

Internet Engineering Task Force
Internet-Draft
Intended status: Experimental
Expires: April 15, 2013

M. Goyal
University of Wisconsin
Milwaukee
D. Barthel
France Telecom Orange
E. Baccelli
INRIA
October 12, 2012

DIS Modifications
draft-goyal-roll-dis-modifications-01

Abstract

This document specifies the DIS flags and options that allow an RPL node to control how neighbor RPL routers respond to its solicitation for DIOs.

Status of this Memo

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

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

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

This Internet-Draft will expire on April 15, 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

Internet-Draft [draft-goyal-roll-dis-modifications-01](#)

October 2012

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.	New Flags in the DIS Base Object	4
4.	DIS Options	4
4.1.	Metric Container	5
4.2.	Response Spreading Option	5
5.	Applications	6
5.1.	A Leaf Node Joining a DAG	6
5.2.	Identifying A Defunct DAG	6
6.	IANA Considerations	8
6.1.	DIS Flags	8
6.2.	RPL Control Message Options	9
7.	Security Considerations	9
8.	References	9
8.1.	Normative References	9
8.2.	Informative References	9
	Authors' Addresses	10

1. Introduction

An RPL node can use a DODAG Information Solicitation (DIS) message to solicit DODAG Information Object (DIO) messages from its neighbor RPL routers. A DIS may carry a Solicited Information option that specifies the predicates of the DAG(s) the node is interested in. In the absence of a Solicited Information option, it is assumed that the node generating the DIS is interested in receiving DIOs for all the DAGs. A DIS can be multicast to all the in-range routers or it can be unicast to a specific neighbor router. RPL requires a router to consider the receipt of a multicast DIS as an inconsistency and hence reset its Trickle timers [[RFC6206](#)] for the matching DAGs. The receipt of a unicast DIS causes an RPL router to generate the DIOs for all the matching DAGs without resetting the Trickle timers.

Consider an RPL leaf node that desires to join a certain DAG. This node can either wait for its neighbor RPL routers to voluntarily transmit DIOs or it can proactively solicit DIOs using a DIS message. Voluntary DIO transmissions may happen after a very long time if the network is stable and the Trickle timer intervals have reached large values. Thus, proactively seeking DIOs using a DIS may be the only reasonable option. Since the node does not know which neighbor routers belong to the DAG, it must solicit the DIOs using a multicast DIS (with predicates of the desired DAG specified inside a Solicited Information option). On receiving this DIS, the neighbor routers that belong to the desired DAG will reset their Trickle timers and quickly transmit their DIOs. The downside of resetting Trickle timers is that the routers would continue to transmit the DIOs frequently for a considerable time interval. These DIO transmissions are unnecessary, consume precious energy and may contribute to congestion in the network.

There are other scenarios where resetting of Trickle timer following the receipt of a multicast DIS is not appropriate. For example, consider an RPL router that desires to free up memory by deleting state for the defunct DAGs it belongs to. Identifying a defunct DAG

This document defines two new flags inside the DIS base object:

- o "No Inconsistency" (N) flag: On receiving a unicast/multicast DIS with N flag set, an RPL router MUST NOT reset the Trickle timers for the matching DAGs. Also, a DIO generated in response to a DIS with N flag set MUST always contain a Configuration option.
- o "DIO Type" (T) flag: This flag specifies whether the responding routers should transmit a multicast DIO or a unicast one. The responding router MUST transmit a multicast DIO if this flag is set.

The modified DIS base object is shown in Figure 1.

4. DIS Options

4.1. Metric Container

In order to limit the number of routers that will respond to a multicast DIS, this document allows the specification of routing constraints inside a DIS that a router must satisfy in order to respond to the DIS. These routing constraints are specified inside a Metric Container option contained in the DIS. Thus, this document allows the inclusion of a Metric Container option inside a DIS. An RPL router that receives a DIS with a Metric Container option **MUST** ignore any Metric object in it, and **MUST** evaluate the "mandatory" Constraint objects in it by comparing the constraint value to the aggregated value of the corresponding routing metric that the router maintains for the matching DAG(s). The aggregated routing metric values **MUST** satisfy all the mandatory constraints in order for the router to generate DIOs for the matching DAG(s).

4.2. Response Spreading Option

0										1										2									
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3						
+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-						
Type = 0x0A										Length										Spread. Inter.									

+--+

Figure 2: The Response Spreading Option

A multicast DIS may lead to a large number of RPL routers responding with DIOs. Concurrent transmissions by multiple routers are not desirable. Such transmissions may end up in collisions. Unicast DIOs may be able to avail of link-level retransmissions. However, multicast DIOs have no such protection. These transmissions and retransmissions may also cause congestion in the network. To avoid such problems, this document specifies an optional DIO response spreading mechanism.

This document defines a new RPL control message option called "Response Spreading", shown in Figure 2, with a recommended Type value 0x0A (to be confirmed by IANA). A Response Spreading option may be included only inside a multicast DIS message. An RPL router that responds to a multicast DIS, that includes a Response Spreading option, MUST wait for a time uniformly chosen in the interval $[0..2^{\text{SpreadingInterval}}]$, expressed in ms, before attempting to transmit its DIO. If the DIS does not include a Response Spreading option, the node is free to transmit the DIO as it otherwise would.

[5.](#) Applications

This section details two example mechanisms that use the DIS flags and options defined in this document. The first mechanism describes how a leaf node may join a desired DAG. The second mechanism details how a node may identify defunct DAGs for which it still maintains state.

[5.1.](#) A Leaf Node Joining a DAG

A new leaf node that joins an established LLN runs an iterative algorithm in which it requests (using multicast DIS) DIOs from routers belonging to the desired DAG. The DIS message has the "No Inconsistency" flag set (to prevent resetting of Trickle timer in responding routers) and the "DIO Type" flag reset (to make responding

routers send unicast DIOs back). The DIS message can include a Response Spreading option listing a suitable spreading interval and a Metric Container listing the routing constraints that the responding routers must satisfy. In each iteration, the node multicasts such a DIS and waits for the DIOs. Once the spreading interval has expired, the node considers the current iteration to be unsuccessful. Now the node relaxes the routing constraints somewhat and proceeds to the next iteration. The cycle repeats until the node receives one or more DIOs in a particular iteration or if maximum number of iterations have been reached.

[5.2.](#) Identifying A Defunct DAG

An RPL node may remove a neighbor from its parent set for a DAG for a number of reasons:

- o The neighbor is no longer reachable, as determined using a mechanism such as Neighbor Unreachability Detection (NUD) [[RFC4861](#)], Bidirectional Forwarding Detection (BFD) [[RFC5881](#)] or L2 triggers [[RFC5184](#)]; or
- o The neighbor advertises an infinite rank in the DAG; or
- o Keeping the neighbor as a parent would require the node to increase its rank beyond $L + \text{DAGMaxRankIncrease}$, where L is the minimum rank the node has had in this DAG; or
- o The neighbor advertises membership in a different DAG within the same RPL Instance, where a different DAG is recognised by a different DODAGID or a different DODAGVersionNumber.

Even if the conditions listed above exist, an RPL node may fail to remove a neighbor from its parent set because:

- o The node may fail to receive the neighbor's DIOs advertising an increased rank or the neighbor's membership in a different DAG;
- o The node may not check, and hence may not detect, the neighbor's unreachability for a long time. For example, the node may not have any data to send to this neighbor and hence may not encounter any event (such as failure to send data to this neighbor) that would trigger a check for the neighbor's reachability.

In such cases, a node would continue to consider itself attached to a DAG even if all its parents in the DAG are unreachable or have moved to different DAGs. Such a DAG can be characterized as being defunct from the node's perspective. If the node maintains state about a large number of defunct DAGs, such state may prevent a considerable portion of the total memory in the node from being available for more useful purposes.

To alleviate the problem described above, an RPL node may invoke the following procedure to identify a defunct DAG and delete the state it maintains for this DAG. Note that, given the proactive nature of RPL protocol, the lack of data traffic using a DAG can not be considered a reliable indication of the DAG's defunction. Further, the Trickle timer based control of DIO transmissions means the possibility of an indefinite delay in the receipt of a new DIO from a functional DAG parent. Hence, the mechanism described next is based on the use of a DIS message to solicit DIOs about a DAG suspected of defunction. Further, a multicast DIS is used so as to avoid the need to query each parent individually and also to discover other neighbor routers that may serve as the node's new parents in the DAG.

When an RPL node has not received a DIO from any of its parents in a DAG for more than a locally configured time duration:

- o The node generates a multicast DIS message with:
 - * "No Inconsistency" flag set so that the responding routers do not reset their Trickle timers.
 - * "DIO Type" flag set so that the responding routers send multicast DIOs and other nodes in the vicinity do not need to invoke this procedure.
 - * A Solicited Information option to identify the DAG in question. This option must have the I and D flags set and the RPLInstanceID/DODAGID fields must be set to values identifying the DAG. The V flag inside the Solicited Information option should not be set so as to allow the neighbors to send DIOs advertising the latest version of the DAG.

- * A Response Spreading option specifying a suitable time interval

over which the DIO responses may arrive.

- o After sending the DIS, the node waits for the duration specified inside the Response Spreading option to receive the DIOs generated by its neighbors. At the conclusion of the wait duration:
 - * If the node has received one or more DIOs advertising newer version(s) of the DAG, it joins the latest version of the DAG, selects a new parent set among the neighbors advertising the latest DAG version and marks the DAG status as functional.
 - * Otherwise, if the node has not received a DIO advertising the current version of the DAG from a neighbor in the parent set, it removes that neighbor from the parent set. As a result, if the node has no parent left in the DAG, it marks the DAG as defunct and schedule the deletion of the state it has maintained for the DAG after a locally configured "hold" duration. (This is because, as per RPL specification, when a node no longer has any parents left in a DAG, it is still required to remember the DAG's identity (RPLInstanceID, DODAGID, DODAGVersionNumber), the lowest rank (L) it has had in this DAG and the DAGMaxRankIncrease value for the DAG for a certain time interval to ensure that the node does not join an earlier version of the DAG and does not rejoin the current version of the DAG at a rank higher than $L + \text{DAGMaxRankIncrease}$.)

[6.](#) IANA Considerations

[6.1.](#) DIS Flags

IANA is requested to allocate bits 6 and 7 of the DIS Flag Field to become the "No Inconsistency" and "DIO Type" bits, the functionality of which is described in [Section 3](#) of this document.

Value	Meaning	Reference
6	No Inconsistency	This document
7	DIO Type	This document

[6.2.](#) RPL Control Message Options

IANA is requested to allocate a new code point in the "RPL Control Message Options" registry for the "Response Spreading" option, the behavior of which is described in [Section 4.2](#).

Value	Meaning	Reference
0x0A	Response Spreading	This document

RPL Control Message Options

[7.](#) Security Considerations

TBA

[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.
- [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.

[8.2.](#) Informative References

- [RFC4861] Narten, T., Nordmark, E., Simpson, W., and H. Soliman, "Neighbor Discovery for IP version 6 (IPv6)", [RFC 4861](#), September 2007.
- [RFC5184] Teraoka, F., Gogo, K., Mitsuya, K., Shibui, R., and K. Mitani, "Unified Layer 2 (L2) Abstractions for Layer 3 (L3)-Driven Fast Handover", [RFC 5184](#), May 2008.
- [RFC5881] Katz, D. and D. Ward, "Bidirectional Forwarding Detection (BFD) for IPv4 and IPv6 (Single Hop)", [RFC 5881](#), June 2010.

[RFC6206] Levis, P., Clausen, T., Hui, J., Gnawali, O., and J. Ko,
"The Trickle Algorithm", [RFC 6206](#), March 2011.

Goyal, et al.

Expires April 15, 2013

[Page 9]

Internet-Draft [draft-goyal-roll-dis-modifications-01](#)

October 2012

Authors' Addresses

Mukul Goyal
University of Wisconsin Milwaukee
3200 N Cramer St
Milwaukee, WI 53201
USA

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

Dominique Barthel
France Telecom Orange
28 Chemin Du Vieux Chene, BP 98
Meylan, 38243
France

Email: dominique.barthel@orange-ftgroup.com

Emmanuel Baccelli
INRIA

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

