ROLL   R. Koutsiamanis, Ed.
Internet-Draft   G. Papadopoulos
Intended status: Standards Track   N. Montavont
Expires: September 10, 2020   IMT Atlantique
                                                         P. Thubert
                                                              Cisco
                                                      March 9, 2020

        Common Ancestor Objective Function and Parent Set DAG Metric Container
                                Extension
                     draft-ietf-roll-nsa-extension-07

Abstract

   Packet Replication and Elimination is a method in which several
   copies of a data packet are sent in the network in order to achieve
   high reliability and low jitter. This document details how to apply
   Packet Replication and Elimination in RPL, especially how to exchange
   information within RPL control packets to let a node better select
   the different parents that will be used to forward the multiple
   copies of a packet. This document also describes the Objective
   Function which takes advantage of this information to implement
   multi-path routing.

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 September 10, 2020.

Copyright Notice

Table of Contents

1. Introduction

   Networks in the industrial context must provide stringent guarantees
   in terms of reliability and predictability, with this domain being
   one of main ones addressed by Deterministic Networking [RFC8557].
   Packet Replication and Elimination (PRE)
   [I-D.papadopoulos-6tisch-pre-reqs] is a technique which allows
   redundant paths in the network to be utilized for traffic requiring
   higher reliability. Allowing industrial applications to function
   over wireless networks requires the application of the principles and
   architecture of Deterministic Networking [RFC8655]. This results in

designs which aim at optimizing packet delivery rate and bounding
latency. Additionally, nodes operating on battery need to minimize
their energy consumption.

As an example, to meet this goal, IEEE Std. 802.15.4 [IEEE802154]
provides Time-Slotted Channel Hopping (TSCH), a mode of operation
which uses a common communication schedule based on timeslots to
allow deterministic medium access as well as channel hopping to work
around radio interference. However, since TSCH uses retransmissions
in the event of a failed transmission, end-to-end latency and jitter
performance can deteriorate.

Furthermore, the 6TiSCH working group, focusing on IPv6 over IEEE
Std. 802.15.4-TSCH, has worked on these issues and produced the
"6TiSCH Architecture" [I-D.ietf-6tisch-architecture] to address that
case. Building on this architecture, "Exploiting Packet Replication
and Elimination in Complex Tracks in 6TiSCH LLNs"
[I-D.papadopoulos-6tisch-pre-reqs] leverages PRE to improve the
Packet Delivery Ratio (PDR), to provide a hard bound to the end-to-
end latency, and thus to limit jitter.

PRE is a general method of maximizing packet delivery rate and
potentially minimizing latency and jitter, not limited to 6TiSCH.
More specifically, PRE achieves controlled redundancy by laying
multiple forwarding paths through the network and using them in
parallel for different copies of a same packet. PRE can follow the
Destination-Oriented Directed Acyclic Graph (DODAG) formed by RPL
from a node to the root. Building a multi-path DODAG can be achieved
based on the RPL capability of having multiple parents for each node
in a network, a subset of which is used to forward packets. In order
to select parents to be part of this subset, the RPL Objective
Function (OF) needs additional information regarding the multi-path
nature of PRE. This document describes an OF which implements multi-
path routing for PRE and specifies the transmission of this specific
path information.

This document describes a new Objective Function (OF) called the
Common Ancestor (CA) OF. A detailed description is given of how the
path information is used within the CA OF and how the subset of
parents for forwarding packets is selected. This specification

defines a new Objective Code Point (OCP) for the CA OF.

For the path information, this specification focuses on the
extensions to the DAG Metric Container [RFC6551] required for
providing the PRE mechanism a part of the information it needs to
operate. This information is the RPL [RFC6550] parent address set of
a node and it must be sent to potential children of the node. The
RPL DIO Control Message is the canonical way of broadcasting this
kind of information and therefore its DAG Metric Container [RFC6551]
field is used to append a Node State and Attribute (NSA) object. The
node's parent address set is stored as an optional TLV within the NSA

object. This specification defines the type value and structure for
the parent address set TLV.

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].

The draft uses the following Terminology:

Packet Replication and Elimination (PRE): A method which consists of
   transmitting multiple copies of a packet using multi-path
   forwarding over a multi-hop network and which consolidates
   multiple received packet copies to control flooding. See
   "Exploiting Packet Replication and Elimination in Complex Tracks
   in 6TiSCH LLNs" [I-D.papadopoulos-6tisch-pre-reqs] for more
   details.

Parent Set (PS): Given a RPL node, the set of its neighbor nodes
   which participate in the same RPL DODAG and which can potentially
   take the role of the node's preferred parent.

Alternative Parent (AP): A RPL parent in the parent set of a node
   which is used to forward a packet copy when replicating packets.

Alternative Parent (AP) Selection: The mechanism for choosing the
   next hop node to forward a packet copy when replicating packets.

Preferred Grand Parent (PGP): The preferred parent of the preferred
          parent of a node.

3. Common Ancestor AP Selection Policies

    In the RPL protocol, each node maintains a list of potential parents.
    For PRE, the Preferred Parent (PP) node is defined to be the same as
    the RPL DODAG Preferred Parent node. Furthermore, to construct an
    alternative path toward the root, in addition to the PP node, each
    node in the network selects additional parent(s), called alternative
    parent(s), from its Parent Set (PS).

    There are multiple possible policies of selecting the AP node. This
    section details three such possible policies.

    All three policies defined perform AP selection based on common
    ancestors, named Common Ancestor Strict, Common Ancestor Medium, and
    Common Ancestor Relaxed, depending on how restrictive the selection
    process is. A more restrictive policy will limit flooding but might

    fail to select an appropriate AP, while a less restrictive one will
    more often find an appropriate AP but might increase flooding.

    All three policies apply their corresponding common ancestor
    criterion to filter the list of candidate neighbours in the
    alternative parent set.

3.1. Common Ancestor Strict

    In the CA Strict OF the node will check if its Preferred Grand Parent
    (PGP), the PP of its PP, is the same as the PP of the potential AP.

                (R) root
                  .   PS(S) = {A, B, C, D}
                  .   PP(S) = C
                  .   PP(PP(S)) = Y
                  .
                                       PS(A) = {W, X}
    (W)   (X)   (Y)   (Z)   PP(A) = X
     ^ ^   ^^ ^ ^   ^^^^ ^   ^ ^^
     | \ // | \ // || \ / || PS(B) = {W, X, Y}

```
    |   //    |   //   ||   /    ||          PP(B) = Y
    |  // \   |  // \  ||  / \   ||
    | //   \  | //   \ || /   \  ||          PS(C) = {X, Y, Z}
  ( A )     ( B )     ( C )     ( D )        PP(C) = Y
    ^         ^        ^^        ^
     \         \       ||    /                PS(D) = {Y, Z}
      \         \      ||   /                 PP(D) = Z
       \         \     ||  /
        \----\\  || /                         || Preferred Parent
              ( S   ) source                  |  Potential Alternative Parent
```

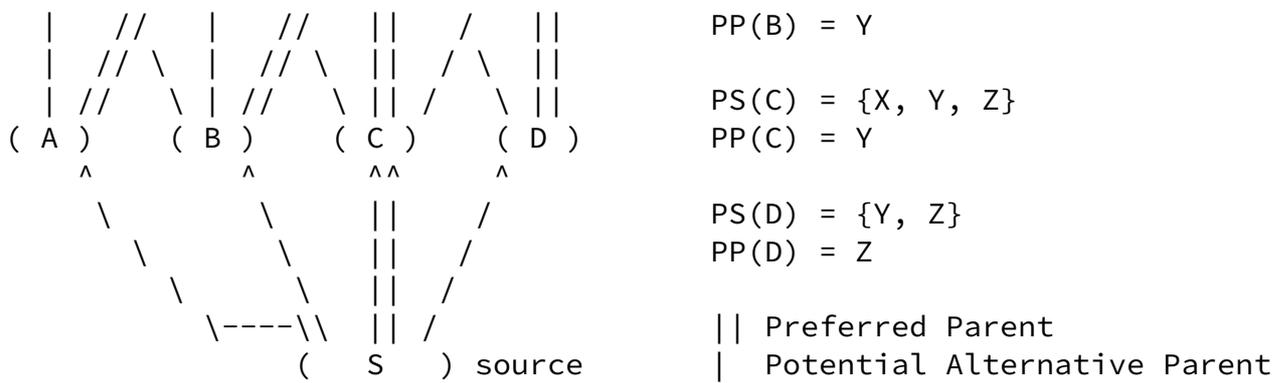   Figure 1: Example Common Ancestor Strict Alternative Parent Selection
                                        policy

   For example, in Figure 1, the source node S must know its grandparent
   sets through nodes A, B, C, and D. The Parent Sets (PS) and the
   Preferred Parents (PS) of nodes A, B, C, and D are shown on the side
   of the figure. The CA Strict parent selection policy will select an
   AP for node S for which PP(PP(S)) = PP(AP). Given that PP(PP(S)) =
   Y:

   o Node A: PP(A) = X and therefore it is different than PP(PP(S))

   o Node B: PS(B) = Y and therefore it is equal to PP(PP(S))

   o Node D: PS(D) = Z and therefore it is different than PP(PP(S))

   node S can decide to use node B as its AP node, since PP(PP(S)) = Y =
   PP(B).

3.2. Common Ancestor Medium

   In the CA Medium OF the node will check if its Preferred Grand Parent
   (PGP), the PP of its PP, is contained in the PS of the potential AP.

   Using the same example, in Figure 1, the CA Medium parent selection
   policy will select an AP for node S for which PP(PP(S)) is in PS(AP).
   Given that PP(PP(S)) = Y:

   o Node A: PS(A) = {W, X} and therefore PP(PP(S)) is not in the set

   o Node B: PS(B) = {W, X, Y} and therefore PP(PP(S)) is in the set

o Node D: PS(D) = {Y, Z} and therefore PP(PP(S)) is in the set

   node S can decide to use node B or D as its AP node.

3.3. Common Ancestor Relaxed

   In the CA Relaxed OF the node will check if the Parent Set (PS) of
   its Preferred Parent (PP) has a node in common with the PS of the
   potential AP.

   Using the same example, in Figure 1, the CA Relaxed parent selection
   policy will select an AP for node S for which PS(PP(S)) has at least
   one node in common with PS(AP). Given that PS(PP(S)) = {X, Y, Z}:

   o Node A: PS(A) = {W, X} and the common nodes are {X}

   o Node B: PS(B) = {W, X, Y} and the common nodes are {X, Y}

   o Node D: PS(D) = {Y, Z} and the common nodes are {Y, Z}

   node S can decide to use node A, B or D as its AP node.

4. Common Ancestor Objective Function

   An OF which allows the multiple paths to remain correlated is
   detailed here. More specifically, when using this OF a node will
   select an AP node close to its PP node to allow the operation of
   overhearing between parents. For more details about overhearing and
   its use in this context see Section 4.3. "Promiscuous Overhearing"
   in "Exploiting Packet Replication and Elimination in Complex Tracks
   in 6TiSCH LLNs" [I-D.papadopoulos-6tisch-pre-reqs]. If multiple

   potential APs match this condition, the AP with the lowest rank will
   be registered.

   The OF described here is an extension of the The Minimum Rank with
   Hysteresis Objective Function [RFC6719] (MRHOF). In general, this OF
   extends MRHOF by specifying how an AP is selected. Importantly, the
   calculation of the rank of the node through each candidate neighbor
   and the selection of the PP is kept the same as in MRHOF.

The ways in which the CA OF modifies MRHOF in a section-by-section manner follows in detail:

MRHOF [Section 3](). "The Minimum Rank with Hysteresis Objective Function":
   Same as MRHOF extended to AP selection. Minimum Rank path selection and switching applies correspondingly to the AP with the extra CA requirement of having some match between ancestors.

MRHOF [Section 3.1](). "Computing the Path Cost": Same as MRHOF extended to AP selection. If a candidate neighbor does not fulfill the CA requirement then the path through that neighbor SHOULD be set to MAX_PATH_COST, the same value used by MRHOF. As a result, the node MUST NOT select the candidate neighbor as its AP.

MRHOF [Section 3.2](). "Parent Selection": Same as MRHOF extended to AP selection. To allow hysteresis, AP selection maintains a variable, cur_ap_min_path_cost, which is the path cost of the current AP.

MRHOF [Section 3.2.1](). "When Parent Selection Runs": Same as MRHOF.

MRHOF [Section 3.2.2](). "Parent Selection Algorithm": Same as MRHOF extended to AP selection. If the smallest path cost for paths through the candidate neighbors is smaller than cur_ap_min_path_cost by less than PARENT_SWITCH_THRESHOLD (the same variable as MRHOF uses), the node MAY continue to use the current AP. Additionally, if there is no PP selected, there MUST NOT be any AP selected as well. Finally, as with MRHOF, a node MAY include up to PARENT_SET_SIZE-1 additional candidate neighbors in its alternative parent set. The value of PARENT_SET_SIZE is the same as in MRHOF.

MRHOF [Section 3.3](). "Computing Rank": Same as MRHOF.

MRHOF [Section 3.4](). "Advertising the Path Cost": Same as MRHOF.

MRHOF [Section 3.5](). "Working without Metric Containers":

It is not possible to work without metric containers, since CA AP

selection requires information from parents regarding their parent
sets, which is transmitted via the NSA object in the DIO Mectric
Container.

MRHOF [Section 4](#). "Using MRHOF for Metric Maximization":
   Same as MRHOF.

MRHOF [Section 5](#). "MRHOF Variables and Parameters": Same as MRHOF
   extended to AP selection. The CA OF operates like MRHOF for AP
   selection by maintaining separate:

   AP: Corresponding to the MRHOF PP. Hysteresis is configured for
      AP with the same PARENT_SWITCH_THRESHOLD parameter as in MRHOF.
      The AP MUST NOT be the same as the PP.

   Alternative parent set: Corresponding to the MRHOF parent set.
      The size is defined by the same PARENT_SET_SIZE parameter as in
      MRHOF. The Alternative parent set MUST be a non-strict subset
      of the parent set.

   cur_ap_min_path_cost: Corresponding to the MRHOF
      cur_min_path_cost variable. To support the operation of the
      hysteresis function for AP selection.

MRHOF [Section 6](#). "Manageability": Same as MRHOF.

MRHOF [Section 6.1](#). "Device Configuration": Same as MRHOF.

MRHOF [Section 6.2](#). "Device Monitoring": Same as MRHOF.

[4.1](#). Usage

All OF policies apply their corresponding criterion to filter the
list of candidate neighbours in the alternative parent set. The AP
is then selected from the alternative parent set based on Rank and
using hysteresis as is done for the PP in MRHOF. It is noteworthy
that the OF uses the same Objective Code Point (OCP): TBD1 for all
policies used.

The PS information can be used by any of the described AP selection
policies or other ones not described here, depending on requirements.
It is optional for all nodes to use the same AP selection policies.
Different nodes may use different AP selection policies, since the
selection policy is local to each node. For example, using different
policies can be used to vary the transmission reliability in each
hop.

5. Node State and Attribute (NSA) object type extension

   In order to select their AP node, nodes need to be aware of their
   grandparent node sets. Within RPL [RFC6550], the nodes use the DODAG
   Information Object (DIO) Control Message to broadcast information
   about themselves to potential children. However, RPL [RFC6550], does
   not define how to propagate parent set related information, which is
   what this document addresses.

   DIO messages can carry multiple options, out of which the DAG Metric
   Container option [RFC6551] is the most suitable structurally and
   semantically for the purpose of carrying the parent set. The DAG
   Metric Container option itself can carry different nested objects,
   out of which the Node State and Attribute (NSA) [RFC6551] is
   appropriate for transferring generic node state data. Within the
   Node State and Attribute it is possible to store optional TLVs
   representing various node characteristics. As per the Node State and
   Attribute (NSA) [RFC6551] description, no TLV has been defined for
   use. This document defines one TLV for the purpose of transmitting a
   node's parent set.

```
    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
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   | RPLInstanceID |Version Number |              Rank             |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |G|0| MOP | Prf |     DTSN      |     Flags     |    Reserved   |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                                                               |
   +                                                               +
   |                                                               |
   +                            DODAGID                            +
   |                                                               |
   +                                                               +
   |                                                               |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   | DAGMC Type (2)| DAGMC Length  |                               |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+                               |
   |                                                               |
   //                 DAG Metric Container data                  //
   |                                                               |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```
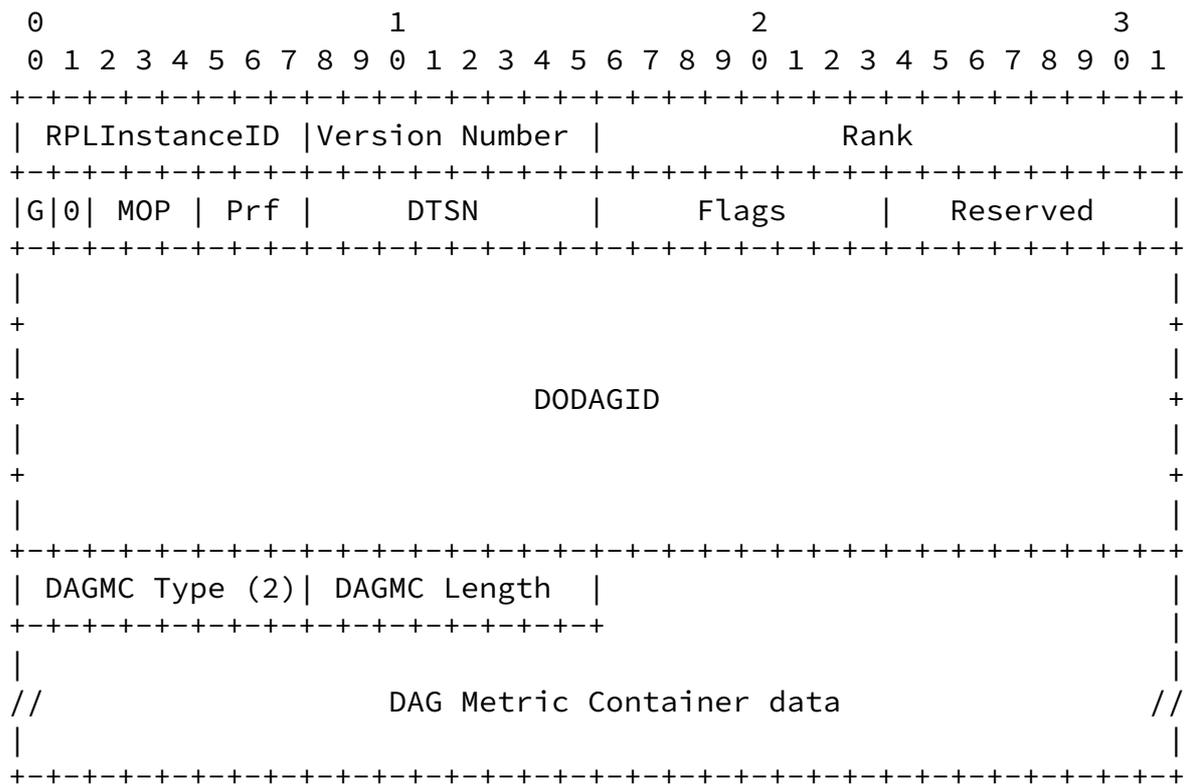
      Figure 2: Example DIO Message with a DAG Metric Container option

Figure 2 shows the structure of the DIO Control Message when a DAG
Metric Container option is included. The DAG Metric Container option

---

type (DAGMC Type in Figure 2) has the value 0x02 as per the IANA
registry for the RPL Control Message Options, and is defined in
[RFC6550]. The DAG Metric Container option length (DAGMC Length in
Figure 2) expresses the DAG Metric Container length in bytes. DAG
Metric Container data holds the actual data and is shown expanded in
Figure 3.

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|Routing-MC-Type|Res Flags|P|C|O|R| A   | Prec | Length (bytes)| |MC
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    Res        | Flags   |A|O|   PS type   |    PS  Length | |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ |NSA
|   PS IPv6 address(es) ...                                    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 3: DAG Metric Container (MC) data with Node State and
Attribute (NSA) object body and a TLV

The structure of the DAG Metric Container data in the form of a Node
State and Attribute (NSA) object with a TLV in the NSA Optional TLVs
field is shown in Figure 3. The first 32 bits comprise the DAG
Metric Container header and all the following bits are part of the
Node State and Attribute object body, as defined in [RFC6551]. This
document defines a new TLV, which CAN be carried in the Node State
and Attribute (NSA) object Optional TLVs field. The TLV is named
Parent Set and is abbreviated as PS in Figure 3.

PS type: The type of the Parent Set TLV. The value is TBD2.

PS Length: The total length of the TLV value field (PS IPv6
     address(es)) in bytes. The length is an integral multiple of
     16, the number of bytes in an IPv6 address.

PS IPv6 address(es) One or more 128-bit IPv6 address(es) without any
     separator between them. The field consists of one IPv6 address
     per parent in the parent set. The parent addresses are listed

in decreasing order of preference and not all parents in the
parent set need to be included. The selection of how many
parents from the parent set are to be included is left to the
implementation. The number of parent addresses in the PS IPv6
address(es) field can be deduced by dividing the length of the
PS IPv6 address(es) field in bytes by 16, the number of bytes
in an IPv6 address.

5.1. Usage

The PS SHOULD be used in the process of parent selection, and
especially in AP selection, since it can help the alternative path to
not significantly deviate from the preferred path. The Parent Set is
information local to the node that broadcasts it.

The PS is used only within NSA objects configured as a metric,
therefore the DAG Metric Container field "C" MUST be 0.
Additionally, since the information in the PS needs to be propagated
downstream but it cannot be aggregated, the DAG Metric Container
field "R" MUST be 1. Finally, since the information contained is by
definition partial, more specifically just the parent set of the DIO-
sending node, the DAG Metric Container field "P" MUST be 1.

It is important that the PS does not affect the calculation of the
rank through candidate neighbors. It is only used with the CA OF to
remove nodes which do not fulfill the CA OF criteria from the
candidate neighbor list.

6. Controlling PRE

PRE is very helpful when the aim is to increase reliability for a
certain path, however its use creates additional traffic as part of
the replication process. It is conceivable that not all paths have
stringent reliability requirements. Therefore, a way to control
whether PRE is applied to a path's packets SHOULD be implemented.
For example, a traffic class label can be used to determine this
behavior per flow type as described in Deterministic Networking
Architecture [RFC8655].

7. Security Considerations

The structure of the DIO control message is extended, within the pre-
defined DIO options. The additional information are the IPv6
addresses of the parent set of the node transmitting the DIO. This
use of this additional information can have the following potential
consequences:

o A malicious node that can receive and read the DIO can "see"
   further than it's own neighbourhood by one hop, learning the
   addresses of it's two hop neighbors. This is a privacy / network
   discovery issue.

o A malicious node that can send DIOs can use the parent set
   extension to convince neighbours to route through itself, instead
   of the normal preferred parent they would use. However, this is
   already possible with other OFs (like OF0 [RFC6552] and MRHOF

   [RFC6719]) by reporting a fake rank value in the DIO, thus
   masquerading as the DODAG root.

8. IANA Considerations

   This proposal requests the allocation of a new value TBD1 from the
   "Objective Code Point (OCP)" sub-registry of the "Routing Protocol
   for Low Power and Lossy Networks (RPL)" registry.

   This proposal also requests the allocation of a new value TBD2 for
   the "Parent Set" TLV from the Routing Metric/Constraint TLVs sub-
   registry from IANA.

9. References

9.1. Informative references

   [I-D.ietf-6tisch-architecture]
             Thubert, P., "An Architecture for IPv6 over the TSCH mode
             of IEEE 802.15.4", draft-ietf-6tisch-architecture-28 (work
             in progress), October 2019.

   [I-D.papadopoulos-6tisch-pre-reqs]
             Papadopoulos, G., Montavont, N., and P. Thubert,
             "Exploiting Packet Replication and Elimination in Complex

             Tracks in 6TiSCH LLNs", draft-papadopoulos-6tisch-pre-
             reqs-02 (work in progress), July 2018.

   [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>.

   [RFC6552] Thubert, P., Ed., "Objective Function Zero for the Routing
              Protocol for Low-Power and Lossy Networks (RPL)",
              RFC 6552, DOI 10.17487/RFC6552, March 2012,
              <https://www.rfc-editor.org/info/rfc6552>.

   [RFC6719] Gnawali, O. and P. Levis, "The Minimum Rank with
              Hysteresis Objective Function", RFC 6719,
              DOI 10.17487/RFC6719, September 2012,
              <https://www.rfc-editor.org/info/rfc6719>.

   [RFC8557] Finn, N. and P. Thubert, "Deterministic Networking Problem
              Statement", RFC 8557, DOI 10.17487/RFC8557, May 2019,
              <https://www.rfc-editor.org/info/rfc8557>.

   [RFC8655] Finn, N., Thubert, P., Varga, B., and J. Farkas,
              "Deterministic Networking Architecture", RFC 8655,
              DOI 10.17487/RFC8655, October 2019,
              <https://www.rfc-editor.org/info/rfc8655>.

9.2. Other Informative References

   [IEEE802154]
              IEEE standard for Information Technology, "IEEE Std
              802.15.4 Standard for Low-Rate Wireless Personal Area
              Networks (WPANs)", December 2015.

9.3. URIs

   [1] https://github.com/ariskou/contiki/tree/draft-koutsiamanis-roll-
        nsa-extension

   [2] https://code.wireshark.org/review/gitweb?p=wireshark.git;a=commit
        ;h=e2f6ba229f45d8ccae2a6405e0ef41f1e61da138

Appendix A. Implementation Status

   A research-stage implementation of the PRE mechanism using the
   proposed extension as part of a 6TiSCH IOT use case was developed at
   IMT Atlantique, France by Tomas Lagos Jenschke and Remous-Aris
   Koutsiamanis. It was implemented on the open-source Contiki OS and
   tested with the Cooja simulator. The DIO DAGMC NSA extension is
   implemented with a configurable number of parents from the parent set
   of a node to be reported.

                      (R)

    (11)   (12)   (13)   (14)   (15)   (16)

    (21)   (22)   (23)   (24)   (25)   (26)

    (31)   (32)   (33)   (34)   (35)   (36)

```
     (41)   (42)   (43)   (44)   (45)   (46)


     (51)   (52)   (53)   (54)   (55)   (56)


                    ( S )
```

                   Figure 4: Simulation Topology

   The simulation setup is:

   Topology: 32 nodes structured in regular grid as show in Figure 4.
      Node S (source) is the only data packet sender, and send data to
      node R (root). The parent set of each node (except R) is all the
      nodes in the immediately higher row, the immediately above 6
      nodes. For example, each node in {51, 52, 53, 54, 55, 56} is
      connected to all of {41, 42, 43, 44, 45, 46}. Node 11, 12, 13,
      14, 15, 16 have a single upwards link to R.

   MAC: TSCH with 1 retransmission

   Platform: Cooja

   Schedule: Static, 2 timeslots per link from each node to each parent
      in its parent set, 1 broadcast EB slot, 1 sender-based shared
      timeslot (for DIO and DIS) per node (total of 32).

   Simulation lifecycle: Allow link formation for 100 seconds before
      starting to send data packets. Afterwards, S sends data packets
      to R. The simulation terminates when 1000 packets have been sent
      by S.

   Radio Links: Every 60 s, a new Packet Delivery Rate is randomly
      drawn for each link, with a uniform distribution spanning the 70%
      to 100% interval.

   Traffic Pattern: CBR, S sends one non-fragmented UDP packet every 5
      seconds to R.

   PS extension size: 3 parents.

Routing Methods:

   * RPL: The default RPL non-PRE implementation in Contiki OS.

   * 2nd ETX: PRE with a parent selection method which picks as AP
      the 2nd best parent in the parent set based on ETX.

   * CA Strict: As described in Section 3.1.

   * CA Medium: As described in Section 3.2.

                        Simulation results:

+-----------+--------------+----------------+---------------------+
Routing	Average	Average	Average
Method	Packet	Traversed	Duplications/packet
	Delivery Rate	Nodes/packet	(#)
	(%)	(#)	
+-----------+--------------+----------------+---------------------+
RPL	82.70	5.56	7.02
2nd ETX	99.38	14.43	31.29
CA Strict	97.32	9.86	18.23
CA Medium	99.66	13.75	28.86
+-----------+--------------+----------------+---------------------+

   Links:

   o Contiki OS DIO DAGMC NSA extension (draft-koutsiamanis-roll-nsa-
      extension branch) [1]

   o Wireshark dissectors (for the optional PS TLV) - currently merged
      / in master [2]

Appendix B. Choosing an AP selection policy

   The manner of choosing an AP selection policy is left to the
   implementation, for maximum flexibility.

   For example, a different policy can be used per traffic type. The
   network configurator can choose the CA Relaxed policy to increase
   reliability (thus producing some flooding) for specific, extremely
   important, alert packets. On the other hand, all normal data traffic

uses the CA Strict policy. Therefore, an exception is made just for
the alert packets.

Another option would be to devise a new disjoint policy, where the
paths are on purpose non-correlated, to increase path diversity and
resilience against whole groups of nodes failing. The disadvantage
may be increased jitter.

Finally, a network configurator may provide the CA policies with a
preference order of Strict > Medium > Relaxed as a means of falling
back to more flood-prone policies to maintain reliability.

Authors' Addresses

   Remous-Aris Koutsiamanis (editor)
   IMT Atlantique
   Office B00 - 126A
   2 Rue de la Chataigneraie
   Cesson-Sevigne - Rennes 35510
   FRANCE

   Phone: +33 299 12 70 49
   Email: aris@ariskou.com

   Georgios Papadopoulos
   IMT Atlantique
   Office B00 - 114A
   2 Rue de la Chataigneraie
   Cesson-Sevigne - Rennes 35510
   FRANCE

   Phone: +33 299 12 70 04
   Email: georgios.papadopoulos@imt-atlantique.fr

   Nicolas Montavont
   IMT Atlantique
   Office B00 - 106A
   2 Rue de la Chataigneraie
   Cesson-Sevigne - Rennes 35510
   FRANCE

   Phone: +33 299 12 70 23
   Email: nicolas.montavont@imt-atlantique.fr

Pascal Thubert
Cisco Systems, Inc
Building D
45 Allee des Ormes - BP1200
MOUGINS - Sophia Antipolis 06254
FRANCE

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