODAG parent set, it may add the DRP message sender into its DODAG parent set and select a new preferred DODAG parent. The receiving node may schedule a DAO message transmission if the DRP message sender is added into its DAO parent set.

If the receiving node is not the DRQ message generator, it must be a router. If the receiving router has no route entry to node DRPID in its downward routing table, it discards the DRP message.

If the receiving router has a downward route entry to node DRPID and its rank is greater than or equal to RankQ, it checks if it can decrease its rank such that  $\text{RankQ} > \text{its rank} > \text{RankP}$ . If not, the receiving router discards the DRP message. If yes, the receiving router decreases its rank to an appropriate value and may add the DRP message sender into its DODAG parent set if the sender is not in its DODAG parent set. The receiving router updates its DODAG parent set caused by its rank decrease, that is, removing DODAG parents whose ranks are greater than or equal to its new rank. If its preferred DODAG parent is removed, it selects a new preferred DODAG parent. The receiving router then updates the RankP of the DRP message to its rank, and forwards the DRP message to next hop node on the downward route. It may schedule a No-Path DAO message transmission if any of its DAO parents is removed due to its rank decrease or the DRP sender was added into its DAO parent set.

If the receiving router has a downward route entry to node DRPID and its rank is lower than RankQ, it updates the RankP field of the DRP message to its rank, and forwards the DRP message to next hop node on the downward route.

The rank of receiving router is less than RankQ if the receiving router is on multiple DODAG repair downward routes. When the receiving router receives a DRP message, it may decrease its rank. Therefore, subsequent DRP messages may carry a RankQ greater than or equal to the rank of receiving router. If the receiving router is only on a single DODAG repair downward route, its rank must be greater than or equal to RankQ based on the DRQ message process procedure.

#### 4.2. DODAG Local Repair in Non-Storing Mode

The handling of unreachable parent in Non-Storing mode is similar to that in Storing mode. However, there are two differences. The first difference is that after removing a DAO parent from its DAO parent

set, if its DODAG parent set is not empty, a node may schedule a DAO message transmission instead of the No-Path DAO message transmission. The second difference is that to generate a DRQ message in Non-Storing mode, a node adds a Path option field by inserting its IPv6 address into the Path option field.

#### 4.2.1. DRQ Message Processing

When a router receives a DRQ message, it performs same filtering process as that in Storing mode. If the DRQ message passes filtering process, the receiving router processes the DRQ message further.

If the receiving router is the DODAG root, it accepts the DRQ message and generates a DRP message similarly as in Storing mode. In addition, the DODAG root adds a Path option in DRP message and copies Path option field from DRQ message to Path option field of DRP message. The DODAG root transmits the DRP message to destination node DRPID along the route specified by the Path option, and on the downward route, intermediate routers obtain route information from the Path option field of DRP message.

If the receiving router is not the DODAG root and its rank is lower than Rank\_DRQ, it accepts the DRQ message and generates a DRP message as the DODAG root does. It then transmits the DRP message to destination node DRPID along the route specified by the Path option.

If the receiving router is not the DODAG root and its rank is greater than or equal to Rank\_Q, it updates the DRQ message by inserting its own IPv6 address into the Path option field, increasing the value of HC field by 1 and forwards the DRQ message to its preferred DODAG parent.

If  $P = 1$  and the receiver is the DODAG root, the DODAG root processes piggybacked data. The DODAG root MAY also mark downward routes to node REQID as invalid and waits for new DAO messages to reestablish downward routes.

If  $P = 1$  and the receiver is not the DODAG root, besides generating the DRP message, the receiver MAY forward the piggybacked data to its parent.

#### 4.2.2. DRP Message Processing

When a node receives a DRP message, it performs the same filtering process as in Storing mode. If the DRP message passes filtering process, the receiving node processes the DRP message further.

The receiving node can be a leaf node or a router.

If the receiving node is the DRQ message generator and the DRP message sender is not in its DODAG parent set, it may add the DRP message sender into its DODAG parent set. The receiving node may select a new preferred DODAG parent. It may also schedule a DAO message transmission if the DRP message sender is added into its DAO parent set.

If the receiving node is not DRQ message generator, it must be a router. If the receiving router is not on the route specified by Path option, it discards the DRP message.

If the receiving router is on the route specified by Path option and its rank is greater than or equal to RankQ, the receiving router checks if it can decrease its rank such that  $\text{RankQ} > \text{its rank} > \text{RankP}$ . If no, the receiving node discards the DRP message. If yes, the receiving node decreases its rank to an appropriate value and may add the DRP message sender into its DODAG parent set if the sender is not in its DODAG parent set. The receiving router updates its DODAG parent set by removing any DODAG parent whose rank is greater than or equal to its new rank. If its preferred DODAG parent is removed, it selects a new preferred DODAG parent. The receiving router updates the RankP of DRP message to its new rank, and forwards the DRP message to the destination node DRPID by obtaining next hop node via the Path option field. Furthermore, the receiving router may schedule a DAO message transmission if any of its DAO parents was removed due to its rank decrease or the DRP sender was added into its DAO parent set.

If the receiving router is on the route specified by Path option and its rank is lower than RankQ, the receiving router updates the RankP to its rank, and forwards the DRP message to destination node DRPID by obtaining next hop node via Path option.

The rank of receiving router is less than RankQ if the receiving router is on multiple DODAG repair downward routes. When the receiving router receives a DRP message, it may decrease its rank. Therefore, subsequent DRP messages may carry a RankQ greater than or equal to the rank of receiving router. If the receiving router is only on a single DODAG repair downward route, its rank must be greater than or equal to RankQ based on the DRQ message process procedure.

## 5. DIS Message with Piggybacked Data

For some delay sensitive applications, data must be sent to data sink as soon as possible. For example, if a building monitoring sensor detects fire, it must send the alert message to system controller

without delay since in this case, any delay could be costly.

If a node has a delay sensitive data and an empty parent set, it MAY piggyback data into the DIS message. To accomplish this, a modified DIS message is proposed.

### 5.1 Format of the Modified DIS Base Object

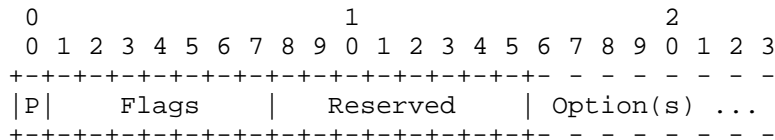


Figure 5: Modified Format of the DIS Base Object

P: 1-bit flag to indicate if sender has included a piggybacked data option. If P = 1, piggybacked data is present, and the receiver MAY consider to forward piggybacked data towards the DODAG root. If P = 0, piggybacked is not present. A sender sets P flag to 1 only if its parent set is empty and it has a delay sensitive data to transmit.

Flags: 7-bit unused field reserved for flags. The field MUST be initialized to zero by the sender and MUST be ignored by the receiver.

8-bit Reserved field and Option(s) field are same as described in RFC 6550.

### 5.2 Modified DIS Options

Besides options specified in RFC 6550, the DIS message MAY carry PiggyData option as shown in Figure 4. However, this option is present only if flag P = 1.

### 5.3 Process of the Modified DIS Message

If flag P = 0, the process of the DIS message is same as specified in RFC 6550. If flag P = 1, the receiver MAY perform extra process. If receiver is the DODAG root, the root processes piggybacked data. The root MAY also mark downward routes to node REQID as invalid and waits for new DAO messages to reestablish downward routes. If the receiver is not the DODAG root and sender is not its parent, it MAY forward the piggybacked data to its parent.

## 6. Security Considerations

This draft introduces an alternative rank computation method and a DODAG local repair mechanism. In general, the security considerations for the DODAG construction and maintenance 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. New RPL control messages (DRQ and DRP) defined in this draft have secure versions as well.

## 7. IANA Considerations

This draft defines two new RPL Control Messages types and a new RPL Control Message Option.

Code field for the DODAG Repair Request (DRQ) message needs to be assigned by IANA.

Code field for the DODAG Repair Reply (DRP) message needs to be assigned by IANA.

Option Type field for Path option field needs to be assigned by IANA.

## 8. References

### 8.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC4443] Conta, A., Deering, S., and M. Gupta, "Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification", RFC 4443, March 2006.
- [RFC6550] Winter, T., Thubert, P., Brandt, A., Hui, J., Kelsey, R., Levis, P., Pister, K., Struik, R., Vasseur, JP., and R. Alexander, "RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks", RFC 6550, March 2012.
- [RFC6206] Levis, P., Clausen, T., Hui, J., Gnawali, O., Ko, J., "The Trickle Algorithm", RFC 6206, March 2011.
- [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.

## 8.2. Informative References

[83rd IETF Meeting Presentation] Clausen, T., Yi, J., Colin de Verdiere, A., Herberg, U., "Experiences with RPL: IPv6 Routing Protocol for Low power and Lossy Networks", Paris, France, March 2012.

## Authors' Addresses

Jianlin Guo  
Mitsubishi Electric Research Laboratories  
201 Broadway  
Cambridge, Massachusetts 02139  
USA

Phone: +1 617 621 7541  
Email: guo@merl.com

Philip Orlik  
Mitsubishi Electric Research Laboratories  
201 Broadway  
Cambridge, Massachusetts 02139  
USA

Phone: +1 617 621 7570  
Email: porlik@merl.com

Ghulam Bhatti  
Mitsubishi Electric Research Laboratories  
201 Broadway  
Cambridge, Massachusetts 02139  
USA

Phone: +1 617 621 7513  
Email: gbhatti@merl.com

ROLL  
Internet-Draft  
Intended status: Standards Track  
Expires: April 22, 2013

J. Hui  
Cisco  
R. Kelsey  
Silicon Labs  
October 19, 2012

Multicast Protocol for Low power and Lossy Networks (MPL)  
draft-ietf-roll-trickle-mcast-02

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 drive packet transmissions for both control and data-plane packets. Specific 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 April 22, 2013.

Copyright Notice

Copyright (c) 2012 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. Overview . . . . .	5
4. Message Formats . . . . .	7
4.1. MPL Option . . . . .	7
4.2. ICMPv6 MPL Message . . . . .	8
4.2.1. MPL Window . . . . .	9
5. MPL Forwarder Behavior . . . . .	11
5.1. Multicast Packet Dissemination . . . . .	11
5.1.1. Trickle Parameters and Variables . . . . .	12
5.1.2. Proactive Propagation . . . . .	12
5.1.3. Reactive Propagation . . . . .	13
5.2. Sliding Windows . . . . .	13
5.3. Transmission of MPL Multicast Packets . . . . .	15
5.4. Reception of MPL Multicast Packets . . . . .	16
5.5. Transmission of ICMPv6 MPL Messages . . . . .	16
5.6. Reception of ICMPv6 MPL Messages . . . . .	17
6. MPL Parameters . . . . .	19
7. Acknowledgements . . . . .	20
8. IANA Considerations . . . . .	21
9. Security Considerations . . . . .	22
10. References . . . . .	23
10.1. Normative References . . . . .	23
10.2. Informative References . . . . .	23
Authors' Addresses . . . . .	24

## 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 topology and forwarding mechanisms. Traditional IP multicast forwarding typically relies on topology maintenance mechanisms to forward multicast messages 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 packets away from the LBR. For the same reasons, a LLN device may not be able to maintain a multicast forwarding topology when operating with limited memory.

Furthermore, the dynamic properties of wireless networks can make the cost of maintaining a multicast forwarding 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 forwarding 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", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

The following terms are used throughout this document:

MPL forwarder	An IPv6 router that subscribes to the MPL multicast group and participates in disseminating MPL multicast packets.
MPL multicast scope	The multicast scope that MPL uses when forwarding MPL multicast packets. In other words, the multicast scope of the IPv6 Destination Address of an MPL multicast packet.
MPL domain	A connected set of MPL forwarders that define the extent of the MPL dissemination process. As a form of flood, all MPL forwarders in an MPL domain will receive MPL multicast packets. The MPL domain MUST be composed of at least one MPL multicast scope and MAY be composed of multiple MPL multicast scopes.
MPL seed	A MPL forwarder that begins the dissemination process for an MPL multicast packet. The MPL seed may be different than the source of the original multicast packet.
MPL seed identifier	An identifier that uniquely identifies an MPL forwarder within its MPL domain.
original multicast packet	An IPv6 multicast packet that is disseminated using MPL.
MPL multicast packet	An IPv6 multicast packet that contains an MPL Hop-by-Hop Option. When either source or destinations are beyond the MPL multicast scope, the MPL multicast packet is an IPv6-in-IPv6 packet that contains an MPL Hop-by-Hop Option in the outer IPv6 header and encapsulates an original multicast packet. When both source and destinations are within the MPL multicast scope, the MPL Hop-by-Hop Option may be included directly within the original multicast packet.

### 3. Overview

MPL delivers IPv6 multicast packets by disseminating them to all MPL forwarders within an MPL domain. MPL dissemination is a form of flood. An MPL forwarder may broadcast/multicast an MPL multicast packet out of the same physical interface on which it was received. Using link-layer broadcast/multicast allows MPL to forward multicast packets without explicitly identifying next-hop destinations. An MPL forwarder may also broadcast/multicast MPL multicast packets out other interfaces to disseminate the message across different links. MPL does not build or maintain a multicast forwarding topology to forward multicast packets.

Any MPL forwarder may initiate the dissemination process by serving as an MPL seed for an original multicast packet. The MPL seed may or may not be the same device as the source of the original multicast packet. When the original multicast packet's source is outside the LLN, the MPL seed may be the ingress router. Even if an original multicast packet source is within the LLN, the source may first forward the multicast packet to the MPL seed using IPv6-in-IPv6 tunneling. Because MPL state requirements grows with the number of active MPL seeds, limiting the number of MPL seeds reduces the amount of state that MPL forwarders must maintain.

Because MPL typically broadcasts/multicasts MPL packets out of the same interface on which they were received, MPL forwarders are likely to receive an MPL multicast packet more than once. The MPL seed tags each original multicast packet with an MPL seed identifier and a sequence number. The sequence number provides a total ordering of MPL multicast packets disseminated by the MPL seed.

MPL defines a new IPv6 Hop-by-Hop Option, the MPL Option, to include MPL-specific information along with the original multicast packet. Each IPv6 multicast packet that MPL disseminates includes the MPL Option. Because the original multicast packet's source and the MPL seed may not be the same device, the MPL Option may be added to the original multicast packet en-route. To allow Path MTU discovery to work properly, MPL encapsulates the original multicast packet in another IPv6 header that includes the MPL Option.

Upon receiving a new MPL multicast packet for forwarding, the MPL forwarder may proactively transmit the MPL multicast packet a limited number of times and then falls back into an optional reactive mode. In maintenance mode, an MPL forwarder buffers recently received MPL multicast packets and advertises a summary of recently received MPL multicast packets from time to time, allowing neighboring MPL forwarders to determine if they have any new multicast packets to offer or receive.

MPL forwarders schedule their packet (control and data) transmissions using the Trickle algorithm [RFC6206]. Trickle's adaptive transmission interval allows MPL to quickly disseminate messages when there are new MPL multicast packets, but reduces transmission overhead as the dissemination process completes. Trickle's suppression mechanism and transmission time selection allow MPL's communication rate to scale logarithmically with density.

## 4. Message Formats

#### 4.1. MPL Option

The MPL Option is carried in an IPv6 Hop-by-Hop Options header, immediately following the IPv6 header. The MPL Option 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	2	3	4	5	6	7	8	9	0	1
																				+-----+-----+-----+-----+										+-----+-----+-----+-----+											
																				Option Type										Opt Data Len											
+-----+-----+-----+-----+										+-----+-----+-----+-----+										+-----+-----+-----+-----+										+-----+-----+-----+-----+											
S   M										rsv										sequence										seed-id (optional)											
+-----+-----+-----+-----+										+-----+-----+-----+-----+										+-----+-----+-----+-----+										+-----+-----+-----+-----+											

Option Type	XX (to be confirmed by IANA).
Opt Data Len	Length of the Option Data field in octets. MUST be set to either 2 or 4.
S	2-bit unsigned integer. Identifies the length of seed-id. 0 indicates that the seed-id is 0 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. 0 indicates that the value in sequence is not the greatest sequence number that was received from the MPL seed.
rsv	5-bit reserved field. MUST be set to zero and incoming MPL multicast packets in which they are not zero MUST be dropped.
sequence	8-bit unsigned integer. Identifies relative ordering of MPL multicast packets from the source identified by seed-id.
seed-id	Uniquely identifies the MPL seed that initiated dissemination of the MPL multicast packet. The size of seed-id is indicated by the S field.

The Option Data of the Trickle Multicast option MUST NOT change as the MPL multicast packet is forwarded. Nodes that do not understand

the Trickle Multicast option MUST discard the packet. Thus, according to [RFC2460] the three high order bits of the Option Type must be set to '010'. The Option Data length is variable.

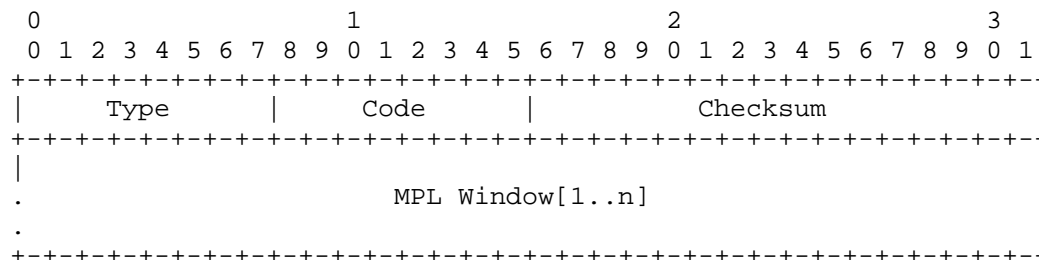
The seed-id uniquely identifies an MPL seed within the MPL domain. When seed-id is 128 bits (S=3), the MPL seed MAY use an IPv6 address assigned to one of its interfaces that is unique within the MPL domain. Managing MPL seed identifiers is not within scope of this document.

The sequence field establishes a total ordering of MPL multicast packets from the same MPL seed. The MPL seed MUST increment the sequence field's value on each new MPL multicast packet that it disseminates. 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.

#### 4.2. ICMPv6 MPL Message

The MPL forwarder uses ICMPv6 MPL messages to advertise information about recently received MPL multicast packets. The ICMPv6 MPL message has the following format:



#### IP Fields:

**Source Address** A link-local address assigned to the sending interface.

**Destination Address** The link-local all-nodes MPL forwarders multicast address (FF02::TBD).

Hop Limit                    255

ICMPv6 Fields:

Type                        XX (to be confirmed by IANA).

Code                        0

Checksum                    The ICMP checksum. See [RFC4443].

MPL Window[1..n]          List of one or more MPL Windows (defined in Section 4.2.1).

An MPL forwarder transmits an ICMPv6 MPL message to advertise information about buffered MPL multicast packets. More explicitly, the ICMPv6 MPL message encodes the sliding window state (described in Section 5.2) that the MPL forwarder maintains for each MPL seed. The advertisement serves to indicate to neighboring MPL forwarders regarding newer messages that it may send or the neighboring MPL forwarders have yet to receive.

#### 4.2.1. MPL Window

An MPL Window encodes the sliding window state (described in Section 5.2) that the MPL forwarder maintains for an MPL seed. Each MPL Window 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
+-----+-----+-----+-----+-----+-----+-----+-----+
|           w-min           |   w-len   | S | seed-id (0, 2 or 16 octets) |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     buffered-mpl-packets (0 to 8 octets)
|                                     |
.                                     .
.                                     .
+-----+-----+-----+-----+-----+-----+-----+-----+

```

w-min                        8-bit unsigned integer. Indicates the first sequence number associated with the first bit in buffered-mpl-packets.

w-len                        6-bit unsigned integer. Indicates the size of the sliding window and the number of valid bits in buffered-mpl-packets. The sliding window's upper bound is the sum of w-min and w-len.

S                    2-bit unsigned integer. Identifies the length of seed-id. 0 indicates that the seed-id value is 0 and not included in the MPL Option. 1 indicates that the seed-id value is a 16-bit unsigned integer. 2 indicates that the seed-id value is a 128-bit unsigned integer. 3 is reserved.

seed-id             Indicates the MPL seed associated with this sliding window.

buffered-mpl-packets   Variable-length bit vector. Identifies the sequence numbers of MPL multicast packets that the MPL forwarder has buffered. The sequence number is determined by  $w\text{-min} + i$ , where  $i$  is the offset within buffered-mpl-packets.

The MPL Window does not have any octet alignment requirement.

## 5. MPL Forwarder Behavior

An MPL forwarder implementation needs to manage sliding windows for each active MPL seed. The sliding window allows the MPL forwarder to determine what multicast packets to accept and what multicast packets are buffered. An MPL forwarder must also manage MPL packet transmissions.

### 5.1. Multicast Packet Dissemination

MPL uses the Trickle algorithm to control packet transmissions when disseminating MPL multicast packets [RFC6206]. MPL provides two propagation mechanisms for disseminating MPL multicast packets.

1. With proactive propagation, an MPL forwarder transmits buffered MPL multicast packets using the Trickle algorithm. This method is called proactive propagation since an MPL forwarder actively transmits MPL multicast packets without discovering that a neighboring MPL forwarder has yet to receive the message.
2. With reactive propagation, an MPL forwarder transmits ICMPv6 MPL messages using the Trickle algorithm. An MPL forwarder only transmits buffered MPL multicast packets upon discovering that neighboring devices have not yet to receive the corresponding MPL multicast packets.

When receiving a new multicast packet, an MPL forwarder first utilizes proactive propagation to forward the MPL multicast packet. Proactive propagation reduces dissemination latency since it does not require discovering that neighboring devices have not yet received the MPL multicast packet. MPL forwarders utilize proactive propagation for newly received MPL multicast packets since they can assume that some neighboring MPL forwarders have yet to receive the MPL multicast packet. After a limited number of MPL multicast packet transmissions, the MPL forwarder may terminate proactive propagation for the MPL multicast packet.

An MPL forwarder may optionally use reactive propagation to continue the dissemination process with lower communication overhead. With reactive propagation, neighboring MPL forwarders use ICMPv6 MPL messages to discover new MPL multicast messages that have not yet been received. When discovering that a neighboring MPL forwarder has not yet received a new MPL multicast packet, the MPL forwarder enables proactive propagation again.

#### 5.1.1. Trickle Parameters and Variables

As specified in RFC 6206 [RFC6206], a Trickle timer runs for a defined interval and has three configuration parameters: the minimum interval size  $I_{min}$ , the maximum interval size  $I_{max}$ , and a redundancy constant  $k$ .

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

Each MPL forwarder maintains a separate Trickle parameter set for the proactive and reactive propagation methods. `TimerExpirations` MUST be greater than 0 for proactive propagation. `TimerExpirations` MAY be set to 0 for reactive propagation, which effectively disables reactive propagation.

As specified in RFC 6206 [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.

#### 5.1.2. Proactive Propagation

With proactive propagation, the MPL forwarder transmits buffered MPL multicast packets using the Trickle algorithm. Each buffered MPL multicast packet that is proactively being disseminated with proactive propagation has an associated Trickle timer. Adhering to Section 5 of RFC 6206 [RFC6206], this document defines the following:

- o This document defines a "consistent" transmission for proactive propagation as receiving an MPL multicast packet that has the same MPL seed identifier and sequence number as a buffered MPL packet.
- o This document defines an "inconsistent" transmission for proactive propagation as receiving an MPL multicast packet that has the same MPL seed identifier, the M flag set, and has a sequence number less than the buffered MPL multicast packet's sequence number.
- o This document does not define any external "events".
- o This document defines both MPL multicast packets and ICMPv6 MPL multicast packets as Trickle messages. These messages are defined in the sections below.

- o The actions outside the Trickle algorithm that the protocol takes involve managing sliding window state, and is specified in Section 5.2.

#### 5.1.3. Reactive Propagation

With reactive propagation, the MPL forwarder transmits ICMPv6 MPL messages using the Trickle algorithm. A MPL forwarder maintains a single Trickle timer for reactive propagation with each MPL domain. When REACTIVE\_TIMER\_EXPIRATIONS is 0, the MPL forwarder does not execute the Trickle algorithm for reactive propagation and reactive propagation is disabled. Adhering to Section 5 of RFC 6206 [RFC6206], this document defines the following:

- o This document defines a "consistent" transmission for reactive propagation as receiving an ICMPv6 MPL message that indicates neither the receiving nor transmitting node has new MPL multicast packets to offer.
- o This document defines an "inconsistent" transmission for reactive propagation as receiving an ICMPv6 MPL message that indicates either the receiving or transmitting node has at least one new MPL multicast packet to offer.
- o This document defines an "event" for reactive propagation as updating any sliding window (i.e. changing the value of WindowMin, WindowMax, or the set of buffered MPL multicast packets) in response to receiving an MPL multicast packet.
- o This document defines both MPL multicast packets and ICMPv6 MPL multicast packets as Trickle messages. These messages are defined in the sections below.
- o The actions outside the Trickle algorithm that the protocol takes involve managing sliding window state, and is specified in Section 5.2.

#### 5.2. Sliding Windows

Every MPL forwarder MUST maintain a sliding window of sequence numbers for each MPL seed of recently received MPL packets. The sliding window performs two functions:

1. Indicate what MPL multicast packets the MPL forwarder should accept.
2. Indicate what MPL multicast packets are buffered and may be transmitted to neighboring MPL forwarders.

Each sliding window logically consists of:

1. A lower-bound sequence number, WindowMin, that represents the sequence number of the oldest MPL multicast packet the MPL forwarder is willing to receive or has buffered. An MPL forwarder MUST ignore any MPL multicast packet that has sequence value less than WindowMin.
2. An upper-bound sequence value, WindowMax, that represents the sequence number of the next MPL multicast packet that the MPL forwarder expects to receive. An MPL forwarder MUST accept any MPL multicast packet that has sequence number greater than or equal to WindowMax.
3. A list of MPL multicast packets, BufferedPackets, buffered by the MPL forwarder. Each entry in BufferedPackets MUST have a sequence number in the range [WindowMin, WindowMax).
4. A timer, HoldTimer, that indicates the minimum lifetime of the sliding window. The MPL forwarder MUST NOT free a sliding window before HoldTimer expires.

When receiving an MPL multicast packet, if no existing sliding window exists for the MPL seed, the MPL forwarder MUST create a new sliding window before accepting the MPL multicast packet. The MPL forwarder may reclaim memory resources by freeing a sliding window for another MPL seed if its HoldTimer has expired. If, for any reason, the MPL forwarder cannot create a new sliding window, it MUST discard the packet.

If a sliding window exists for the MPL seed, the MPL forwarder MUST ignore the MPL multicast packet if the packet's sequence number is less than WindowMin or appears in BufferedPackets. Otherwise, the MPL forwarder MUST accept the packet and determine whether or not to forward the packet and/or pass the packet to the next higher layer.

When accepting an MPL multicast packet, the MPL forwarder MUST update the sliding window based on the packet's sequence number. If the sequence number is not less than WindowMax, the MPL forwarder MUST set WindowMax to 1 greater than the packet's sequence number. If  $\text{WindowMax} - \text{WindowMin} > \text{MPL\_MAX\_WINDOW\_SIZE}$ , the MPL forwarder MUST increment WindowMin such that  $\text{WindowMax} - \text{WindowMin} \leq \text{MPL\_MAX\_WINDOW\_SIZE}$ . At the same time, the MPL forwarder MUST free any entries in BufferedPackets that have a sequence number less than WindowMin.

If the MPL forwarder has available memory resources, it MUST buffer the MPL multicast packet for proactive propagation. If not enough

memory resources are available to buffer the packet, the MPL forwarder **MUST** increment WindowMin and free entries in BufferedPackets that have a sequence number less than WindowMin until enough memory resources are available. Incrementing WindowMin will ensure that the MPL forwarder does not accept previously received packets.

An MPL forwarder **MAY** reclaim memory resources from sliding windows for other MPL seeds. If a sliding window for another MPL seed is actively disseminating messages and has more than one entry in its BufferedPackets, the MPL forwarder may free entries for that MPL seed by incrementing WindowMin as described above.

If the MPL forwarder cannot free enough memory resources to buffer the MPL multicast packet, the MPL forwarder **MUST** set WindowMin to 1 greater than the packet's sequence number.

When memory resources are available, an MPL forwarder **SHOULD** buffer a MPL multicast packet until the proactive propagation completes (i.e. the Trickle algorithm stops execution) and **MAY** buffer for longer. After proactive propagation completes, the MPL forwarder may advance WindowMin to the packet's sequence number to reclaim memory resources. When the MPL forwarder no longer buffers any packets, it **MAY** set WindowMin equal to WindowMax. When setting WindowMin equal to WindowMax, the MPL forwarder **MUST** initialize HoldTimer to WINDOW\_HOLD\_TIME and start HoldTimer. After HoldTimer expires, the MPL forwarder **MAY** free the sliding window to reclaim memory resources.

### 5.3. Transmission of MPL Multicast Packets

The MPL forwarder manages buffered MPL multicast packet transmissions using the Trickle algorithm. When adding a packet to BufferedPackets, the MPL forwarder **MUST** create a Trickle timer for the packet and start execution of the Trickle algorithm.

After PROACTIVE\_TIMER\_EXPIRATIONS Trickle timer events, the MPL forwarder **MUST** stop executing the Trickle algorithm. When a buffered MPL multicast packet does not have an active Trickle timer, the MPL forwarder **MAY** free the buffered packet by advancing WindowMin to 1 greater than the packet's sequence number.

Each interface that supports MPL is configured with exactly one MPL multicast scope. The MPL multicast scope **MUST** be site-local or smaller and defaults to link-local. A scope larger than link-local **MAY** be used only when that scope corresponds exactly to the MPL domain.

An MPL domain may therefore be composed of one or more MPL multicast scopes. For example, the MPL domain may be composed of a single MPL multicast scope when using a site-local scope. Alternatively, the MPL domain may be composed of multiple MPL multicast scopes when using a link-local scope.

IPv6-in-IPv6 encapsulation **MUST** be used when using MPL to forward an original multicast packet whose source or destination address is outside the MPL multicast scope. IPv6-in-IPv6 encapsulation is necessary to support Path MTU discovery when the MPL forwarder is not the source of the original multicast packet. IPv6-in-IPv6 encapsulation also allows an MPL forwarder to remove the MPL Option when forwarding the original multicast packet over a link that does not support MPL. The destination address scope for the outer IPv6 header **MUST** be the MPL multicast scope.

When an MPL domain is composed of multiple MPL multicast scopes (e.g. when the MPL multicast scope is link-local), an MPL forwarder **MUST** decapsulate and encapsulate the original multicast packet when crossing between different MPL multicast scopes. In doing so, the MPL forwarder **MUST** duplicate the MPL Option, unmodified, in the new outer IPv6 header.

The IPv6 destination address of the MPL multicast packet is the all-MPL-forwarders multicast address (TBD). The scope of the IPv6 destination address is set to the MPL multicast scope.

#### 5.4. Reception of MPL Multicast Packets

Upon receiving an MPL multicast packet, the MPL forwarder first determines whether or not to accept and buffer the MPL multicast packet based on its MPL seed and sequence value, as specified in Section 5.2.

If the MPL forwarder accepts the MPL multicast packet, the MPL forwarder determines whether or not to deliver the original multicast packet to the next higher layer. For example, if the MPL multicast packet uses IPv6-in-IPv6 encapsulation, the MPL forwarder removes the outer IPv6 header, which also removes MPL Option.

#### 5.5. Transmission of ICMPv6 MPL Messages

The MPL forwarder generates and transmits a new ICMPv6 MPL message whenever Trickle requests a transmission. The MPL forwarder includes an encoding of each sliding window in the ICMPv6 MPL message.

Each sliding window is encoded using an MPL Window entry, defined in Section 5.2. The MPL forwarder sets the MPL Window fields as

follows:

S If the MPL seed identifier is 0, set S to 0. If the MPL seed identifier is within the range [1, 65535], set S to 2. Otherwise, set S to 3.

w-min Set to the lower bound of the sliding window (i.e. WindowMin).

w-len Set to the length of the window (i.e. WindowMax - WindowMin).

seed-id If S is non-zero, set to the MPL seed identifier.

buffered-mpl-packets Set each bit that represents a sequence number of a packet in BufferedPackets to 1. Set all other bits to 0. The i'th bit in buffered-mpl-packets represents a sequence number of w-min + i.

#### 5.6. Reception of ICMPv6 MPL Messages

An MPL forwarder processes each ICMPv6 MPL message that it receives to determine if it has any new MPL multicast packets to receive or offer.

An MPL forwarder determines if a new MPL multicast packet has not been received from a neighboring node if any of the following conditions hold true:

1. The ICMPv6 MPL message includes an MPL Window for an MPL seed that does not have a corresponding sliding window entry on the MPL forwarder.
2. The neighbor has a packet in its BufferedPackets that has sequence value greater than or equal to WindowMax (i.e. w-min + w-len >= WindowMax).
3. The neighbor has a packet in its BufferedPackets that has sequence number within range of the sliding window but is not included in BufferedPackets (i.e. the i'th bit in buffered-mpl-packets is set to 1, where the sequence number is w-min + i).

When an MPL forwarder determines that it has not yet received a new MPL multicast packet buffered by a neighboring device, the MPL forwarder resets the Trickle timer associated with reactive propagation.

An MPL forwarder determines if an entry in BufferedPackets has not been received by a neighboring MPL forwarder if any of the following

conditions hold true:

1. The ICMPv6 MPL message does not include an MPL Window for the packet's MPL seed.
2. The packet's sequence number is greater than or equal to the neighbor's WindowMax value (i.e. the packet's sequence number is greater than or equal to  $w\text{-min} + w\text{-len}$ ).
3. The packet's sequence number is within the range of the neighbor's sliding window [WindowMin, WindowMax), but not included in the neighbor's BufferedPacket (i.e. the packet's sequence number is greater than or equal to  $w\text{-min}$ , strictly less than  $w\text{-min} + w\text{-len}$ , and the corresponding bit in buffered-mpl-packets is set to 0).

When an MPL forwarder determines that it has at least one buffered MPL multicast packet that has not yet been received by a neighbor, the MPL forwarder resets the Trickle timer associated with reactive propagation. Additionally, for each buffered MPL multicast packet that should be transferred, the MPL forwarder MUST reset the Trickle timer and reset  $e$  to 0 for proactive propagation. If the Trickle timer for proactive propagation has already stopped execution, the MPL forwarder MUST initialize a new Trickle timer and start execution of the Trickle algorithm.

## 6. MPL Parameters

An MPL forwarder maintains two sets of Trickle parameters for the proactive and reactive methods. The Trickle parameters are listed below:

PROACTIVE\_IMIN The minimum Trickle timer interval, as defined in [RFC6206] for proactive propagation.

PROACTIVE\_IMAX The maximum Trickle timer interval, as defined in [RFC6206] for proactive propagation.

PROACTIVE\_K The redundancy constant, as defined in [RFC6206] for proactive propagation.

PROACTIVE\_TIMER\_EXPIRATIONS The number of Trickle timer expirations that occur before terminating the Trickle algorithm. MUST be set to a value greater than 0.

REACTIVE\_IMIN The minimum Trickle timer interval, as defined in [RFC6206] for reactive propagation.

REACTIVE\_IMAX The maximum Trickle timer interval, as defined in [RFC6206] for reactive propagation.

REACTIVE\_K The redundancy constant, as defined in [RFC6206] for reactive propagation.

REACTIVE\_TIMER\_EXPIRATIONS The number of Trickle timer expirations that occur before terminating the Trickle algorithm. MAY be set to 0, which disables reactive propagation.

WINDOW\_HOLD\_TIME The minimum lifetime for sliding window state.

## 7. Acknowledgements

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

## 8. IANA Considerations

The Trickle Multicast option requires an IPv6 Option Number.

HEX	act	chg	rest
---	---	---	-----
C	01	0	TBD

The first two bits indicate that the IPv6 node MUST discard the packet if it doesn't recognize the option type, and the third bit indicates that the Option Data MUST NOT change en-route.

## 9. Security Considerations

TODO.

## 10. References

### 10.1. 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.
- [RFC2328] Moy, J., "OSPF Version 2", STD 54, RFC 2328, April 1998.
- [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.
- [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.
- [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

- [I-D.ietf-roll-terminology] Vasseur, J., "Terminology in Low power And Lossy Networks", draft-ietf-roll-terminology-06 (work in progress), September 2011.

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



Networking Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: April 23, 2013

J. Ko  
J. Jeong  
J. Park  
J. Jun  
N. Kim  
Electronics and  
Telecommunications Research  
Institute  
O. Gnawali  
University of Houston  
Oct 20, 2012

RPL Routing Pathology In a Network With a Mix of Nodes Operating in  
Storing and Non-Storing Modes  
draft-ko-roll-mix-network-pathology-01

Abstract

The RPL specification allows nodes running with storing or non-storing modes to operate in the same network. We describe how such a mix can result in network partitioning even when there are plenty of physical links available in the network. The partitioning affects both upwards (nodes to root) and downwards (root to leaf) traffic. This routing pathology stems from a recommendation made in the RPL specification forcing nodes with different modes of operation to join the RPL network as leaf nodes only. We propose a solution that modifies RPL by mandating that all the nodes parse and interpret source routing headers and storing mode nodes to sometimes act like a non-storing mode root by attaching source routing headers.

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 April 23, 2013.

## Copyright Notice

Copyright (c) 2012 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. Storing and Non-storing modes . . . . .	3
4. Routing Pathology . . . . .	4
5. Fixing the Pathology . . . . .	5
6. Acknowledgements . . . . .	6
7. IANA Considerations . . . . .	6
8. Security Considerations . . . . .	6
9. References . . . . .	7
9.1. Normative References . . . . .	7
9.2. Informative References . . . . .	7
Authors' Addresses . . . . .	7

## 1. Introduction

RPL [RFC6550] can operate in storing and non-storing modes. These modes introduce two different ways to perform downward routing. Downward routing is used when a node needs to send a packet to an arbitrary node (e.g., non-DODAG root node) in the network: the packet can go from a node "upward" towards the root and "downwards" to the final destination.

The RPL specification allows operating a network with a mix of storing and non-storing modes. RFC 6550 describes special rules to operate such a network: a node that operates with a different Mode of Operation (MOP) than the DODAG root will act as a leaf node in the network. The consensus was that it is unknown if the network would work properly because no one had designed such a network and was left to be explored in the future.

In this draft, we document a case in which we allow a mix of nodes operating in storing and non-storing modes to form a single network (e.g, despite having different MOPs) and introduce that RPL's two downwards routing modes, as it is, can cause a routing pathology. This pathology can partition the network, i.e., it can result in scenarios where nodes cannot send packets to the root and the root cannot send packets to the nodes even though these nodes have plenty of physical connectivity in the network.

We propose one approach of modifying RPL to prevent this routing pathology. The methodology, introducing a new mode of operation (MOP), has been implemented and tested on an LLN testbed and in process of publication. It is possible there are more elegant approaches to prevent the pathology described.

## 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 RFC 2119 [RFC2119].

The terminologies used in this document are consistent with the terminologies described in [I-D.ietf-roll-terminology], [RFC6551], and [RFC6550].

## 3. Storing and Non-storing modes

Before we describe the routing pathology that arises due to the

existence of a mix of nodes operating in storing and non-storing modes, we review the storing and non-storing downwards routing modes that RPL introduces.

In Storing mode, a node keeps a (not necessarily) complete list of (nodid, nexthop) for nodes in its subtree. When a node receives a packet, it forwards the packet to the nexthop if the node finds the destination in the list. If it does not find the destination in the list, it forwards the packet to the preferred parent.

In Non-storing mode, if a packet does not have routing path in the header, it forwards the packet to the preferred parent. The root in this mode collects and maintains topology information of the network. If the packet makes it to the root, the root computes the path to the destination based on this topology information. The root puts this path in the header and sends it to the next hop. The nodes, upon receiving a packet with a path in the header, forward the packet to the next hop as indicated in the path in the header.

#### 4. Routing Pathology

We first examine the effect of this routing pathology for routing collection traffic packets. Lets consider the following network topology.

A -> B -> N -> S -> Root (Storing)

Note that RPL indicates that, storing mode nodes and non-storing mode nodes use a different mode of operation (MOP) field. Furthermore, if the MOP supported from a DODAG root is not supported at a RPL node, the node can only participate in the RPL network as a leaf node. Say that the Root of the topology is a storing mode node. In this case, S (e.g., a storing mode node) can connect to the Root (a storing mode root) properly as a RPL router node. On the other hand, using the DIOs initiated at the Root, N (e.g., a non-storing mode node) will notice that the Root's MOP and its MOP is different; therefore, will only connect join the RPL network as a leaf node. As a result, A and B, which physically have connections to the root will not be able to join the RPL network. Thus, there is needless network partitioning. While this is an extreme case, other cases where using routes that non-storing mode nodes provide can help optimize the collection routes that RPL nodes form.

Next, we examine the routing pathology for downwards traffic packets. Lets consider the same topology as above.

Say that we eliminate the rule that RPL introduces of forcing nodes

with different MOPs to act as leaf nodes (e.g., no other modifications). In this case, A and B will be able to forward their collection traffic using N. Nevertheless, think of the case where N wants to send a packet to A. Since N is a non-storing mode node, N sends this packet to S because S is the preferred parent. S is operating in storing mode so it looks up node A in its forwarding table and finds that the next hop to reach A is using node N. With the assumption that node N will also know how to reach node A it will forward the packet back to node N. N is operating in non-storing mode so without a source routing header, it will forward the packet back to S. Thus the packet bounces between N and S. Optionally, when using the RPL routing headers, an ICMPv6 error message will be initiated.

With the increasing diversity of applications we can envision a network where a part of the network consists of computationally powerful nodes with route storing capabilities and the other part of the network with low-resource nodes that use a non-storing mode and operate together in a single RPL network. Furthermore, on a practical perspective, it is meaningful to use nodes that can contribute in constructing a more efficient DODAG that optimizes the data collection process rather than ignoring a node just because it supports a different MOP. Unfortunately, in such cases, the pathology that we discuss above can arise and cause downwards packets to be dropped and even more, restrict the formation of efficient collection routes.

## 5. Fixing the Pathology

We describe one way to fix RPL to prevent the pathology described above, while acknowledging that there might be more elegant solutions. In this approach, we acknowledge the fact that non-storing mode nodes are more likely to have strict resource limitations compared to nodes implementing the storing mode. Therefore, we make sure that the most of the required additional capabilities occur at the storing mode nodes rather than the non-storing mode nodes.

1. A new mode of operation (MOP) that allows a node to choose either to implement the storing or non-storing features, or both. The changes below are made compared to the original storing and non-storing modes.
2. Require storing and non-storing nodes to implement source routing header parsing capabilities.
3. RPL DODAG Root nodes supporting this MOP should have the capability to store routes (similar to the non-storing mode

option) and also identify storing mode node children nodes.

4. Non-storing nodes send hop-by-hop DAO.
5. Storing nodes keep a table of all the DAO senders and a flag indicating if each of those sender is operating in storing or non-storing mode. This requires allocating one of the bits in the DAO message for a node to indicate if it is operating in storing or non-storing mode.
6. Change the forwarding mechanism in the storing mode node when it receives a downward bound packet:
7.
  1. If packet does not have source routing header and the next hop is a storing-mode node, forward as in [RFC6550]. If the next hop is a non-storing node, insert the source routing header [RFC6554] into the packet and forward, i.e., act like a non-storing root.
  2. Using the flag indicating the storing status of nodes in its sub-DODAG, a node constructing a source routing header MAY choose to construct a source routing header only up to the next storing mode node.
  3. If the incoming packet has a source routing header, a storing mode node SHOULD obey the route specified in the source routing header to comply with the strict source routing requirements in [RFC6554].

If there is a mix of storing and non-storing nodes, we should also be more aggressive about loop detection. More aggressive loop detection will quickly remove the looping packets from the network. Even with the implementation of this suggestion, nodes beyond storing / non-storing nodes will still remain unreachable.

## 6. Acknowledgements

## 7. IANA Considerations

## 8. Security Considerations

Future work.

## 9. References

### 9.1. Normative References

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

### 9.2. Informative References

- [I-D.ietf-roll-terminology]  
Vasseur, J., "Terminology in Low power And Lossy Networks", draft-ietf-roll-terminology-05 (work in progress), March 2011.

## Authors' Addresses

JeongGil Ko  
Electronics and Telecommunications Research Institute  
218 Gajeong-Ro  
Yuseong-Gu, Daejeon 305-700  
Korea

Phone: +82-42-860-5824  
Email: jeonggil.ko@etri.re.kr

Jongsoo Jeong  
Electronics and Telecommunications Research Institute  
218 Gajeong-Ro  
Yuseong-Gu, Daejeon 305-700  
Korea

Phone: +82-42-860-1806  
Email: jsjeong@etri.re.kr

Jongjun Park  
Electronics and Telecommunications Research Institute  
218 Gajeong-Ro  
Yuseong-Gu, Daejeon 305-700  
Korea

Phone: +82-42-860-5413  
Email: juny@etri.re.kr

Jong Arm Jun  
Electronics and Telecommunications Research Institute  
218 Gajeong-Ro  
Yuseong-Gu, Daejeon 305-700  
Korea

Phone: +82-42-860-4835  
Email: jajun@etri.re.kr

Naesoo Kim  
Electronics and Telecommunications Research Institute  
218 Gajeong-Ro  
Yuseong-Gu, Daejeon 305-700  
Korea

Phone: +82-42-860-5214  
Email: nskim@etri.re.kr

Omprakash Gnawali  
University of Houston  
PGH 577, University of Houston  
Houston, TX 77204  
USA

Phone: +1-713-743-3356  
Email: gnawali@cs.uh.edu



ROLL  
Internet-Draft  
Intended status: Informational  
Expires: April 25, 2013

T. Phinney, Ed.  
consultant  
P. Thubert  
Cisco  
RA. Assimiti  
Nivis  
October 22, 2012

RPL applicability in industrial networks  
draft-phinney-roll-rpl-industrial-applicability-01

Abstract

The wide deployment of wireless devices, with their low installed cost (compared to wired devices), will significantly improve the productivity and safety of industrial plants. It will simultaneously increase the efficiency and safety of the plant's workers, by extending and making more timely the information set available about plant operations. The new Routing Protocol for Low Power and Lossy Networks (RPL) defines a Distance Vector protocol that is designed for such networks. The aim of this document is to analyze the applicability of that routing protocol in industrial LLNs formed of field devices.

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 April 25, 2013.

Copyright Notice

Copyright (c) 2012 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 . . . . .	4
1.1.	Requirements Language . . . . .	5
1.2.	Required Reading . . . . .	5
1.3.	Out of scope requirements . . . . .	5
2.	Deployment Scenario . . . . .	6
2.1.	Network Topologies . . . . .	8
2.1.1.	Traffic Characteristics . . . . .	8
2.1.2.	Topologies . . . . .	9
2.1.3.	Source-sink (SS) communication paradigm . . . . .	11
2.1.4.	Publish-subscribe (PS, or pub/sub) communication paradigm . . . . .	12
2.1.5.	Peer-to-peer (P2P) communication paradigm . . . . .	14
2.1.6.	Peer-to-multipeer (P2MP) communication paradigm . . . . .	15
2.1.7.	Additional considerations: Duocast and N-cast . . . . .	15
2.1.8.	RPL applicability per communication paradigm . . . . .	17
2.2.	Layer 2 applicability. . . . .	19
3.	Using RPL to Meet Functional Requirements . . . . .	20
4.	RPL Profile . . . . .	23
4.1.	RPL Features . . . . .	23
4.1.1.	RPL Instances . . . . .	23
4.1.2.	Storing vs. Non-Storing Mode . . . . .	25
4.1.3.	DAO Policy . . . . .	25
4.1.4.	Path Metrics . . . . .	26
4.1.5.	Objective Function . . . . .	26
4.1.6.	DODAG Repair . . . . .	26
4.1.7.	Multicast . . . . .	27
4.1.8.	Security . . . . .	27
4.1.9.	P2P communications . . . . .	27
4.2.	Layer-two features . . . . .	28
4.2.1.	Need layer-2 expert here. . . . .	28
4.2.2.	Security functions provided by layer-2. . . . .	28
4.2.3.	6LowPAN options assumed. . . . .	28
4.2.4.	MLE and other things . . . . .	28
4.3.	Recommended Configuration Defaults and Ranges . . . . .	28
4.3.1.	Trickle Parameters . . . . .	28
4.3.2.	Other Parameters . . . . .	29

5. Manageability Considerations . . . . .	30
6. Security Considerations . . . . .	31
6.1. Security Considerations during initial deployment . . . .	31
6.2. Security Considerations during incremental deployment . .	31
7. Other Related Protocols . . . . .	32
8. IANA Considerations . . . . .	33
9. Acknowledgements . . . . .	34
10. References . . . . .	35
10.1. Normative References . . . . .	35
10.2. Informative References . . . . .	35
10.3. External Informative References . . . . .	36
Authors' Addresses . . . . .	37

## 1. Introduction

Information Technology (IT) is already, and increasingly will be applied to Industrial Automation and Control System (IACS) technology in application areas where those IT technologies can be constrained sufficiently by Service Level Agreements (SLA) or other modest change that they are able to meet the operational needs of IACS. When that happens, the IACS benefits from the large intellectual, experiential and training investment that has already occurred in those IT precursors. One can conclude that future reuse of additional IT protocols for IACS will continue to occur due to the significant intellectual, experiential and training economies which result from that reuse.

Following that logic, many vendors are already extending or replacing their local field-bus technology with Ethernet and IP-based solutions. Examples of this evolution include CIP EtherNet/IP, Modbus/TCP, Foundation Fieldbus HSE, PROFINet and Invensys/Foxboro FOXnet. At the same time, wireless, low power field devices are being introduced that facilitate a significant increase in the amount of information which industrial users can collect and the number of control points that can be remotely managed.

IPv6 appears as a core technology at the conjunction of both trends, as illustrated by the current [ISA100.11a] industrial Wireless Sensor Networking (WSN) specification, where layers 1-4 technologies developed for end uses other than IACS - IEEE 802.15.4 PHY and MAC, 6LoWPAN and IPv6, and UDP - are adapted to IACS use. But due to the lack of open standards for routing in Low power and Lossy Networks (LLN) at the time ISA100.11a was crafted, routing was accomplished at the link layer and is specific to that standard.

The IETF ROLL Working Group has defined application-specific routing requirements for a LLN routing protocol, specified in:

Routing Requirements for Urban LLNs [RFC5548],  
Industrial Routing Requirements in LLNs [RFC5673],  
Home Automation Routing Requirements in LLNs [RFC5826], and  
Building Automation Routing Requirements in LLNs [RFC5867].

The Routing Protocol for Low Power and Lossy Networks (RPL) [I-D.ietf-roll-rpl] specification and its point to point extension/optimization [I-D.ietf-roll-p2p-rpl] define a generic Distance Vector protocol that is adapted to a variety of Low Power and Lossy Networks (LLN) types by the application of specific Objective Functions (OFs).

RPL forms Destination Oriented Directed Acyclic Graphs (DODAGs) within instances of the protocol, each instance being associated with an Objective Function to form a routing topology.

A field device that belongs to an instance uses the OF to determine which DODAG and which Version of that DODAG the device should join. The device also uses the OF to select a number of routers within the DODAG current and subsequent Versions to serve as parents or as feasible successors. A new Version of the DODAG is periodically reconstructed to enable a global reoptimization of the graph.

A RPL OF states the outcome of the process used by a RPL node to select and optimize routes within a RPL Instance based on the information objects available. The separation of OFs from the core protocol specification allows RPL to be adapted to meet the different optimization criteria required by the wide range of industrial classes of traffic and applications.

This document provides information on how RPL can accommodate the industrial requirements for LLNs, in particular as specified in [RFC5673].

#### 1.1. 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].

Additionally, this document uses terminology from [I-D.ietf-roll-terminology], and uses usual terminology from the Process Control and Factory Automation industries, some of which is recapitulated below:

FEC: Forward error correction

IACS: Industrial automation and control systems

RAND: reasonable and non-discriminatory (relative to licensing of patents)

#### 1.2. Required Reading

#### 1.3. Out of scope requirements

This applicability statement does not address requirements related to wireless LLNs employed in factory automation and related applications.

## 2. Deployment Scenario

[RFC5673] describes in detail the routing requirements for industrial LLNs. This RFC provides information on the varying deployment scenarios for such LLNs and how RPL assists in meeting those requirements.

Large industrial plants, or major operating areas within such plants, repeatedly go through four major phases, each of which typically lasts from months to years:

P1: Construction or major modification phase

P2: Planned startup phase

P3: Normal operation phase

P4: Planned shutdown phase

followed eventually by an (at least theoretical)

P5: Plant decommissioning phase.

It is also likely, after a major catastrophe at a plant, to have a

P6: Post-emergency recovery and repair phase.

The deployment scenarios for wireless LLN devices may be different in each of these phases. In particular, during the Construction or major modification phase (P1), LLN devices may be installed months before the intended LLN can become usefully operational (because needed routers and infrastructure devices are not yet installed or active), and there are likely to be many personnel in whom the plant owner/operator has only limited trust, such as subcontractors and others in the plant area who have undergone only a cursory background investigation (if any at all). In general, during this phase, plant instrumentation is not yet operational, so could be removed and replaced by a Trojaned device without much likelihood of physical detection of the substitution. Thus physical security of LLN devices is generally a more significant risk factor during this phase than once the plant is operational, where simple replacement of device electronics is detectable.

Extra LLN devices and even extra LLN subnets may be employed during Planned startup (P2) and Planned shutdown (P4) phases, in support of the task of transitioning the plant or plant area between operational and shutdown states. The extra devices typically provide extra monitoring as the plant transitions infrequent activity states. (In

many continuous process plants, up to 2x extra staff are employed at monitoring and control workstations during these two phases, precisely because the plant is undergoing extraordinary behavior as it transitions to or from its steady-state operational condition.)

Similar transient devices and subnets may be used during an unscheduled Post-emergency recovery and repair phase (P6) of operation, but in that case the extra devices usually are routers substituting for plant LLN devices that have been damaged by the incident (such as a fire, explosion, flood, tornado or hurricane) that induced the emergency.

The Planned startup (P2) and Planned shutdown (P4) phases are similar in many respects, but the LLN environment of the two can be quite different, since the Planned shutdown phase can assume that the stable LLN environment used for Normal operation (P3) is functional during shutdown, whereas that stable environment usually is still being established during startup.

The Post-emergency recovery and repair phase (P6) typically operates in an LLN environment that is somewhere between that of the Planned startup (P2) and Normal operation (P3) phases, but with an indeterminate number of temporary routers placed to facilitate communication across and around the area affected by the catastrophe.

Smaller industrial plants and sites may go through similar phases, but often commingle the phases because, in those smaller plants, the phases require less planning and structuring of personnel responsibilities and thus permit less formalization and partitioning of the operating scenarios. For example, it is much simpler, and usually requires much less planning, to bring new equipment on a skid into a plant, using a forklift, than to lay temporary railroad track or employ an extended-axle heavy haul tractor-trailer to deliver a multi-ton process vessel, and temporarily deploy and use very large heavy-lift cranes to install it. In the former cases, nearby equipment usually can continue normal operation while the installation proceeds; in the latter case that is almost always impossible, due to safety and other concerns.

The domain of applicability for the RPL protocol may include all phases but the Normal Operation phase, where the bandwidth allocation and the routes are usually optimized by an external Path Computing Engine (PCE), e.g. an ISA100.11a System Manager.

Additionally, it could be envisioned to include RPL in the normal operation provided that a new Objective Function is defined that actually interacts with the PCE in order to establish the reference topology, in which case RPL operations would only apply to emergency

repair actions. when the reference topology becomes unusable for some failure, and as long as the problem persists.

## 2.1. Network Topologies

### 2.1.1. Traffic Characteristics

The industrial market classifies process applications into three broad categories and six classes.

- o Safety
  - \* Class 0: Emergency action - Always a critical function
- o Control
  - \* Class 1: Closed loop regulatory control - Often a critical function
  - \* Class 2: Closed loop supervisory control - Usually non-critical function
  - \* Class 3: Open loop control - Operator takes action and controls the actuator (human in the loop)
- o Monitoring
  - \* Class 4: Alerting - Short-term operational effect (for example event-based maintenance)
  - \* Class 5: Logging and downloading / uploading - No immediate operational consequence (e.g., history collection, sequence-of-events, preventive maintenance)

Safety critical functions effect the basic safety integrity of the plant. These normally dormant functions kick in only when process control systems, or their operators, have failed. By design and by regular interval inspection, they have a well-understood probability of failure on demand in the range of typically once per 10-1000 years.

In-time deliveries of messages becomes more relevant as the class number decreases.

Note that for a control application, the jitter is just as important as latency and has a potential of destabilizing control algorithms.

The domain of applicability for the RPL protocol probably matches the

range of classes where industrial users are interested in deploying wireless networks. This domain includes monitoring classes (4 and 5), and the non-critical portions of control classes (2 and 3). RPL might also be considered as an additional repair mechanism in all situations, and independently of the flow classification and the medium type.

It appears from the above sections that whether and the way RPL can be applied for a given flow depends both on the deployment scenario and on the class of application / traffic. At a high level, this can be summarized by the following matrix:

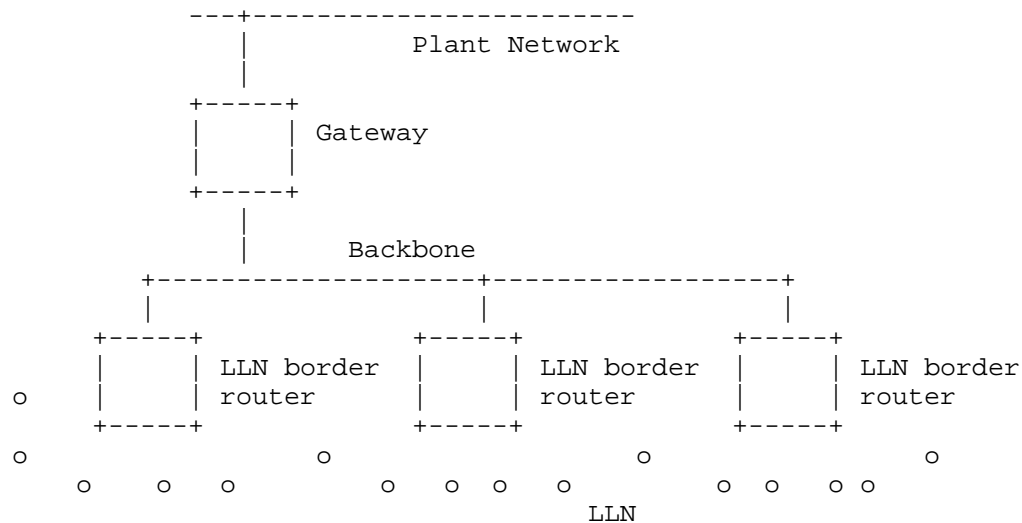
Phase \ Class	0	1	2	3	4	5
Construction			X	X	X	X
Planned startup			X	X	X	X
Normal operation				?	?	?
Planned shutdown			X	X	X	X
Plant decommissioning			X	X	X	X
Recovery and repair	X	X	X	X	X	X

? : typically usable for all but higher-rate classes 0,1 PS traffic

Figure 1: RPL applicability matrix

### 2.1.2. Topologies

In an IACS, high-rate communications flows (e.g., 1 Hz or 4 Hz for a traditional process automation network) typically are such that only a single wireless LLN hop separates the source device from a LLN Border Router (LBR) to a significantly higher data-rate backbone network, typically based on IEEE 802.3, IEEE 802.11, or IEEE 802.16, as illustrated in Figure 2.

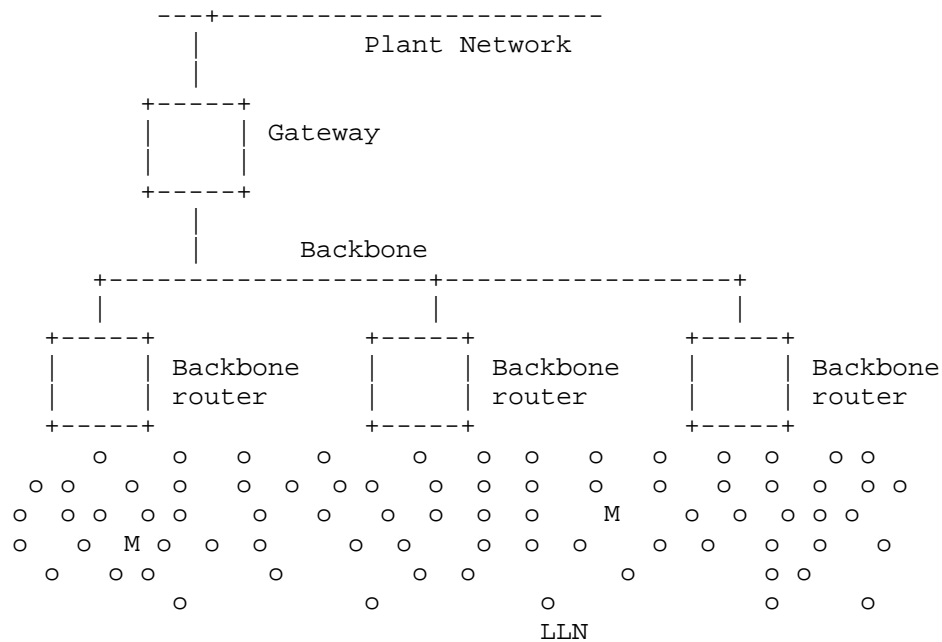


o : stationary wireless field device, seldom acting as an LLN router

Figure 2: High-rate low-delay low-variance IACS topology

For factory automation networks, the basic communications cycle for control is typically much faster, on the order of 100 Hz or more. In this case the LLN itself may be based on high-data-rate IEEE 802.11 or a 100 Mbit/s or faster optical link, and the higher-rate network used by the LBRs to connect the LLN to superior automation equipment typically might be based on fiber-optic IEEE 802.3, with multiple LBRs around the periphery of the factory area, so that most high-rate communications again requires only a single wireless LLN hop.

Multi-hop LLN routing is used within the LLN portion of such networks to provide backup communications paths when primary single-hop LLN paths fail, or for lower repetition rate communications where longer LLN transit times and higher variance are not an issue. Typically, the majority of devices in an IACS can tolerate such higher-delay higher-variance paths, so routing choices often are driven by energy considerations for the affected devices, rather than simply by IACS performance requirements, as illustrated in Figure 3.



o : stationary wireless field device, often acting as an LLN router  
 M : mobile wireless device

Figure 3: Low-rate higher-delay higher-variance IACS topology

Two decades of experience with digital fieldbuses has shown that four communications paradigms dominate in IACS:

SS: Source-sink

PS: Publish-subscribe

P2P: Peer-to-peer

P2MP: Peer-to-multipeer

### 2.1.3. Source-sink (SS) communication paradigm

In SS, the source-sink communication paradigm, each of many devices in one set, S1, sends UDP-like messages, usually infrequently and intermittently, to a second set of devices, S2, determined by a common multicast address. A typical example would be that all devices within a given process unit N are configured to send process alarm messages to the multicast address `Receivers_of_process_alarms_for_unit_N`. Receiving devices, typically

on non-LLN networks accessed via LBRs, are configured to receive such multicast messages if their work assignment covers process unit N, and not otherwise.

Timeliness of message delivery is a significant aspect of some SS communication. When the SS traffic conveys process alarms or device alerts, there is often a contractual requirement, and sometimes even a regulatory requirement, on the maximum end-to-end transit delay of the SS message, including both the LLN and non-LLN components of that delay. However, there is no requirement on relative jitter in the delivery of multiple SS messages from the same source, and message reordering during transit is irrelevant.

Within the LLN, the SS paradigm simply requires that messages so addressed be forwarded to the responsible LBR (or set of equivalent LBRs) for further forwarding outside the LLN. Within the LLN such traffic typically is device-to-LBR or device-to-redundant-set-of-equivalent-LBRs. In general, SS traffic may be aggregated before forwarding when both the multicast destination address and other QoS attributes are identical. If information on the target delivery times for SS messages is available to the aggregating forwarding device, that device may intentionally delay forwarding somewhat to facilitate further aggregation, which can significantly reduce LLN alarm-reporting traffic during major plant upset events.

#### 2.1.4. Publish-subscribe (PS, or pub/sub) communication paradigm

In PS, the publish-subscribe communication paradigm, a device sends UDP-like messages, usually periodically or cyclicly (i.e., repetitively but without fixed periodicity), to a single multicast address derived from or correlated with the device's own address. A typical example would be that each sensor and actuator device within a given process unit N is configured to send process state messages to the multicast address that designates its specific publications. In essence the derived multicast address for device D is Receivers\_of\_publications\_by\_device\_D. Typically those receivers are in two categories: controllers (C) for control loops in which device D participates, and devices accessed via the LLN's LBRs that monitor and/or accumulate historical information about device D's status and outputs.

If the controller(s) that receive device D's publication are all outside the LLN and accessed by LBRs, then within the LLN such traffic typically is device-to-LBR or device-to-redundant-set-of-equivalent-LBRs. But if a controller (Cn) is within the LLN, then a number of different LLN-local traffic patterns may be employed, depending on the capabilities of the underlying link technology and on configured performance requirements for such reporting. Typically

in such a case, publication by device D is forwarded up a DODAG to an LLN router that is also on a downward DODAG to a destination controller Cn, then forwarded down that second DODAG to that destination controller Cn. Of course, if the LLN router (or even the LBR) is itself the intended destination controller, which will often be the case, then no downward forwarding occurs.

Timeliness of message delivery is a critical aspect of PS communication. Individual messages can be lost without significant impact on the controlled physical process, but typically a sequence of four consecutive lost messages will trigger fallback behavior of the control algorithms, which is considered a system failure by most system owner/operators. (In general, and unless a local catastrophic event such as a major explosion or a tornado occurs in the plant, invocation of more than one instance of such fallback handling per year, per plant, is considered unacceptable.)

Message loss, delay and jitter in delivery of PS messaging is a relative matter. PS messaging is used for transfer of process measurements and associated status from sensors to control computation elements, from control computation elements to actuators, and of current commanded position and status from actuators back to control computation elements. The actual time interval of interest is that which starts with sensing of the physical process (which necessarily occurs before the sensed value can be sent in the first message) and which ends when the computed control correction is applied to the physical process by the appropriate actuator (which cannot occur until after the second message containing the computed control output has been received by that actuator). With rare exception, the control algorithms used with PS messaging in the process automation industries - those managing continuous material flows - rely on fixed-period sampling, computation and transfer of outputs, while those in the factory automation industries - those managing discrete manufacturing operations - rely on bounded delay between sampling of inputs, control computation and transfer of outputs to physical actuators that affect the controlled process.

Deliberately manipulated message delay and jitter in delivery of PS messaging has the potential to destabilize control loops. It is the responsibility of conveyed higher-level protocols to protect against such potential security attacks by detecting overly delayed or jittered messages at delivery, converting them into instances of message loss. Thus network and data-link protocols such as IPv6 and Ethernet need not themselves address such issues, although their selection and employment should take the existence (or lack) of such higher-layer protection mechanisms, and the resulting consequences due to excessive delay and jitter, into consideration in their parameterization.

In general, PS traffic within the LLN is not aggregated before forwarding, to minimize message loss and delay in reception by any relevant controller(s) that are outside the LLN. However, if all intended destination controllers are within the LLN, and at least one of those intended controllers also serves as an LLN router on a DODAG to off-LLN destinations that all are not controllers, then the router functions in that device may aggregate PS traffic before forwarding when the required routing and other QoS attributes are identical. If information on the target delivery times for PS messages to non-controller devices is available to the aggregating forwarding device, that device may intentionally delay forwarding somewhat to facilitate further aggregation.

In some system architectures, message streams that use PS to convey current process measurements and status are compressed at the source through a 2-dimensional winnowing process that compares

- 1) the process measurement values and status of the about-to-be-sent message with that of the last actually-sent message, and
- 2) the current time vs. the queueing time for the last actually-sent message.

If the interval since that last-sent message is less than a predefined maximum time, and the status is unchanged, and the process measurement(s) conveyed in the message is within predefined deadband(s) of the last-sent measurement value(s), then transmission of the new message is suppressed. Often this suppression takes the form of not queuing the new message for transmission, but in some protocols a brief placeholder message indicating "no significant change" is queued in its stead.

#### 2.1.5. Peer-to-peer (P2P) communication paradigm

In P2P, the peer-to-peer communication paradigm, a device sends UDP-like or TCP-like messages from one device (D1) to a second device (D2), usually with bidirectional but asymmetric flow of application data, where the amount of data is significantly greater in one direction than the other. Typical examples are transfer of configuration information to or from a process field device, or transfer of captured process diagnostics (e.g., time-stamped noise signatures from a coriolis flowmeter) to an off-LLN higher-level asset management system. Unicast addressing is used in both directions of data flow.

In general, specific P2P traffic has only loose timeliness requirements, typically just those required so that response times to human-operator-initiated actions meet human factors requirements. As

a consequence, in general, message aggregation is permitted, although few opportunities are likely to present themselves for such aggregation due to the sporadic nature of such messaging to a single destination, and/or due to the large message payloads that often occur in at least one direction of transmission.

#### 2.1.6. Peer-to-multipeer (P2MP) communication paradigm

In P2MP, the peer-to-multipeer communication paradigm, a device sends UDP-like messages downward, from one device (D1) to a set of other devices (Dn). Typical examples are bulk downloads to a set of devices that use identical code image segments or identically-structured database segments; group commands to enable device state transitions that are quasi-synchronized across all or part of the local network (e.g., switch to the next set of point-to-point downloaded session keys, or notifying that the network is switching to an emergency repair and recovery mode); etc. Multicast addressing is used in the downward direction of data flow.

Devices can be assigned to a number of multicast groups, for instance by device type. Then, if it becomes necessary to reflash all devices of a given type with a new load image, a multicast distribution mechanism can be leveraged to optimize the distribution operation.

In general, P2MP traffic has only loose timeliness requirements. As a consequence, in general, message aggregation is permitted, although few opportunities are likely to present themselves for such aggregation due to the sporadic nature of such messaging to a single multicast group destination, and/or due to the large message payloads that often occur when P2MP is used for group downloads. However, in general, message aggregation negatively impacts the delivery success rate for each of the aggregated messages, since the probability of error in a received message increases with message length. Together these considerations often lead to a policy of non-aggregation for P2MP messaging.

Note: Reliable group download protocols, such as the no-longer-published IEEE 802.1E (ISO/IEC 15802-4) system load protocol, and reliable multicast protocols based on the guidance of RFC2887, are instructive in how P2MP can be used for initial bulk download, followed by either P2MP or P2P selective retransmissions for missed download segments.

#### 2.1.7. Additional considerations: Duocast and N-cast

In industrial automation systems, some traffic is from (relatively) high-rate monitoring and control loops, of Class 0 and Class 1 as described in [RFC5673]. In such systems, the wireless link protocol,

which typically uses immediate in-band acknowledgement to confirm delivery (or, on failure, conclude that a retransmission is required), can be adapted to attempt simultaneous delivery to more than one receiving device, with separated, sequenced immediate in-band acknowledgement by each of those intended receivers. (This mechanism is known colloquially as "duocast" (for two intended receivers), or more generically as "N-cast" (for N intended receivers).) Transmission is deemed successful if at least one such immediate acknowledgement is received by the sending device; otherwise the device queues the message for retransmission, up until the maximum configured number of retries has been attempted.

The logic behind duocast/N-cast is very simple: In wireless systems without FEC (forward error correction), the overall rate of success for transactions consisting of an initial transmission and an immediate acknowledgement is typically 95%. In other words, 5% of such transactions fail, either because the initial message of the transaction is not received correctly by the intended receiver, or because the immediate acknowledgment by that receiver is not received correctly by the transaction initiator.

In the generalized case of N-cast, where any received acknowledgement serves to complete the transaction, and where the N intended receivers are spatially diverse, physically separated from each other by multiple wavelengths, the probability that all such receivers fail to receive the initial message of the transaction, or that all generated immediate acknowledgements are not received by the transaction initiator, is typically approximately  $(5\%)^N$ . Thus, for duocast, the expected success rate for a single transaction goes from 95% ( $1.0 - 0.05$ ) to 99.75% ( $1.0 - 0.05^2$ ), to 99.9875% ( $1.0 - 0.05^3$ ) when  $N=3$ , and even higher when  $N>3$ .

From the above analysis, it is obvious that the primary benefit of N-cast occurs when N goes from  $N=1$  (unicast) to  $N=2$  (duocast); the reduction in transaction loss rate for increasing  $N>2$  is quite small, and for  $N>3$  it is infinitesimal. In the typical industrial automation environment of class 1 process control loops, which typically repeat at a 1 Hz or 4 Hz rate, in a very large process plant with thousands of field devices reporting at that rate, the maximum number of transmission retries that must be planned, and for which capacity must be scheduled (within the requisite 250 ms or 1 s interval) is seven (7) retries for unicast PS reporting, but only three (3) retries with duocast PS reporting. (This is determined by the requirement to not miss four successive reports more than once per year, across the entire plant, as such a loss typically triggers fallback behavior in the controlled loop, which is considered a failure of the wireless system by the plant owner/operator.) In practice, the enormous reduction in both planned and used

retransmission capacity provided by duocast/N-cast is what enables 4 Hz loops to be supported in large wireless systems.

When available, duocast/N-cast typically is used only for one-hop PS traffic on Class 1 and Class 0 control loops. It may also be employed for rapid, reliable one-hop delivery of Class 0 and sometimes Class 1 process alarms and device alerts, which use the SS paradigm. Because it requires scheduling of multiple receivers that are prepared to acknowledge the received message during the transaction, in general it is not appropriate for the other types of traffic in such systems - P2P and P2MP - and is not needed for other classes of control loops or other types of traffic, which do not have such stringent reporting requirements.

Note: Although there are known patent applications for duocast and N-cast, at the time of this writing the patent assignee, Honeywell International, has offered to permit cost-free RAND use in those industrial wireless standards that have chosen to employ the technology, under a reciprocal licensing requirement relative to that use. Since duocast and N-cast provide performance and energy optimizations, they are not essential for use in wireless systems. However, in practice, their use makes it possible to support 4 Hz wireless loops and meet sub-second safety alarm reporting requirements in large plants, where that might otherwise be impractical without use of a wired network. When duocast/N-cast is not employed, the wireless retransmission capacity that is needed to support such fast loops often is excessive, typically over 100x that actually used for retransmission (i.e., providing for seven retries per transaction when the mean number used is only 0.06 retries).

#### 2.1.8. RPL applicability per communication paradigm

To match the requirements above, RPL provides a number of RPL Modes of Operation (MOP):

No downward route: defined in [I-D.ietf-roll-rpl], section 6.3.1, MOP of 0. This mode allows only upward routing, that is from nodes (devices) that reside inside the RPL network toward the outside via the DODAG root.

Non-storing mode: defined in [I-D.ietf-roll-rpl], section 6.3.1, MOP of 1. This mode improves MOP 0 by adding the capability to use source routing from the root towards registered targets within the instance DODAG.

Storing mode without multicast support: defined in [I-D.ietf-roll-rpl], section 6.3.1, MOP of 2. This mode improves MOP 0 by adding the capability to use stateful routing from the root towards registered targets within the instance DODAG.

Storing mode with link-scope multicast DAO: defined in [I-D.ietf-roll-rpl] section 9.10, this mode improves MOP 2 by adding the capability to send Destination Advertisements to all nodes over a single Layer 2 link (e.g. a wireless hop) and enables line-of-sight direct communication.

Storing mode with multicast support: defined in [I-D.ietf-roll-rpl], Mode-of-operation (MOP) of 3. This mode improves MOP 2 by adding the capability to register multicast groups and perform multicast forwarding along the instance DODAG (or a spanning subtree within the DODAG).

Reactive: defined in [I-D.ietf-roll-p2p-rpl], the reactive mode creates on-demand additional DAGs that are used to reach a given node acting as DODAG root within a certain number of hops. This mode can typically be used for an ad-hoc closed-loop communication.

The RPL MOP that can be applied for a given flow depends on the communication paradigm. It must be noted that a DODAG that is used for PS traffic can also be used for SS traffic since the MOP 2 extends the MOP 0, and that a DODAG that is used for P2MP distribution can also be used for downward PS since the MOP 3 extends the MOP 2.

On the other hand, an Objective Function (OF) that optimizes metrics for a pure upwards DODAG might differ from the OF that optimizes a mixed upward and downward DODAG.

As a result, it can be expected that different RPL instances are installed with different OFs, different channel allocations, etc... that result in different routing and forwarding topologies, sometimes with differing delay vs. energy profiles, optimized separately for the different flows at hand.

This can be broadly summarized in the following table:

Paradigm\RPL MOP	RPL spec	Mode of operation
Peer-to-peer	RPL P2P	reactive (on-demand)
P2P line-of-sight	RPL base	2 (storing) with multicast DAO
P2MP distribution	RPL base	3 (storing with multicast)
Publish-subscribe	RPL base	1 or 2 (storing or not-storing)
Source-sink	RPL base	0 (no downward route)
N-cast publish	RPL base	0 (no downward route)

Figure 4: RPL applicability per communication paradigm

## 2.2. Layer 2 applicability.

To be completed.

### 3. Using RPL to Meet Functional Requirements

The functional requirements for most industrial automation deployments are similar to those listed in [RFC5673]:

The routing protocol MUST be capable of supporting the organization of a large number of nodes into regions, usually corresponding to partitions of the automated process, each containing on the order of 30 to 3000 nodes.

The routing protocol MUST provide mechanisms to support configuration of the routing protocol itself.

The routing protocol MUST provide mechanisms to support instructed configuration of explicit routing, so that in the absence of failure the routing used for selected flow classes is that which has been remotely configured (typically by a centralized configurator). In such circumstances RPL is used

- for local network repair;

- for flow classes to which explicit routing has not been assigned;

- during bootstrapping of the network itself (which is really just an instance of routing without such an externally-imposed assignment).

The routing protocol SHOULD support directed flows with different QoS characteristics, typically with different energy vs. delay tradeoffs, for traffic directed to LBRs. In practice only two such sets of QoS are relevant:

- one that emphasizes energy minimization for energy-constrained nodes at the expense of greater mean transit delay and variance in transit delay; and

- one that emphasizes minimization of mean transit delay and transit delay variance at the expense of greater energy demand on originating and intermediary energy-constrained nodes, typically used for critical SS traffic (e.e., infrequent and unpredictable safety alarms with legally-mandated maximum reporting delays) and critical PS traffic (e.g., predictable periodic (for process automation) or cyclic (for factory automation) high-speed safety control loops needed to protect life, the environment, and/or critical national infrastructure assets).

In the absence of configured routing, or when such routes have failed, the routing protocol MUST dynamically compute and select effective routes composed of low-power and lossy links. Local network dynamics SHOULD NOT impact the entire network. The routing protocol MUST compute multiple paths when possible.

The routing protocol MUST support multicast addressing, including

- multicast originating with a LBR or off the LLN, directed to a predefined group within the LLN

- multicast originating within the LLN, directed to one or more equivalent LBRs, in support of SS traffic

- multicast originating within the LLN, directed to one or more equivalent LBRs, in support of PS traffic, including all three of the following situations:

- 1: <to be added>

- 2: <to be added>

- 3: <to be added>

The routing protocol SHOULD support and utilize a large number of highly directed flows to a few LBRs, to handle scalability.

The routing protocol SHOULD support formation of groups of field devices in the network.

The routing protocol NEED NOT support anycast addressing because, as of the date of writing of this document, such addressing is not used by automation and control field devices. In general, no two such devices are equivalent, except perhaps for intermediary LBRs, so unicast suffices for situations where anycast might otherwise be employed.

RPL supports:

- Large-scale networks characterized by highly directed traffic flows between each field device and servers close to the head-end of the automation network. To this end, RPL builds Directed Acyclic Graphs (DAGs) rooted at LBRs.

- Zero-touch configuration. This is done through in-band methods for configuring RPL variables using DIO messages.

The use of links with time-varying availability and quality characteristics. This is accomplished by allowing the use of metrics that effectively capture the quality of a path (e.g., in terms of the mean and maximum impact of use of that path on packet delivery timing and on endpoint energy demands), and by limiting the impact of changing local conditions by discovering and maintaining multiple DAG parents, and by using local repair mechanisms when DAG links break.

For wireless installations of small size with undemanding communication requirements, RPL is likely to generate satisfactory routing without any special effort. However, in larger installations or where timeliness considerations do not permit multi-second wireless-subnet transit times, then flow labeling is likely required so that forwarding routers can make informed tradeoffs between conserving their own energy resources and meeting overall system needs.

#### 4. RPL Profile

This section outlines a RPL profile for a representative deployment in a process control application. Process monitoring without control is typically less demanding, so a subset of this profile generally will suffice.

##### 4.1. RPL Features

###### 4.1.1. RPL Instances

RPL allows formation of multiple instances that operate independently of each other. Each instance may use a different objective function and different modes of operation. It is highly recommended that wireless field devices participate in different instances that utilize objective functions that meet different optimization goals. These optimization goals target: 1) Minimizing and ensuring that a guaranteed latency is being met 2) Maximizing the communication reliability of the packets transferred over the wireless media 3) Minimizing aggregate power consumption for multi-hop LLNs that are composed of battery powered field devices. Some of these optimization goals will have to be met concurrently in a single instance by imposing various constraints. Each wireless field device should participate in a set composed of a minimum of three instances that meet optimization goals associated with three traffic flows which need to be supported by all industrial LLNs. Management Instance: Wireless industrial networks are highly deterministic in nature, meaning that wireless field devices do not make any decisions locally but are managed by a centralized System Manager that oversees the join process as well as all communication and security settings present in the devices. The management traffic flow is downward traffic and needs to meet strictly enforced latency and reliability requirements in order to ensure proper operation of the wireless LLN. Hence each field device should participate in an instance dedicated to management traffic. All decisions made while constructing this instance will need to be approved by the Path Computaton Engine present in the System Manager due to the deterministic, centralized nature of wireless industrial LLNs. Shallow LLNs with a hop count of up to one, accommodate this downward traffic using non-storing mode. Non-storing involves source routing that is detrimental to the packet size. For large transfers such as image download and configuration files, this can be factorized for a large packet. In that case, a method such as draft-thubert-roll-forwarding-frags-00 is required over multi-hop networks to forward and recover individual fragments without the overhead of the source route information in each fragment. If the hop count in the wireless LLN grows (LLN becomes deeper) it is higly recommended that the management instance rely on storing mode in order to relay management related packets.

Operational Instance: The bulk of the data that is transferred over wireless LLN consists of process automation related payloads. This data is of paramount importance to the smooth operation of the process that is being monitored. Hence data reliability is of paramount importance. It is also important to note that a vast majority of the wireless field devices that operate in industrial LLNs are battery powered. The operational instance should hence ensure high reliability of the data transmitted while also minimizing the aggregate power consumption of the field devices operating in the LLN. All decisions made while constructing this instance will need to be approved by the Path Computaton Engine present in the System Manager. This is due to the deterministic, centralized nature of wireless LLNs. Autonomous instance: An autonomous instance requires limited to no configuration. It, primary purpose is to serve as a backup for the operational instance in case the operational instance fails. It is also useful in non-production phases of the network, when the plant is installed or dismantled.

[draft-thubert-roll-asymlink] provides rules and mechanisms whereby an instance can be used as a fallback to another upon failure to forward a packet further. The autonomic instance should always be active and during normal operations it should be maintained through local repair mechanisms. In normal operation global repairs should be sparingly employed in order to conserve batteries. But a global repair is also probably the fastest and most economical technique in the case the network is extensively damaged. It is recommended to rely on automation that will trigger a global repair upon the detection of a large scale incident such as an explosion or a crash. As the name suggests, the autonomous instance is formed without any dependence on the System Manager. Decisions made during the construction of the autonomous instance do not need approval from the Path Computation Engine present in the in the System Manager. Participation of each wireless field device in at least one instance that hosts a DODAG with a virtual root is highly recommended. Wireless industrial networks are typically composed of multiple LLNs that terminate in a LLN Border Router (LBR). The LBRs communicate with each other and with other entities present on the backbone (such as the Gateway and the System Manager) over a wired or wireless backbone infrastructure. When a device A that operates in LLN 1 sends a packet to a device B that operates in LLN2, the packets egresses LLN1 through LBR1 and ingresses LLN2 through LBR2 after travelling over the backbone infrastructure that connects the LBRs. In order to accommodate this packet flow that travels from one LLN to another, it is highly recommended that wireless field devices participate in at least one instance that has a DODAG with a virtual root.

#### 4.1.2. Storing vs. Non-Storing Mode

In general, storing mode is required for high-reporting-rate devices (where "high rate" is with respect to the underlying link data conveyance capability). Such devices, in the absence of path failure, are typically only one hop from the LBR(s) that convey their messaging to other parts of the system. Fortunately, in such cases, the routing tables required by such nodes are small, even when they include information on DODAGs that are used as backup alternate routes.

Deeper multi-hop wireless LLNs (hop count > 1) should support storing mode in order to minimize the overhead associated with source routing given the limited header capacity associated with typical physical layers employed in wireless LLNs. Support for storing mode requires additional RAM resources be present in the constrained wireless field devices. Typical wireless LLNs scale to a maximum of one hundred field devices. Hence the appropriate RAM resources for supporting storing mode should be part of the hardware requirements imposed upon wireless field devices during the design phase.

The ISA100.11a standard mandates that all LBRs maintain routing tables with enough capacity to accommodate operation in storing mode. The standard also mandates that all wireless field devices maintain routing tables but it does not make any capacity assumptions, allowing for null routing tables. The System Manager should read the routing table capacity of each wireless field router and LBR during their join phase, and determine if support for storing mode in a particular LLN is feasible.

Lack of support for storing mode is also detrimental to battery operated wireless field devices due to the power consumption associated with transporting the hefty headers associated with source routing. Support for storing mode also ensures path redundancy which in turn allows for better prediction of the latency associated with downward traffic flows. Guaranteed latencies are of paramount importance for various traffic flows in wireless industrial LLNs.

#### 4.1.3. DAO Policy

Support for both upward and downward traffic flows is a requirement in industrial automation systems. As a result, nodes send DAO messages to establish downward paths from the root to themselves. DAO messages are not acknowledged in wireless industrial LLNs that are composed of battery operated field devices in order to minimize the power consumption overhead associated with path discovery. Given that wireless field devices in LLNs will typically participate in multiple RPL instances and DODAGs, it is highly recommended that both

the RPLInstance ID and the DODAGID be included in the DAO.

#### 4.1.4. Path Metrics

RPL relies on an Objective Function for selecting parents and computing path costs and rank. This objective function is decoupled from the core RPL mechanisms and also from the metrics in use in the network. Two objective functions for RPL have been defined at the time of this writing, OF0 and MRHOF, both of which define the selection of a preferred parent and backup parents, and are suitable for industrial automation network deployments.

#### 4.1.5. Objective Function

Industrial wireless LLNs are subject to swift variations in terms of the propagation of the wireless signal, variations that can affect the quality of the links between field devices. This is due to the nature of the environment in which they operate which can be characterized as metal jungles that cause wireless propagation distortions, multi-path fading and scattering. Hence support for hysteresis is needed in order to ensure relative link stability which in turn ensures route stability.

As mentioned in previous sections of this document, different traffic flows require different optimization goals. Wireless field devices should participate in multiple instances associated with multiple objective functions. Management instance: Should utilize an objective function that focuses on optimization of latency and data reliability. Operational instance: Should utilize an objective function that focuses on data reliability and minimizing aggregate power consumption for battery operated field devices. Autonomous instance: Should utilize an objective function that optimizes data latency. The primary purpose of the autonomous instance is as a fallback instance in case the operational instance fails. Data latency is hence paramount for ensuring that the wireless field devices can exchange packets in order to repair the operational instance.

More complex objective functions are needed that take in consideration multiple constraints and utilize weighted sums of multiple additive and multiplicative metrics. Additional objective functions specifically designed for such networks may be defined in companion RFCs.

#### 4.1.6. DODAG Repair

To effectively handle time-varying link characteristics and availability, industrial automation network deployments SHOULD

utilize the local repair mechanisms in RPL.

Local repair is triggered by broken link detection, and in storing mode also by loop detection.

The first local repair mechanism consists of a node detaching from a DODAG and then re-attaching to the same or to a different DODAG at a later time. While detached, a node advertises an infinite rank value so that its children can select a different parent. This process is known as poisoning and is described in Section 8.2.2.5 of [I-D.ietf-roll-rpl]. While RPL provides an option to form a local DODAG, doing so in industrial automation network deployments is of little benefit since applications typically communicate through a LBR. After the detached node has made sufficient effort to send notification to its children that it is detached, the node can rejoin the same DODAG with a higher rank value. The configured duration of the poisoning mechanism needs to take into account the disconnection time applications running over the network can tolerate. Note that when joining a different DODAG, the node need not perform poisoning.

The second local repair mechanism controls how much a node can increase its rank within a given DODAG Version (e.g., after detaching from the DODAG as a result of broken link or loop detection). Setting the DAGMaxRankIncrease to a non-zero value enables this mechanism, and setting it to a value of less than infinity limits the cost of count-to-infinity scenarios when they occur, thus controlling the duration of disconnection applications may experience.

#### 4.1.7. Multicast

#### 4.1.8. Security

Industrial automation network deployments typically operate in areas that provide limited physical security (relative to the risk of attack). For this reason, the link layer, transport layer and application layer technologies utilized within such networks typically provide security mechanisms to ensure authentication, confidentiality, integrity, timeliness and freshness. As a result, such deployments may not need to implement RPL's security mechanisms and could rely on link layer and higher layer security features.

#### 4.1.9. P2P communications

<to be added>

#### 4.2. Layer-two features

4.2.1. Need layer-2 expert here.

4.2.2. Security functions provided by layer-2.

4.2.3. 6LowPAN options assumed.

4.2.4. MLE and other things

#### 4.3. Recommended Configuration Defaults and Ranges

##### 4.3.1. Trickle Parameters

Trickle was designed to be density-aware and perform well in networks characterized by a wide range of node densities. The combination of DIO packet suppression and adaptive timers for sending updates allows Trickle to perform well in both sparse and dense environments.

Node densities in industrial automation network deployments can vary greatly, from nodes having only one or a handful of neighbors to nodes having several hundred neighbors. In high density environments, relatively low values for *Imin* may cause a short period of congestion when an inconsistency is detected and DIO updates are sent by a large number of neighboring nodes nearly simultaneously. While the Trickle timer will exponentially backoff, some time may elapse before the congestion subsides. Although some link layers employ contention mechanisms that attempt to avoid congestion, relying solely on the link layer to avoid congestion caused by a large number of DIO updates can result in increased communication latency for other control and data traffic in the network.

To mitigate this kind of short-term congestion, this document recommends a more conservative set of values for the Trickle parameters than those specified in [RFC6206]. In particular, *DIOIntervalMin* is set to a larger value to avoid periods of congestion in dense environments, and *DIORefundancyConstant* is parameterized accordingly as described below. These values are appropriate for the timely distribution of DIO updates in both sparse and dense scenarios while avoiding the short-term congestion that might arise in dense scenarios.

Because the actual link capacity depends on the particular link technology used within an industrial automation network deployment, the Trickle parameters are specified in terms of the link's maximum capacity for conveying link-local multicast messages. If the link can convey *m* link-local multicast packets per second on average, the expected time it takes to transmit a link-local multicast packet is

1/m seconds.

DIOIntervalMin: Industrial automation network deployments SHOULD set DIOIntervalMin such that the Trickle Imin is at least 50 times as long as it takes to convey a link-local multicast packet. This value is larger than that recommended in [RFC6206] to avoid congestion in dense plant deployments as described above.

DIOIntervalDoublings: Industrial automation network deployments SHOULD set DIOIntervalDoublings such that the Trickle Imax is at least TBD minutes or more.

DIORedundancyConstant: Industrial automation network deployments SHOULD set DIORedundancyConstant to a value of at least 10. This is due to the larger chosen value for DIOIntervalMin and the proportional relationship between Imin and k suggested in [RFC6206]. This increase is intended to compensate for the increased communication latency of DIO updates caused by the increase in the DIOIntervalMin value, though the proportional relationship between Imin and k suggested in [RFC6206] is not preserved. Instead, DIORedundancyConstant is set to a lower value in order to reduce the number of packet transmissions in dense environments.

#### 4.3.2. Other Parameters

<to be added>

## 5. Manageability Considerations

RPL enables automatic and consistent configuration of RPL routers through parameters specified by the DODAG root and disseminated through DIO packets. The use of Trickle for scheduling DIO transmissions ensures lightweight yet timely propagation of important network and parameter updates and allows network operators to choose the trade-off point they are comfortable with respect to overhead vs. reliability and timeliness of network updates.

The metrics in use in the network along with the Trickle Timer parameters used to control the frequency and redundancy of network updates can be dynamically varied by the root during the lifetime of the network. To that end, all DIO messages SHOULD contain a Metric Container option for disseminating the metrics and metric values used for DODAG setup. In addition, DIO messages SHOULD contain a DODAG Configuration option for disseminating the Trickle Timer parameters throughout the network.

The possibility of dynamically updating the metrics in use in the network as well as the frequency of network updates allows deployment characteristics (e.g., network density) to be discovered during network bring-up and to be used to tailor network parameters once the network is operational rather than having to rely on precise pre-configuration. This also allows the network parameters and the overall routing protocol behavior to evolve during the lifetime of the network.

RPL specifies a number of variables and events that can be tracked for purposes of network fault and performance monitoring of RPL routers. Depending on the memory and processing capabilities of each smart grid device, various subsets of these can be employed in the field.

## 6. Security Considerations

Industrial automation network deployments typically operate in areas that provide limited physical security (relative to the risk of attack). For this reason, the link layer, transport layer and application layer technologies utilized within such networks typically provide security mechanisms to ensure authentication, confidentiality, integrity, timeliness and freshness. As a result, such deployments may not need to implement RPL's security mechanisms and could rely on link layer and higher layer security features.

This document does not specify operations that could introduce new threats. Security considerations for RPL deployments are to be developed in accordance with recommendations laid out in, for example, [I-D.tsao-roll-security-framework].

Industrial automation networks are subject to stringent security requirements as they are considered a critical infrastructure component. At the same time, since they are composed of large numbers of resource- constrained devices inter-connected with limited-throughput links, many available security mechanisms are not practical for use in such networks. As a result, the choice of security mechanisms is highly dependent on the device and network capabilities characterizing a particular deployment.

In contrast to other types of LLNs, in industrial automation networks centralized administrative control and access to a permanent secure infrastructure is available. As a result link-layer, transport-layer and/or application-layer security mechanisms are typically in place and may make use of RPL's secure mode unnecessary.

### 6.1. Security Considerations during initial deployment

### 6.2. Security Considerations during incremental deployment

## 7. Other Related Protocols

## 8. IANA Considerations

This specification has no requirement on IANA.

## 9. Acknowledgements

## 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.

### 10.2. Informative References

- [I-D.ietf-roll-rpl]  
Brandt, A., Vasseur, J., Hui, J., Pister, K., Thubert, P., Levis, P., Struik, R., Kelsey, R., Clausen, T., and T. Winter, "RPL: IPv6 Routing Protocol for Low power and Lossy Networks", draft-ietf-roll-rpl-19 (work in progress), March 2011.
- [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-14 (work in progress), October 2012.
- [I-D.ietf-roll-terminology]  
Vasseur, J., "Terminology in Low power And Lossy Networks", draft-ietf-roll-terminology-06 (work in progress), September 2011.
- [RFC5548] Dohler, M., Watteyne, T., Winter, T., and D. Barthel, "Routing Requirements for Urban Low-Power and Lossy Networks", RFC 5548, May 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.
- [RFC5673] Pister, K., Thubert, P., Dwars, S., and T. Phinney, "Industrial Routing Requirements in Low-Power and Lossy Networks", RFC 5673, October 2009.
- [I-D.ietf-roll-of0]  
Thubert, P., "RPL Objective Function Zero", draft-ietf-roll-of0-20 (work in progress), September 2011.
- [I-D.tsao-roll-security-framework]

Tsao, T., Alexander, R., Daza, V., and A. Lozano, "A Security Framework for Routing over Low Power and Lossy Networks", draft-tsao-roll-security-framework-02 (work in progress), March 2010.

### 10.3. External Informative References

[HART] [www.hartcomm.org](http://www.hartcomm.org), "Highway Addressable Remote Transducer, a group of specifications for industrial process and control devices administered by the HART Foundation".

[ISA100.11a] ISA, "ISA100, Wireless Systems for Automation", May 2008, <<http://www.isa.org/Community/SP100WirelessSystemsforAutomation>>.

Authors' Addresses

Tom Phinney (editor)  
consultant  
5012 W. Torrey Pines Circle  
Glendale, AZ 85308-3221  
USA

Phone: +1 602 938 3163  
Email: tom.phinney@cox.net

Pascal Thubert  
Cisco Systems  
Village d'Entreprises Green Side  
400, Avenue de Roumanille  
Batiment T3  
Biot - Sophia Antipolis 06410  
FRANCE

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

Robert Assimiti  
Nivis  
1000 Circle 75 Parkway SE, Ste 300  
Atlanta, GA 30339  
USA

Phone: +1 678 202 6859  
Email: robert.assimiti@nivis.com



Network Working Group  
Internet-Draft  
Intended status: Informational  
Expires: February 1, 2013

M. Richardson  
SSW  
July 31, 2012

ROLL Applicability Statement Template  
draft-richardson-roll-applicability-template-00

Abstract

This document is a template applicability statement for the Routing over Low-power and Lossy Networks (ROLL) WG.

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 February 1, 2013.

Copyright Notice

Copyright (c) 2012 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 . . . . .	4
1.1. Requirements Language . . . . .	4
1.2. Required Reading . . . . .	4
1.3. Out of scope requirements . . . . .	4
2. Deployment Scenario . . . . .	5
2.1. Network Topologies . . . . .	5
2.2. Network Topologies . . . . .	5
2.2.1. Traffic Characteristics . . . . .	5
2.2.2. General . . . . .	5
2.2.3. Source-sink (SS) communication paradigm . . . . .	5
2.2.4. Publish-subscribe (PS, or pub/sub) communication paradigm . . . . .	5
2.2.5. Peer-to-peer (P2P) communication paradigm . . . . .	5
2.2.6. Peer-to-multipeer (P2MP) communication paradigm . . . . .	5
2.2.7. Additional considerations: Duocast and N-cast . . . . .	5
2.2.8. RPL applicability per communication paradigm . . . . .	5
2.3. Layer 2 applicability. . . . .	5
3. Using RPL to Meet Functional Requirements . . . . .	6
4. RPL Profile . . . . .	7
4.1. RPL Features . . . . .	7
4.1.1. RPL Instances . . . . .	7
4.1.2. Storing vs. Non-Storing Mode . . . . .	7
4.1.3. DAO Policy . . . . .	7
4.1.4. Path Metrics . . . . .	7
4.1.5. Objective Function . . . . .	7
4.1.6. DODAG Repair . . . . .	7
4.1.7. Multicast . . . . .	7
4.1.8. Security . . . . .	7
4.1.9. P2P communications . . . . .	7
4.2. Layer-two features . . . . .	7
4.2.1. Need layer-2 expert here. . . . .	7
4.2.2. Security functions provided by layer-2. . . . .	7
4.2.3. 6LoWPAN options assumed. . . . .	7
4.2.4. MLE and other things . . . . .	7
4.3. Recommended Configuration Defaults and Ranges . . . . .	7
4.3.1. Trickle Parameters . . . . .	7
4.3.2. Other Parameters . . . . .	7
5. Manageability Considerations . . . . .	8
6. Security Considerations . . . . .	9
6.1. Security Considerations during initial deployment . . . . .	9
6.2. Security Considerations during incremental deployment . . . . .	9
7. Other Related Protocols . . . . .	10
8. IANA Considerations . . . . .	11
9. Acknowledgements . . . . .	12
10. References . . . . .	13
10.1. Informative References . . . . .	13

10.2. Normative References . . . . .	13
11. Normative references . . . . .	14
Author's Address . . . . .	15

## 1. Introduction

Hello.

### 1.1. Requirements Language

(RFC2119 reference)

### 1.2. Required Reading

References/Overview of requirements documents, both IETF and industry group. (two pages maximum. This text should be (very) technical, should be aimed at IETF \*participants\*, not industry group participants, and should explain this industries' specific issues)

### 1.3. Out of scope requirements

This should list other documents (if any) which deal with situations where things are not in scope for this document.

(For instance, the AMI document tries to cover both line-powered urban metering networks, and energy-constrained metering networks, and also tries to deal with rural requirements. This should be three or four documents, so this section should list the limits of what this document covers)

## 2. Deployment Scenario

### 2.1. Network Topologies

describe a single scenario, with possibly multiple topologies that a single utility would employ.

### 2.2. Network Topologies

#### 2.2.1. Traffic Characteristics

Explain what kind of traffic is being transmitted, where it is initiated, and what kinds of protocols (CoAP, multicast, HTTPS, etc.) are being used. Explain what assumptions are being made about authentication and authorization in those protocols.

#### 2.2.2. General

#### 2.2.3. Source-sink (SS) communication paradigm

#### 2.2.4. Publish-subscribe (PS, or pub/sub) communication paradigm

#### 2.2.5. Peer-to-peer (P2P) communication paradigm

#### 2.2.6. Peer-to-multipeer (P2MP) communication paradigm

#### 2.2.7. Additional considerations: Duocast and N-cast

#### 2.2.8. RPL applicability per communication paradigm

### 2.3. Layer 2 applicability.

Explain what layer-2 technologies this statement applies to, and if there are options, they should be listed generally here, and specifically in section 4.2.

### 3. Using RPL to Meet Functional Requirements

This should explain in general terms how RPL is going to be used in this network topology. If trees that are multiple layers deep are expected, then this should be described so that the fan out is understood. Some sample topologies (from simulations) should be explained, perhaps with images references from other publications.

This section should tell an \*implementer\* in a lab, having a simulation tool or a building/city/etc. to use as a testbed, how to construct an LLN of sufficient complexity (but not too much) to validate an implementation.

#### 4. RPL Profile

This section should list the various features of RPL plus other layers of the LLN, and how they will be used.

##### 4.1. RPL Features

###### 4.1.1. RPL Instances

###### 4.1.2. Storing vs. Non-Storing Mode

###### 4.1.3. DAO Policy

###### 4.1.4. Path Metrics

###### 4.1.5. Objective Function

###### 4.1.6. DODAG Repair

###### 4.1.7. Multicast

###### 4.1.8. Security

###### 4.1.9. P2P communications

##### 4.2. Layer-two features

###### 4.2.1. Need layer-2 expert here.

###### 4.2.2. Security functions provided by layer-2.

###### 4.2.3. 6LowPAN options assumed.

###### 4.2.4. MLE and other things

##### 4.3. Recommended Configuration Defaults and Ranges

###### 4.3.1. Trickle Parameters

###### 4.3.2. Other Parameters

## 5. Manageability Considerations

## 6. Security Considerations

### 6.1. Security Considerations during initial deployment

(This section explains how nodes get their initial trust anchors, initial network keys. It explains if this happens at the factory, in a deployment truck, if it is done in the field, perhaps like <http://www.lix.polytechnique.fr/hipercom/SmartObjectSecurity/papers/CullenJennings.pdf>)

### 6.2. Security Considerations during incremental deployment

(This section explains how that replaces a failed node takes on the dead nodes' identity, or not. How are nodes retired. How are nodes removed if they are compromised)

## 7. Other Related Protocols

## 8. IANA Considerations

## 9. Acknowledgements

## 10. References

### 10.1. Informative References

### 10.2. Normative References

11. Normative references

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

Author's Address

Michael C. Richardson  
Sandelman Software Works  
470 Dawson Avenue  
Ottawa, ON K1Z 5V7  
CA

Email: [mcr+ietf@sandelman.ca](mailto:mcr+ietf@sandelman.ca)  
URI: <http://www.sandelman.ca/>

