

ROLL  
Internet-Draft  
Intended status: Standards Track  
Expires: 5 November 2026

C. Gundogan  
HAW Hamburg  
E. Baccelli  
INRIA  
G. Z. Papadopoulos, Ed.  
IMT Atlantique  
4 May 2026

RPL DIS Modifications and Applications  
draft-ietf-roll-dis-modifications-02

## Abstract

This document augments [RFC6550] by defining new DODAG Information Solicitation (DIS) flags and options that enable a RPL node to exert finer control over how neighboring RPL routers respond to its DIO solicitations. In addition, this document describes several use cases that motivate these DIS extensions and illustrate scenarios in which enhanced control of DIO responses improves network efficiency, responsiveness, and robustness.

## 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 <https://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 5 November 2026.

## Copyright Notice

Copyright (c) 2026 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 (<https://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 Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

## Table of Contents

1. Introduction . . . . .	2
1.1. RFC 6550 refresher . . . . .	2
1.2. Undesirable effects . . . . .	4
1.3. Desired improvements . . . . .	4
2. Terminology . . . . .	5
3. DIS Base Object flags . . . . .	5
4. DIS Options . . . . .	6
4.1. Metric Container . . . . .	6
4.2. Response Spreading . . . . .	7
4.3. DIO Option Request . . . . .	7
5. Full behavior illustration . . . . .	8
6. Applications . . . . .	10
6.1. A Leaf Node Joining a DAG . . . . .	10
6.2. Identifying A Defunct DAG . . . . .	11
6.3. Adjacencies probing with RPL . . . . .	13
6.3.1. Deliberations . . . . .	14
7. IANA Considerations . . . . .	14
7.1. DIS Flags . . . . .	14
7.2. RPL Control Message Options . . . . .	14
8. Security Considerations . . . . .	14
9. Acknowledgements . . . . .	15
10. References . . . . .	15
10.1. Normative References . . . . .	15
10.2. Informative References . . . . .	15
Appendix A. Implementation Status . . . . .	16
Authors' Addresses . . . . .	16

## 1. Introduction

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

### 1.1. RFC 6550 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, do not reset Trickle timer	Do reset Trickle timer
Solicited Information option present, not matching	Do nothing	Do nothing
Solicited Information option present, matching	Unicast DIO, do not 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

As presented in [Sourailidis2020], there are number of undesirable effects linked to the operation of the DIS control message.

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.

Certain scenarios may require a RPL router to solicit a DIO from a parent by using a unicast DIS. The parent is forced to include a Configuration option within the unicast DIO, although the requesting node might still have this information locally available. Since the information within the Configuration option is described as generally static and unchanging throughout the DODAG, it inflates the unicast DIO unnecessarily by 16 bytes for each request.

## 1.3. Desired improvements

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:

- \* the routing constraints that routers should meet to be allowed to respond, thereby lowering the number of responders.

- \* whether the responding routers should reset their Trickle timers or not, thereby limiting the cumulated number of transmitted DIOs.
- \* whether the responding routers should respond with a unicast DIO instead of a multicast one, thereby lowering the overhearing cost in the network.
- \* whether the responding routers should omit DIO options that were not requested explicitly and thus reducing the amount of traffic and giving full control over the options of the solicited DIO.
- \* the time interval over which the responding routers should schedule their DIO transmissions, thereby lowering the occurrence of collisions.

These results have been attained by the modification implemented and presented in [Sourailidis2020].

## 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].

## 3. DIS Base Object flags

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

- \* 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).
- \* 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.
- \* the "DIO Option Request" (R) flag: On receiving a DIS with the R flag set, the receiver MUST include all options that were requested by the DIS containing one or multiple DIO Option Request options. A responding RPL router MUST NOT include DIO options

that were not explicitly requested. Note that this behaviour contradicts with [RFC6550] for the case of including a Configuration option in all DIOs requested by a unicast DIS.

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. When the R flag is unset, then a RPL router may include or omit DIO options like specified in [RFC6550]. A RPL router responding to a DIS with the R flag set MUST only include all requested DIO options in the solicited DIO.

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
+-----+-----+-----+-----+-----+-----+-----+-----+
|N|T|R|  Flags  |   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

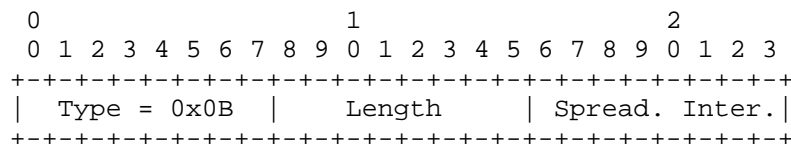


Figure 2: The Response Spreading option

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 0x0B (to be confirmed by IANA). A RPL router that explicitly 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  $[0..2^{\text{SpreadingInterval}}]$ , 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.

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.

## 4.3. DIO Option Request

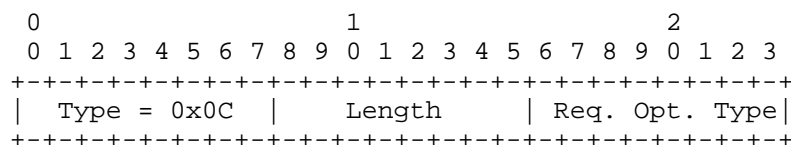


Figure 3: The DIO Option Request option

If a unicast DIS is used to request a DIO, then [RFC6550] mandates that a Configuration option MUST be included in this DIO. The Configuration option contains generally static information that stays unmodified throughout the DAG. For scenarios where a RPL node is already part of a DAG and hence is holding the information that is propagated with the Configuration option, an inclusion of such leads to an unnecessary inflation of 16 bytes for each solicited DIO.

As per [RFC6550], no process is defined to trigger the inclusion of other DIO options in a solicited DIO.

This document defines a new RPL control message option called DIO Option Request option, shown in Figure 3, with a recommended Type value of 0x0C (to be confirmed by IANA). This new option allows full control over the options of the solicited DIO. The target of a unicast or multicast DIS with the R flag set and with one or more DIO Option Request options included, MUST include these requested options in the solicited DIO. For a DIS with the R flag unset, a RPL router behaves like described in [RFC6550] with regard to DIO options.

## 5. Full behavior illustration

Table 2 and Figure 5 illustrate the normative behavior described in Section 3 and Section 4.1.



	Unicast DIS	Multicast DIS	Multicast DIS	Multicast DIS
		N=0	N=1, T=0	N=1, T=1
No option present	Unicast DIO,  do not reset Trickle timer	Do reset Trickle timer	Multicast DIO,  do not reset Trickle timer	Unicast DIO,  do not reset Trickle timer
Solicited Information/ Metric Container  option present, not matching	Do nothing	Do nothing	Do nothing	Do nothing
Solicited Information/ Metric Container  option present, matching	Unicast DIO,  do not reset Trickle timer	Do reset Trickle timer	Multicast DIO,  do not reset Trickle timer	Unicast DIO,  do not reset Trickle timer

Table 2: Router behavior on receiving a DIS, as per [RFC6550]

Notice that Table 2 is indeed identical to Table 1 when Metric Container options are not used in DIS.

For the sake of completeness, let's remind here that a specific, one-shot DIO generated in response to a DIS with the R flag unset MUST contain a Configuration option. If the R flag is set, then this DIO contains only explicitly requested DIO options. This DIO's transmission is delayed according to the Delay Spreading option of the DIS, if one such option is present.

## 6. Applications

This section details some use cases that require DIS modifications compared to the behaviour currently defined in [RFC6550]. The first use case is that of a new leaf node joining an established DAG in an energy efficient manner. The second use case describes why node might want to use DIS to identify defunct DAGs for which it still maintains state. The third use case describes the need for adjacency probing and how DIS can be used for that.

### 6.1. A Leaf Node Joining a DAG

This use case is typically of a smart meter being replaced in the field, while a RPL network is operating and stable. The new smart meter must join the network quickly, without draining the energy of the surrounding nodes, be they battery-operated RPL routers or leaf nodes. In this use case, the issues with the current RPL specification are

- \* Just waiting for a gratuitous DIO may take a long time if the Trickle timers have relaxed to the steady state. A technician who has just installed the new meter needs to positively assess that the meter has joined the network before it leaves the premise. It is not economically viable to ask the technician to standby the meter until a gratuitous DIO has arrived, which may take hours.
- \* If the meter sends a DIS, it needs to do so using multicast, because it has no knowledge of its surroundings. Sending a multicast DIS is considered an inconsistency by the nearby RPL routers. They will reset their Trickle timer to the shortest period. This will trigger sending a stream of DIOs until the Trickle timers relax again. The DIOs will be sent in multicast, which will trigger energy expenditure at nearby nodes, which had no need for the DIOs.

A proposed solution could be the following. 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 [RFC6551].

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.

## 6.2. Identifying A Defunct DAG

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

- \* 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
- \* The neighbor advertises an infinite rank in the DAG; or
- \* Keeping the neighbor as a parent would required the node to increase its rank beyond  $L + \text{DAGMaxRankIncrease}$ , where  $L$  is the minimum rank the node has had in this DAG; or
- \* 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:

- \* The node may fail to receive the neighbor's DIOs advertising an increased rank or the neighbor's membership in a different DAG;

- \* 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:

- \* 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.

- \* 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  $L + \text{DAGMaxRankIncrease}$ .)

### 6.3. Adjacencies probing with RPL

RPL avoids periodic hello messaging as compared to other distance vector protocols. It uses trickle timer based mechanism to update configuration parameters. This significantly reduces the RPL control overhead. One of the fallout of this design choice is that, in the absence of regular traffic, the adjacencies could not be tested and repaired if broken.

RPL provides a mechanism in the form of unicast DIS to query a particular node for its DIO. A node receiving a unicast DIS MUST respond with a unicast DIO with Configuration Option. This mechanism could as well be made use of for probing adjacencies and certain implementations such as Contiki uses this. The periodicity of the probing is implementation dependent, but the node is expected to invoke probing only when

- \* There is no data traffic based on which the links could be tested.
- \* There is no L2 feedback. In some case, L2 might provide periodic beacons at link layer and the absence of beacons could be used for link tests.

### 6.3.1. Deliberations

- \* Should the probing scheme be standardized?
- \* In some cases using multicast based probing may prove advantageous. Currently RPL does not have multicast based probing. Multicast DIS/DIO may not be suitable for probing because it could possibly lead to change of states.

## 7. IANA Considerations

### 7.1. DIS Flags

IANA is requested to allocate bits 0, 1 and 2 of the DIS Flag Field to become the "No Inconsistency", "DIO Type", and "DIO Option Request" bits, the functionality of which is described in Section 3 of this document.

Value	Meaning	Reference
0	No Inconsistency	This document
1	DIO Type	This document
2	DIO Option Request	This document

Figure 4: DIS Flag Field

### 7.2. 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 and the "DIO Option Request" option, the behavior of which are described in Section 4.2 and Section 4.3, respectively.

Value	Meaning	Reference
0x0B	Response Spreading	This document
0x0C	DIO Option Request	This document

Figure 5: RPL Control Message Options

## 8. Security Considerations

TBA

## 9. 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.

## 10. References

### 10.1. 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, <<https://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, <<https://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, <<https://www.rfc-editor.org/info/rfc6551>>.

### 10.2. 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, <<https://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, <<https://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, <<https://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, <<https://www.rfc-editor.org/info/rfc6206>>.
- [Sourailidis2020] Sourailidis, D., Koutsiamanis, R., Papadopoulos, G. Z., Barthel, D., and N. Montavont, "RFC 6550: On Minimizing the Control Plane Traffic of RPL-based Industrial Networks", Proceedings of the 21st IEEE International Symposium on "A World of Wireless, Mobile and Multimedia Networks" (WoWMoM), Cork, Ireland, 2020, <<https://doi.org/10.1109/WoWMoM49955.2020.00080>>.

## Appendix A. Implementation Status

TBA

## Authors' Addresses

Cenk Gundogan  
HAW Hamburg  
Email: [cenk.guendogan@haw-hamburg.de](mailto:cenk.guendogan@haw-hamburg.de)

Emmanuel Baccelli  
INRIA  
Email: [Emmanuel.Baccelli@inria.fr](mailto:Emmanuel.Baccelli@inria.fr)  
URI: <https://www.emmanuelbaccelli.org/>

Georgios Z. Papadopoulos (editor)  
IMT Atlantique  
2 rue de la Chataigneraie  
CS 17607  
35576 Cesson-Sevigne Cedex  
France  
Email: [georgios.papadopoulos@imt-atlantique.fr](mailto:georgios.papadopoulos@imt-atlantique.fr)