                            DIS Modifications
                   draft-zhong-roll-dis-modifications-00

Abstract

   This document augments [RFC6550] with DIS flags and options that
   allow a RPL node to better control how neighbor RPL routers respond
   to its solicitation for DIOs.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

   This document augments [RFC6550], the RPL routing protocol
   specification.

1.1. RFC6550 refresher

   Per [RFC6550], a RPL node can send a DODAG Information Solicitation
   (DIS) message to solicit DODAG Information Object (DIO) messages from
   neighbor RPL routers.

   A DIS can be multicast to all the routers in range or it can be
   unicast to a specific neighbor router.

   A DIS may carry a Solicited Information option that specifies the
   predicates of the DAG(s) the soliciting node is interested in. In
   the absence of such Solicited Information option, the soliciting node
   is deemed interested in receiving DIOs for all the DAGs known by the
   solicited router(s).

[RFC6550] requires a router to treat the receipt of a multicast DIS as an inconsistency and hence reset its Trickle timers for the matching DAGs. As a result of the general Trickle timer mechanism, future DIOs will be sent at a higher rate. See [RFC6206] for the specification of Trickle timers and the definition of "inconsistency".

[RFC6550] requires a router that receives a unicast DIS to respond by unicasting a DIO for each matching DAG and to not reset the associated Trickle timer. Such a DIO generated in response to a unicast DIS must contain a Configuration option.

This description is summarized in Table 1.

	Unicast DIS	Multicast DIS
no option present	unicast DIO, don't reset Trickle timer	do reset Trickle timer
Solicited Information option present, not matching	do nothing	do nothing
Solicited Information option present, matching	unicast DIO, don't reset Trickle timer	do reset Trickle timer

Table 1: Router behavior on receiving a DIS, as per RFC6550

More precisely, Table 1 describes the behavior of routers for each DAG they belong to. In the general case where multiple RPL instances co-exist in a network, routers will maintain a Trickle timer for the one DAG of each RPL instance they belong to, and nodes may send a DIS with multiple Solicited Information options pertaining to different DAGs or instances. In this more general case, routers will respond for each individual DAG/instance they belong to as per Table 1.

1.2. Undesirable effects

Now, consider a 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 will keep transmitting frequent DIOs for a considerable duration until the Trickle timers again reach long intervals. 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 a RPL router that desires to free up memory by deleting state for the defunct DAGs it belongs to. Identifying a defunct DAG may require the node to solicit DIOs from its DAG parents using a multicast DIS.

1.3. Desired improvments

To deal with the situations described above, there is a need in the industry for DIS flags and options that allow a RPL node to control how neighbor RPL routers respond to its solicitation for DIOs, for example by expressing:

o the routing constraints that routers should meet to be allowed to respond, thereby lowering the number of responders

o whether the responding routers should reset their Trickle timers or not, thereby limiting the cumulated number of transmitted DIOs

o whether the responding routers should respond with a unicast DIO instead of a multicast one, thereby lowering the overhearing cost in the network

o the time interval over which the responding routers should schedule their DIO transmissions, thereby lowering the occurence of collisions.

2. Terminology

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

Additionally, this document uses terminology from [RFC6550]. Specifically, the term RPL node refers to a RPL router or a RPL host as defined in [RFC6550].

3. DIS Base Object flags

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

   o the "No Inconsistency" (N) flag: On receiving a multicast DIS with
      the N flag set, a RPL router MUST NOT reset the Trickle timers for
      the matching DAGs. In addition, it MUST take specific action,
      which is to respond by explicitly sending a DIO. This DIO MUST
      include a Configuration option. This behavior augments [RFC6550],
      which had provision for such flag. Since this specific, one-shot
      DIO is not a consequence of the general Trickle timer mechanism,
      it will be sent right away if no Response Spreading option is
      present or it will be scheduled according to the Response
      Spreading option if one is present in the DIS (see Section 4.2).

   o the "DIO Type" (T) flag: In case the N flag is set, this T flag
      specifies what type of DIO is sent in response. It MUST be a
      unicast DIO if this flag is set and it MUST be a multicast DIO if
      this flag is reset.

   When a unicast DIS is transmitted, both its N and T flags SHOULD be
   0, which are the default values per [RFC6550]. On receiving a
   unicast DIS, the N and T flags MUST be ignored and treated as 00.

   The modified DIS base object is shown in Figure 1.

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Flags    |N|T|    Reserved    | Option(s)...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```
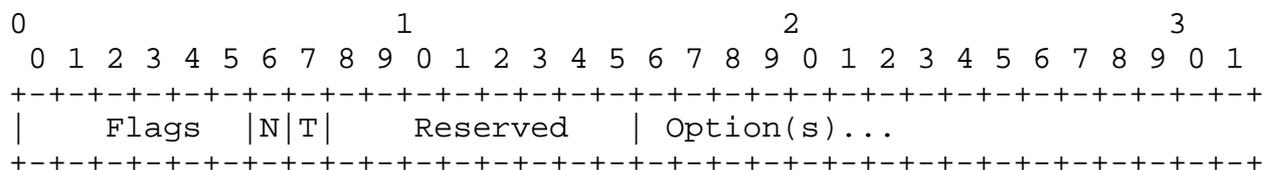
                   Figure 1: Modified DIS Base Object

4. DIS Options

4.1. Metric Container

   In order to lower the number of routers that will respond to a DIS,
   this document allows routing constraints to be carried by a DIS.
   Only the router(s) that satisfy these constraints is (are) allowed to
   respond to the DIS.

   These routing constraints are described using a Metric Container
   option contained in the DIS. Metric Containers are defined in
   [RFC6550] and [RFC6551]. Metric Containers options were previously

only allowed in DIOs. This document augments [RFC6550] by allowing
the inclusion of a Metric Container option inside a DIS as well.

A 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
value of the corresponding routing metric that the router maintains
for the matching DAG(s). These routing metric values MUST satisfy
all the mandatory constraints in order for the router to consider the
solicitation successful for the matching DAG(s). This augments the
behavior already present in [RFC6550] with the Solicited Information
option.

This option can be used in both unicast and multicast DIS.

4.2. Response Spreading

```
 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

Even with the use of the Solicited Information and the Section 4.1
options, a multicast DIS may still lead to a large number of RPL
routers taking immediate action and responding with DIOs. Concurrent
transmissions by multiple routers are not desirable since they may
lead to poor channel utilization or even to packet loss. Unicast
DIOs may be able to avail of link-level retransmissions. However,
multicast DIOs usually have no such protection, since they commonly
make use of link layer broadcast. 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 option, shown in Figure 2, with a recommended Type
value 0x0A (to be confirmed by IANA). A RPL router that explicitely
responds with a specific, one-shot DIO to a DIS that includes a
Response Spreading option, MUST wait for a time uniformly chosen in
the interval [O..2^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.

A Response Spreading option MAY be included inside a unicast DIS
message, but there is no benefit in doing so.

Multiple Response Spreading options SHOULD NOT be used inside a same DIS message.

This mechanism MUST NOT affect the Trickle timer mechanism.

5. Full behavior illustration

Figure 3 and Figure 4 illustrate the normative behavior described in Section 3 and Section 4.1.

```
                        +--------------------+--------------------
                        |   Unicast DIS      |      Multicast DIS
                        +--------------------+--------------------+-
                        |                    |        N=0         |
     +------------------+--------------------+--------------------+-
     |                  | unicast DIO,       |                    |
     | no option present| don't              | do                 |
     |                  | reset Trickle timer| reset Trickle timer|
     +------------------+--------------------+--------------------+-
     | Solicited Informa-|                   |                    |
     | tion/Metric Contai-| do nothing       | do nothing         |
     | ner option present,|                  |                    |
     | not matching.     |                   |                    |
     +------------------+--------------------+--------------------+-
     | Solicited Informa- | unicast DIO,     |                    |
     | tion/Metric Contai-| don't            | do                 |
     | ner option present,| reset Trickle timer| reset Trickle timer|
     | matching.         |                   |                    |
     +------------------+--------------------+--------------------+-
```

Figure 3: Overall DIS behavior, part 1

Notice that Figure 3 is indeed identical to Table 1 when Metric Container options are not used in DIS.

```
         -----------------------------------------+
         | Multicast DIS                          |
         --------------------+--------------------+
         |     N=1, T=0       |     N=1, T=1       |
        -+--------------------+--------------------+
         | multicast DIO,     | unicast DIO,       |
         | don't              | don't              |
         | reset Trickle timer| reset Trickle timer|
        -+--------------------+--------------------+
         |                    |                    |
         | do nothing         | do nothing         |
         |                    |                    |
         |                    |                    |
        -+--------------------+--------------------+
         | multicast DIO,     | unicast DIO,       |
         | don't              | don't              |
         | reset Trickle timer| reset Trickle timer|
         |                    |                    |
        -+--------------------+--------------------+
```

                Figure 4: Overall DIS behavior, part 2

   For the sake of completeness, let's remind here that a specific, one-
   shot DIO generated in response to a DIS must contain a Configuration
   option and that its transmission is delayed according to the Delay
   Spreading option of the DIS, if one such option is present.

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. Acknowledgements

   A lot of text in this document originates from now-expired [I-
   D.goyal-roll-dis-modifications] co-authored with M. Goyal. The
   requirements and solutions also draw from now-expired [I-D.dejean-
   roll-selective-dis] co-authored with N. Dejean. Their contribution
   is deeply acknowledged.

   We also thank (TBA) for their useful feedback and discussion.

9. References

9.1. Normative References

   [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate
              Requirement Levels", BCP 14, RFC 2119,
              DOI 10.17487/RFC2119, March 1997,
              <http://www.rfc-editor.org/info/rfc2119>.

   [RFC6550] Winter, T., Ed., Thubert, P., Ed., 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,
              DOI 10.17487/RFC6550, March 2012,
              <http://www.rfc-editor.org/info/rfc6550>.

   [RFC6551] Vasseur, JP., Ed., Kim, M., Ed., Pister, K., Dejean, N.,
              and D. Barthel, "Routing Metrics Used for Path Calculation
              in Low-Power and Lossy Networks", RFC 6551,
              DOI 10.17487/RFC6551, March 2012,
              <http://www.rfc-editor.org/info/rfc6551>.

9.2. Informative References

   [RFC4861] Narten, T., Nordmark, E., Simpson, W., and H. Soliman,
              "Neighbor Discovery for IP version 6 (IPv6)", RFC 4861,
              DOI 10.17487/RFC4861, September 2007,
              <http://www.rfc-editor.org/info/rfc4861>.

   [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,
              DOI 10.17487/RFC5184, May 2008,
              <http://www.rfc-editor.org/info/rfc5184>.

   [RFC5881] Katz, D. and D. Ward, "Bidirectional Forwarding Detection
              (BFD) for IPv4 and IPv6 (Single Hop)", RFC 5881,
              DOI 10.17487/RFC5881, June 2010,
              <http://www.rfc-editor.org/info/rfc5881>.

   [RFC6206] Levis, P., Clausen, T., Hui, J., Gnawali, O., and J. Ko,
              "The Trickle Algorithm", RFC 6206, DOI 10.17487/RFC6206,
              March 2011, <http://www.rfc-editor.org/info/rfc6206>.

Appendix A. 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 in an energy efficient manner.
   The second mechanism details how a node may identify defunct DAGs for
   which it still maintains state.

A.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, thereby keeping the
   aggregated number of transmissions low. It also has the "DIO Type"
   flag set to make responding routers send unicast DIOs back, thereby
   not triggering full reception in nearby nodes that have state-of-the-
   art radio receivers with hardware-based address filtering.

   The DIS message can include a Response Spreading option prescribing a
   suitable spreading interval based on the expected density of nearby
   routers and on the expected Layer 2 technology.

The DIS will likely include a Metric Container listing the routing
constraints that the responding routers must satisfy in order to be
allowed to respond.

At each iteration, the node multicasts such a DIS and waits for
forthcoming DIOs. After a time equal to the spreading interval, the
node considers the current iteration to be unsuccessful. The node
consequently relaxes the routing constraints somewhat and proceeds to
the next iteration.

The cycle repeats until the node receives one or more DIOs or until
it has relaxed the constraints to the lowest acceptable values.

This algorithm has been proven in the field to be extremely energy-
efficient, especially when routers have a wide communication range.

A.2. Identifying A Defunct DAG

A 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 Unreachanility 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 required the node to
    increase its rank beyond L + 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, a 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, a 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 here 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 a 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:

   * the "No Inconsistency" flag set so that the responding routers do not reset their Trickle timers.

   * the "DIO Type" flag not 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 +
    DAGMaxRankIncrease.)

Appendix B. Experimental data

   The effectiveness of these flags and options has been measured on
   real industrial hardware.

   Data to be added

Authors' Addresses

   Zhong Denglin (editor)
   INSA Lyon
   20 Avenue Albert Einstein
   Villeurbanne 69621
   France

   Email: denglin.zhong@insa-lyon.fr

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

   Email: dominique.barthel@orange.com

Emmanuel Baccelli
INRIA

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