                         Efficient Route Invalidation
                      draft-ietf-roll-efficient-npdao-17

Abstract

   This document explains the problems associated with the current use
   of NPDAO messaging and also discusses the requirements for an
   optimized route invalidation messaging scheme. Further a new
   proactive route invalidation message called as "Destination Cleanup
   Object" (DCO) is specified which fulfills requirements of an
   optimized route invalidation messaging.

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 May 1, 2020.

Copyright Notice

Table of Contents

1. Introduction

RPL [RFC6550] (Routing Protocol for Low power and lossy networks)
specifies a proactive distance-vector based routing scheme. RPL has
optional messaging in the form of DAO (Destination Advertisement
Object) messages, which the 6LBR (6Lo Border Router) and 6LR (6Lo
Router) can use to learn a route towards the downstream nodes. In
storing mode, DAO messages would result in routing entries being
created on all intermediate 6LRs from the node's parent all the way
towards the 6LBR.

RPL allows the use of No-Path DAO (NPDAO) messaging to invalidate a
routing path corresponding to the given target, thus releasing
resources utilized on that path. A NPDAO is a DAO message with route
lifetime of zero, originates at the target node and always flows
upstream towards the 6LBR. This document explains the problems
associated with the current use of NPDAO messaging and also discusses
the requirements for an optimized route invalidation messaging
scheme. Further a new proactive route invalidation message called as
"Destination Cleanup Object" (DCO) is specified which fulfills
requirements of an optimized route invalidation messaging.

The document only caters to the RPL's storing mode of operation
(MOP). The non-storing MOP does not require use of NPDAO for route
invalidation since routing entries are not maintained on 6LRs.

1.1. Requirements Language and 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 [BCP 14](RFC2119) [[RFC2119]] [[RFC8174]] when, and only when, they appear in all capitals, as shown here.

This specification requires readers to be familiar with all the terms and concepts that are discussed in "RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks" [[RFC6550]].

Low Power and Lossy Networks (LLN):
   Network in which both the routers and their interconnect are constrained. LLN routers typically operate with constraints on processing power, memory, and energy (batter power). Their

   interconnects are characterized by high loss rates, low data rates, and instability.
6LoWPAN Router (6LR):
   An intermediate router that is able to send and receive Router Advertisements (RAs) and Router Solicitations (RSs) as well as forward and route IPv6 packets.
Directed Acyclic Graph (DAG):
   A directed graph having the property that all edges are oriented in such a way that no cycles exist.
Destination-Oriented DAG (DODAG):
   A DAG rooted at a single destination, i.e., at a single DAG root with no outgoing edges.
6LoWPAN Border Router (6LBR):
   A border router which is a DODAG root and is the edge node for traffic flowing in and out of the 6LoWPAN network.
Destination Advertisement Object (DAO):
   DAO messaging allows downstream routes to the nodes to be established.
DODAG Information Object (DIO):
   DIO messaging allows upstream routes to the 6LBR to be established. DIO messaging is initiated at the DAO root.
Common Ancestor node
   6LR/6LBR node which is the first common node between two paths of a target node.
No-Path DAO (NPDAO):
   A DAO message which has target with lifetime 0 used for the purpose of route invalidation.
Destination Cleanup Object (DCO):
   A new RPL control message code defined by this document. DCO

messaging improves proactive route invalidation in RPL.
Regular DAO:
   A DAO message with non-zero lifetime. Routing adjacencies are
   created or updated based on this message.
Target node:
   The node switching its parent whose routing adjacencies are
   updated (created/removed).

1.2. Current NPDAO messaging

RPL uses NPDAO messaging in the storing mode so that the node
changing its routing adjacencies can invalidate the previous route.
This is needed so that nodes along the previous path can release any
resources (such as the routing entry) they maintain on behalf of
target node.

For the rest of this document consider the following topology:

```
                           (6LBR)
                             |
                             |
                             |
                            (A)
                            / \
                           /   \
                          /     \
                        (G)     (H)
                         |       |
                         |       |
                         |       |
                        (B)     (C)
                          \       ;
                           \      ;
                            \   ;
                            (D)
                            / \
                           /   \
                          /     \
                        (E)     (F)
```
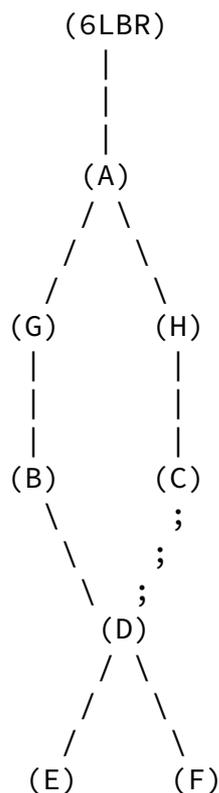
Node (D) is connected via preferred parent (B). (D) has an alternate
path via (C) towards the 6LBR. Node (A) is the common ancestor for
(D) for paths through (B)-(G) and (C)-(H). When (D) switches from
(B) to (C), RPL allows sending NPDAO to (B) and regular DAO to (C).

1.3. Why Is NPDAO Important?

Nodes in LLNs may be resource constrained. There is limited memory
available and routing entry records are one of the primary elements
occupying dynamic memory in the nodes. Route invalidation helps 6LR
nodes to decide which entries could be discarded to better optimize
resource utilization. Thus it becomes necessary to have an efficient
route invalidation mechanism. Also note that a single parent switch
may result in a "sub-tree" switching from one parent to another.
Thus the route invalidation needs to be done on behalf of the sub-
tree and not the switching node alone. In the above example, when
Node (D) switches parent, the route updates needs to be done for the
routing tables entries of (C),(H),(A),(G), and (B) with destination
(D),(E) and (F). Without efficient route invalidation, a 6LR may
have to hold a lot of stale route entries.

2. Problems with current NPDAO messaging

2.1. Lost NPDAO due to link break to the previous parent

When a node switches its parent, the NPDAO is to be sent to its
previous parent and a regular DAO to its new parent. In cases where
the node switches its parent because of transient or permanent parent
link/node failure then the NPDAO message is bound to fail.

2.2. Invalidate Routes of Dependent Nodes

RPL does not specify how route invalidation will work for dependent
nodes rooted at the switching node, resulting in stale routing
entries of the dependent nodes. The only way for 6LR to invalidate
the route entries for dependent nodes would be to use route lifetime

expiry which could be substantially high for LLNs.

In the example topology, when Node (D) switches its parent, Node (D)
generates an NPDAO on its behalf. There is no NPDAO generated by the
dependent child nodes (E) and (F), through the previous path via (D)
to (B) and (G), resulting in stale entries on nodes (B) and (G) for
nodes (E) and (F).

2.3. Possible route downtime caused by asynchronous operation of NPDAO
      and DAO

A switching node may generate both an NPDAO and DAO via two different
paths at almost the same time. There is a possibility that an NPDAO
generated may invalidate the previous route and the regular DAO sent
via the new path gets lost on the way. This may result in route
downtime impacting downward traffic for the switching node.

In the example topology, consider Node (D) switches from parent (B)
to (C). An NPDAO sent via the previous route may invalidate the
previous route whereas there is no way to determine whether the new
DAO has successfully updated the route entries on the new path.

3. Requirements for the NPDAO Optimization

3.1. Req#1: Remove messaging dependency on link to the previous parent

When the switching node sends the NPDAO message to the previous
parent, it is normal that the link to the previous parent is prone to
failure (that's why the node decided to switch). Therefore, it is
required that the route invalidation does not depend on the previous
link which is prone to failure. The previous link referred here
represents the link between the node and its previous parent (from
whom the node is now disassociating).

3.2. Req#2: Dependent nodes route invalidation on parent switching

It should be possible to do route invalidation for dependent nodes
rooted at the switching node.

3.3. Req#3: Route invalidation should not impact data traffic

While sending the NPDAO and DAO messages, it is possible that the

NPDAO successfully invalidates the previous path, while the newly
sent DAO gets lost (new path not set up successfully). This will
result in downstream unreachability to the node switching paths.
Therefore, it is desirable that the route invalidation is
synchronized with the DAO to avoid the risk of route downtime.

4. Changes to RPL signaling

4.1. Change in RPL route invalidation semantics

As described in Section 1.2, the NPDAO originates at the node
changing to a new parent and traverses upstream towards the root. In
order to solve the problems as mentioned in Section 2, the document
adds a new proactive route invalidation message called "Destination
Cleanup Object" (DCO) that originates at a common ancestor node and
flows downstream between the new and old path. The common ancestor
node generates a DCO in response to the change in the next-hop on
receiving a regular DAO with updated Path Sequence for the target.

The 6LRs in the path for DCO take action such as route invalidation
based on the DCO information and subsequently send another DCO with
the same information downstream to the next hop. This operation is
similar to how the DAOs are handled on intermediate 6LRs in storing
MOP in [RFC6550]. Just like DAO in storing MOP, the DCO is sent
using link-local unicast source and destination IPv6 address. Unlike
DAO, which always travels upstream, the DCO always travels
downstream.

In Figure 1, when node D decides to switch the path from B to C, it
sends a regular DAO to node C with reachability information
containing the address of D as the target and an incremented Path
Sequence. Node C will update the routing table based on the
reachability information in the DAO and in turn generate another DAO
with the same reachability information and forward it to H. Node H
also follows the same procedure as Node C and forwards it to node A.
When node A receives the regular DAO, it finds that it already has a
routing table entry on behalf of the target address of node D. It
finds however that the next hop information for reaching node D has
changed i.e., node D has decided to change the paths. In this case,
Node A which is the common ancestor node for node D along the two

paths (previous and new), should generate a DCO which traverses

downwards in the network. Node A handles normal DAO forwarding to
6LBR as required by [RFC6550].

4.2. Transit Information Option changes

   Every RPL message is divided into base message fields and additional
   Options as described in Section 6 of [RFC6550]. The base fields
   apply to the message as a whole and options are appended to add
   message/use-case specific attributes. As an example, a DAO message
   may be attributed by one or more "RPL Target" options which specify
   the reachability information for the given targets. Similarly, a
   Transit Information option may be associated with a set of RPL Target
   options.

   This document specifies a change in the Transit Information Option to
   contain the "Invalidate previous route" (I) flag. This 'I' flag
   signals the common ancestor node to generate a DCO on behalf of the
   target node with a RPL Status of 130 indicating that the address has
   moved. The 'I' flag is carried in the Transit Information Option
   which augments the reachability information for a given set of RPL
   Target(s). Transit Information Option with 'I' flag set should be
   carried in the DAO message when route invalidation is sought for the
   corresponding target(s).

```
     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
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |   Type = 0x06 | Option Length |E|I|  Flags    | Path Control |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    | Path Sequence | Path Lifetime |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```
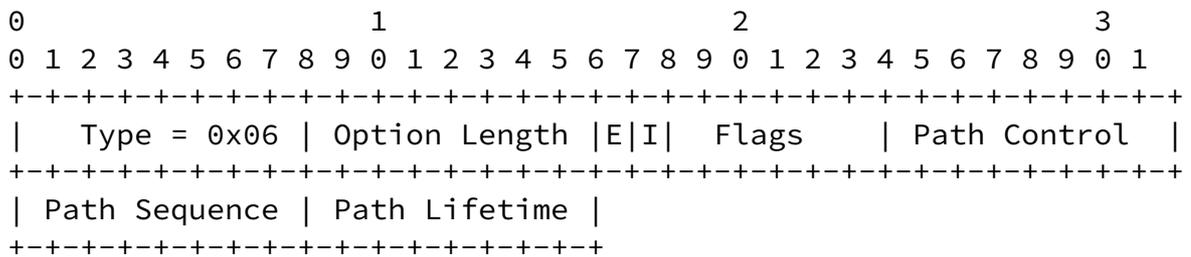
       Figure 2: Updated Transit Information Option (New I flag added)

   I (Invalidate previous route) flag: The 'I' flag is set by the target
   node to indicate to the common ancestor node that it wishes to
   invalidate any previous route between the two paths.

   [RFC6550] allows the parent address to be sent in the Transit
   Information Option depending on the mode of operation. In case of
   storing mode of operation the field is usually not needed. In case
   of DCO, the parent address field MUST NOT be included.

   The common ancestor node SHOULD generate a DCO message in response to
   this 'I' flag when it sees that the routing adjacencies have changed
   for the target. The 'I' flag is intended to give the target node

control over its own route invalidation, serving as a signal to
request DCO generation.

4.3. Destination Cleanup Object (DCO)

A new ICMPv6 RPL control message code is defined by this
specification and is referred to as "Destination Cleanup Object"
(DCO), which is used for proactive cleanup of state and routing
information held on behalf of the target node by 6LRs. The DCO
message always traverses downstream and cleans up route information
and other state information associated with the given target.

```
    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 |K|D|   Flags   |  RPL  Status  |  DCOSequence  |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                                                               |
   +                                                               +
   |                                                               |
   +                        DODAGID(optional)                      +
   |                                                               |
   +                                                               +
   |                                                               |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |   Option(s)...
   +-+-+-+-+-+-+-+-+
```
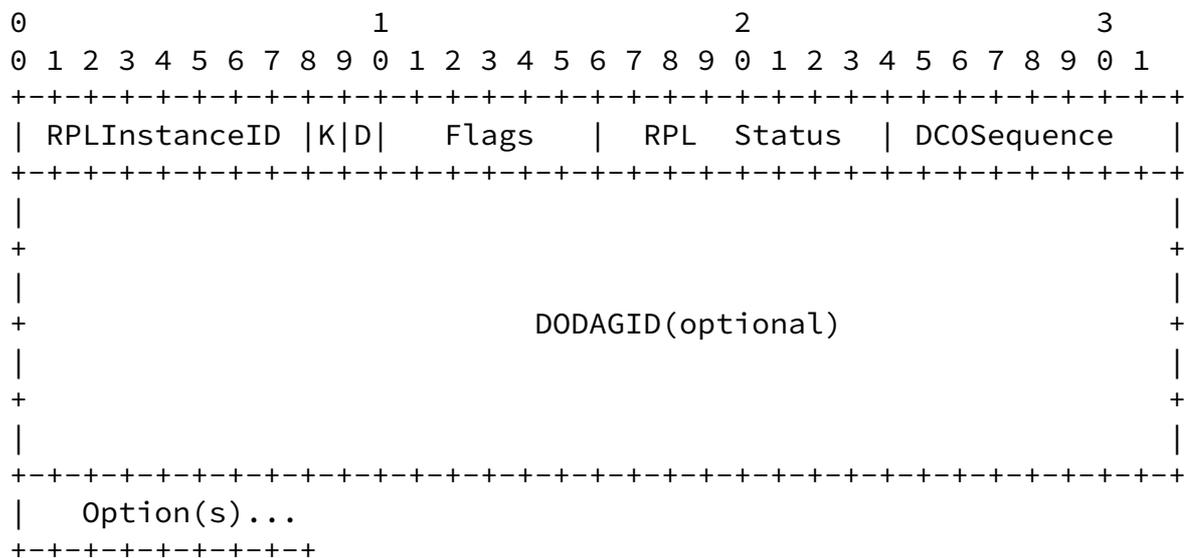
                     Figure 3: DCO base object

RPLInstanceID: 8-bit field indicating the topology instance
associated with the DODAG, as learned from the DIO.

K: The 'K' flag indicates that the recipient of DCO message is
expected to send a DCO-ACK back. If the DCO-ACK is not received even
after setting the 'K' flag, an implementation may retry the DCO at a
later time. The number of retries are implementation and deployment
dependent and are expected to be kept similar with those used in DAO
retries in [RFC6550]. Section 4.6.3 specifies the considerations for
DCO retry. A node receiving a DCO message without the 'K' flag set
MAY respond with a DCO-ACK, especially to report an error condition.
An example error condition could be that the node sending the DCO-ACK
does not find the routing entry for the indicated target. When the
sender does not set the 'K' flag it is an indication that the sender
does not expect a response, and the sender SHOULD NOT retry the DCO.

D: The 'D' flag indicates that the DODAGID field is present. This
flag MUST be set when a local RPLInstanceID is used.

Flags: The 6 bits remaining unused in the Flags field are reserved
for future use. These bits MUST be initialized to zero by the sender
and MUST be ignored by the receiver.

RPL Status: As defined in [RFC6550] and updated in
[I-D.ietf-roll-unaware-leaves]. The root or common parent that
generates a DCO is authoritative for setting the status information
and the information is unchanged as propagated down the DODAG. This
document does not specify a differentiated action based on the RPL
status.

DCOSequence: 8-bit field incremented at each unique DCO message from
a node and echoed in the DCO-ACK message. The initial DCOSequence
can be chosen randomly by the node. Section 4.4 explains the
handling of the DCOSequence.

DODAGID (optional): 128-bit unsigned integer set by a DODAG root that
uniquely identifies a DODAG. This field MUST be present when the 'D'
flag is set and MUST NOT be present if 'D' flag is not set. DODAGID
is used when a local RPLInstanceID is in use, in order to identify
the DODAGID that is associated with the RPLInstanceID.

4.3.1. Secure DCO

A Secure DCO message follows the format in [RFC6550] Figure 7, where
the base message format is the DCO message shown in Figure 3.

4.3.2. DCO Options

The DCO message MUST carry at least one RPL Target and the Transit
Information Option and MAY carry other valid options. This
specification allows for the DCO message to carry the following
options:

```
0x00 Pad1
0x01 PadN
0x05 RPL Target
0x06 Transit Information
0x09 RPL Target Descriptor
```

Section 6.7 of [RFC6550] defines all the above mentioned options.
The DCO carries an RPL Target Option and an associated Transit
Information Option with a lifetime of 0x00000000 to indicate a loss
of reachability to that Target.

4.3.3. Path Sequence number in the DCO

A DCO message may contain a Path Sequence in the Transit Information
Option to identify the freshness of the DCO message. The Path
Sequence in the DCO MUST use the same Path Sequence number present in
the regular DAO message when the DCO is generated in response to a
DAO message. Thus if a DCO is received by a 6LR and subsequently a
DAO is received with an old sequence number, then the DAO MUST be
ignored. When the DCO is generated in response to a DCO from
upstream parent, the Path Sequence MUST be copied from the received
DCO.

4.3.4. Destination Cleanup Option Acknowledgment (DCO-ACK)

The DCO-ACK message SHOULD be sent as a unicast packet by a DCO
recipient in response to a unicast DCO message with 'K' flag set. If
'K' flag is not set then the receiver of the DCO message MAY send a
DCO-ACK, especially to report an error condition.

```
 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 |D|   Flags     |  DCOSequence  | DCO-ACK Status|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
+                                                               +
|                                                               |
+                      DODAGID(optional)                        +
|                                                               |
+                                                               +
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 4: DCO-ACK base object

RPLInstanceID: 8-bit field indicating the topology instance
associated with the DODAG, as learned from the DIO.

D: The 'D' flag indicates that the DODAGID field is present. This
flag MUST be set when a local RPLInstanceID is used.

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

DCOSequence: 8-bit field. The DCOSequence in DCO-ACK is copied from
the DCOSequence received in the DCO message.

DCO-ACK Status: Indicates the completion. A value of 0 is defined as
unqualified acceptance in this specification. A value of 1 is
defined as "No routing-entry for the Target found". The remaining
status values are reserved as rejection codes.

DODAGID (optional): 128-bit unsigned integer set by a DODAG root that
uniquely identifies a DODAG. This field MUST be present when the 'D'
flag is set and MUST NOT be present when 'D' flag is not set.
DODAGID is used when a local RPLInstanceID is in use, in order to
identify the DODAGID that is associated with the RPLInstanceID.

4.3.5. Secure DCO-ACK

A Secure DCO-ACK message follows the format in [RFC6550] Figure 7,
where the base message format is the DCO-ACK message shown in
Figure 4.

4.4. DCO Base Rules

1. If a node sends a DCO message with newer or different information
    than the prior DCO message transmission, it MUST increment the
    DCOSequence field by at least one. A DCO message transmission
    that is identical to the prior DCO message transmission MAY
    increment the DCOSequence field. The DCOSequence counter follows
    the sequence counter operation as defined in Section 7.2 of

[RFC6550].

2. The RPLInstanceID and DODAGID fields of a DCO message MUST be the
    same value as that of the DAO message in response to which the
    DCO is generated on the common ancestor node.
3. A node MAY set the 'K' flag in a unicast DCO message to solicit a
    unicast DCO-ACK in response in order to confirm the attempt.
4. A node receiving a unicast DCO message with the 'K' flag set
    SHOULD respond with a DCO-ACK. A node receiving a DCO message
    without the 'K' flag set MAY respond with a DCO-ACK, especially
    to report an error condition.
5. A node receiving a unicast DCO message MUST verify the stored
    Path Sequence in context to the given target. If the stored Path
    Sequence is more fresh, newer than the Path Sequence received in
    the DCO, then the DCO MUST be dropped.
6. A node that sets the 'K' flag in a unicast DCO message but does
    not receive DCO-ACK in response MAY reschedule the DCO message
    transmission for another attempt, up until an implementation
    specific number of retries.
7. A node receiving a unicast DCO message with its own address in
    the RPL Target Option MUST strip-off that Target Option. If this
    Target Option is the only one in the DCO message then the DCO
    message MUST be dropped.

Jadhav, et al.   Expires May 1, 2020   [Page 12]

Internet-Draft   Efficient Route Invalidation   October 2019

   The scope of DCOSequence values is unique to the node which generates
   it.

4.5. Unsolicited DCO

   A 6LR may generate an unsolicited DCO to unilaterally cleanup the
   path on behalf of the target entry. The 6LR has all the state
   information, namely, the Target address and the Path Sequence,
   required for generating DCO in its routing table. The conditions why
   6LR may generate an unsolicited DCO are beyond the scope of this
   document but some possible reasons could be:

   1. On route expiry of an entry, a 6LR may decide to graciously
       cleanup the entry by initiating DCO.
   2. 6LR needs to entertain higher priority entries in case the
       routing table is full, thus resulting in eviction of an existing
       routing entry. In this case the eviction can be handled
       graciously using DCO.

Note that if the 6LR initiates a unilateral path cleanup using DCO
and if it has the latest state for the target then the DCO would
finally reach the target node. Thus the target node would be
informed of its invalidation.

4.6. Other considerations

4.6.1. Dependent Nodes invalidation

Current RPL [RFC6550] does not provide a mechanism for route
invalidation for dependent nodes. This document allows the dependent
nodes invalidation. Dependent nodes will generate their respective
DAOs to update their paths, and the previous route invalidation for
those nodes should work in the similar manner described for switching
node. The dependent node may set the 'I' flag in the Transit
Information Option as part of regular DAO so as to request
invalidation of previous route from the common ancestor node.

Dependent nodes do not have any indication regarding if any of their
parents in turn have decided to switch their parent. Thus for route
invalidation the dependent nodes may choose to always set the 'I'
flag in all its DAO message's Transit Information Option. Note that
setting the 'I' flag is not counterproductive even if there is no
previous route to be invalidated.

4.6.2. NPDAO and DCO in the same network

The current NPDAO mechanism in [RFC6550] can still be used in the
same network where DCO is used. The NPDAO messaging can be used, for
example, on route lifetime expiry of the target or when the node
simply decides to gracefully terminate the RPL session on graceful
node shutdown. Moreover, a deployment can have a mix of nodes
supporting the DCO and the existing NPDAO mechanism. It is also
possible that the same node supports both the NPDAO and DCO signaling
for route invalidation.

[Section 9.8 of [RFC6550]](#) states, "When a node removes a node from its
DAO parent set, it SHOULD send a No-Path DAO message to that removed
DAO parent to invalidate the existing router". This document
introduces an alternative and more optimized way of route
invalidation but it also allows existing NPDAO messaging to work.
Thus an implementation has two choices to make when a route
invalidation is to be initiated:

1. Use NPDAO to invalidate the previous route and send regular DAO
    on the new path.
2. Send regular DAO on the new path with the 'I' flag set in the
    Transit Information Option such that the common ancestor node
    initiates the DCO message downstream to invalidate the previous
    route.

This document recommends using option 2 for reasons specified in
[Section 3](#) in this document.

This document assumes that all the 6LRs in the network support this
specification. If there are 6LRs en-route DCO message path which do
not support this document, then the route invalidation for
corresponding targets may not work or may work partially i.e., only
part of the path supporting DCO may be invalidated. Alternatively, a
node could generate an NPDAO if it does not receive a DCO with itself
as target within specified time limit. The specified time limit is
deployment specific and depends upon the maximum depth of the network
and per hop average latency. Note that sending NPDAO and DCO for the
same operation would not result in unwanted side-effects because the
acceptability of NPDAO or DCO depends upon the Path Sequence
freshness.

[4.6.3](#). Considerations for DCO retry

A DCO message could be retried by a sender if it sets the 'K' flag
and does not receive a DCO-ACK. The DCO retry time could be
dependent on the maximum depth of the network and average per hop
latency. This could range from 2 seconds to 120 seconds depending on

the deployment. In case the latency limits are not known, an
implementation MUST NOT retry more than once in 3 seconds and MUST
NOT retry more than 3 times.

The number of retries could also be set depending on how critical the
route invalidation could be for the deployment and the link layer
retry configuration. For networks supporting only MP2P and P2MP
flows, such as in AMI and telemetry applications, the 6LRs may not be
very keen to invalidate routes, unless they are highly memory-
constrained. For home and building automation networks which may
have substantial P2P traffic, the 6LRs might be keen to invalidate
efficiently because it may additionally impact the forwarding
efficiency.

4.6.4. DCO with multiple preferred parents

[RFC6550] allows a node to select multiple preferred parents for
route establishment. Section 9.2.1 of [RFC6550] specifies, "All DAOs
generated at the same time for the same Target MUST be sent with the
same Path Sequence in the Transit Information". Subsequently when
route invalidation has to be initiated, RPL mentions use of NPDAO
which can be initiated with an updated Path Sequence to all the
parent nodes through which the route is to be invalidated.

With DCO, the Target node itself does not initiate the route
invalidation and it is left to the common ancestor node. A common
ancestor node when it discovers an updated DAO from a new next-hop,
it initiates a DCO. With multiple preferred parents, this handling
does not change. But in this case it is recommended that an
implementation initiates a DCO after a time period (DelayDCO) such
that the common ancestor node may receive updated DAOs from all
possible next-hops. This will help to reduce DCO control overhead
i.e., the common ancestor can wait for updated DAOs from all possible
directions before initiating a DCO for route invalidation. After
timeout, the DCO needs to be generated for all the next-hops for whom
the route invalidation needs to be done.

This document recommends using a DelayDCO timer value of 1sec. This
value is inspired by the default DelayDAO value of 1sec in [RFC6550].
Here the hypothesis is that the DAOs from all possible parent sets
would be received on the common ancestor within this time period.

It is still possible that a DCO is generated before all the updated
DAOs from all the paths are received. In this case, the ancestor
node would start the invalidation procedure for paths from which the
updated DAO is not received. The DCO generated in this case would
start invalidating the segments along these paths on which the
updated DAOs are not received. But once the DAO reaches these

segments, the routing state would be updated along these segments and
should not lead to any inconsistent routing state.

Note that there is no requirement for synchronization between DCO and
DAOs. The DelayDCO timer simply ensures that the DCO control
overhead can be reduced and is only needed when the network contains
nodes using multiple preferred parent.

5. Acknowledgments

Many thanks to Alvaro Retana, Cenk Gundogan, Simon Duquennoy,
Georgios Papadopoulous, Peter Van Der Stok for their review and
comments. Alvaro Retana helped shape this document's final version
with critical review comments.

6. IANA Considerations

IANA is requested to allocate new codes for the DCO and DCO-ACK
messages from the RPL Control Codes registry.

Code	Description	Reference
TBD1	Destination Cleanup Object	This
		document
TBD2	Destination Cleanup Object Acknowledgment	This
		document
TBD3	Secure Destination Cleanup Object	This
		document
TBD4	Secure Destination Cleanup Object	This
	Acknowledgment	document

IANA is requested to allocate bit 1 from the Transit Information
Option Flags registry for the 'I' flag (Section 4.2)

6.1. New Registry for the Destination Cleanup Object (DCO) Flags

IANA is requested to create a registry for the 8-bit Destination
Cleanup Object (DCO) Flags field. This registry should be located in
existing category of "Routing Protocol for Low Power and Lossy
Networks (RPL)".

New bit numbers may be allocated only by an IETF Review. Each bit is
tracked with the following qualities:

o Bit number (counting from bit 0 as the most significant bit)

o Capability description

   o Defining RFC

   The following bits are currently defined:

```
    +-----------+---------------------------+---------------+
    | Bit number |        Description        |   Reference   |
    +-----------+---------------------------+---------------+
    |     0     |    DCO-ACK request (K)    | This document |
    |     1     | DODAGID field is present (D) | This document |
    +-----------+---------------------------+---------------+
```

                        DCO Base Flags

6.2. New Registry for the Destination Cleanup Object Acknowledgment
      (DCO-ACK) Status field

   IANA is requested to create a registry for the 8-bit Destination
   Cleanup Object Acknowledgment (DCO-ACK) Status field. This registry
   should be located in existing category of "Routing Protocol for Low
   Power and Lossy Networks (RPL)".

   New Status values may be allocated only by an IETF Review. Each
   value is tracked with the following qualities:

   o Status Code
   o Description
   o Defining RFC

   The following values are currently defined:

```
    +------------+--------------------------------------+------------+
    |   Status   |             Description               | Reference  |
    |    Code    |                                       |            |
    +------------+--------------------------------------+------------+
    |     0      |        Unqualified acceptance         |    This    |
    |            |                                       |  document  |
    |     1      |   No routing-entry for the indicated  |    This    |
    |            |             Target found              |  document  |
    +------------+--------------------------------------+------------+
```

DCO-ACK Status Codes

6.3. New Registry for the Destination Cleanup Object (DCO)
      Acknowledgment Flags

   IANA is requested to create a registry for the 8-bit Destination
   Cleanup Object (DCO) Acknowledgment Flags field. This registry

   should be located in existing category of "Routing Protocol for Low
   Power and Lossy Networks (RPL)".

   New bit numbers may be allocated only by an IETF Review. Each bit is
   tracked with the following qualities:

   o Bit number (counting from bit 0 as the most significant bit)
   o Capability description
   o Defining RFC

   The following bits are currently defined:

   +-----------+------------------------------+---------------+
   | Bit number | Description | Reference |
   +-----------+------------------------------+---------------+
   | 0 | DODAGID field is present (D) | This document |
   +-----------+------------------------------+---------------+

                        DCO-ACK Base Flags

7. Security Considerations

   This document introduces the ability for a common ancestor node to
   invalidate a route on behalf of the target node. The common ancestor
   node could be directed to do so by the target node using the 'I' flag
   in DCO's Transit Information Option. However, the common ancestor
   node is in a position to unilaterally initiate the route invalidation
   since it possesses all the required state information, namely, the
   Target address and the corresponding Path Sequence. Thus a rogue
   common ancestor node could initiate such an invalidation and impact
   the traffic to the target node.

   The DCO carries a RPL Status value, which is informative. New Status

values may be created over time and a node will ignore an unknown
Status value. This enables RPL Status field to be used as a cover
channel. But the channel only works once since the message destroys
its own medium, that is the existing route that it is removing.

This document also introduces an 'I' flag which is set by the target
node and used by the ancestor node to initiate a DCO if the ancestor
sees an update in the route adjacency. However, this flag could be
spoofed by a malicious 6LR in the path and can cause invalidation of
an existing active path. Note that invalidation will happen only if
the other conditions such as Path Sequence condition is also met.
Having said that, such a malicious 6LR may spoof a DAO on behalf of
the (sub) child with the 'I' flag set and can cause route
invalidation on behalf of the (sub) child node. Note that, using
existing mechanisms offered by [RFC6550], a malicious 6LR might also

spoof a DAO with lifetime of zero or otherwise cause denial of
service by dropping traffic entirely, so the new mechanism described
in this document does not present a substantially increased risk of
disruption.

This document assumes that the security mechanisms as defined in
[RFC6550] are followed, which means that the common ancestor node and
all the 6LRs are part of the RPL network because they have the
required credentials. A non-secure RPL network needs to take into
consideration the risks highlighted in this section as well as those
highlighted in [RFC6550].

All RPL messages support a secure version of messages which allows
integrity protection using either a MAC or a signature. Optionally,
secured RPL messages also have encryption protection for
confidentiality.

The document adds new messages (DCO, DCO-ACK) which are syntactically
similar to existing RPL messages such as DAO, DAO-ACK. Secure
versions of DCO and DCO-ACK are added similar to other RPL messages
(such as DAO, DAO-ACK).

RPL supports three security modes as mentioned in Section 10.1 of
   [RFC6550]:

1. Unsecured: In this mode, it is expected that the RPL control

messages are secured by other security mechanisms, such as link-
layer security. In this mode, the RPL control messages,
including DCO, DCO-ACK, do not have Security sections. Also note
that unsecured mode does not imply that all messages are sent
without any protection.
   2. Preinstalled: In this mode, RPL uses secure messages. Thus
       secure versions of DCO, DCO-ACK MUST be used in this mode.
   3. Authenticated: In this mode, RPL uses secure messages. Thus
       secure versions of DCO, DCO-ACK MUST be used in this mode.

8. Normative References

   [I-D.ietf-roll-unaware-leaves]
              Thubert, P. and M. Richardson, "Routing for RPL Leaves",
              draft-ietf-roll-unaware-leaves-04 (work in progress),
              September 2019.

   [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>.

   [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC
              2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174,
              May 2017, <https://www.rfc-editor.org/info/rfc8174>.

Appendix A. Example Messaging

A.1. Example DCO Messaging

   In Figure 1, node (D) switches its parent from (B) to (C). This
   example assumes that Node D has already established its own route via
   Node B-G-A-6LBR using pathseq=x. The example uses DAO and DCO
   messaging convention and specifies only the required parameters to

explain the example namely, the parameter 'tgt', which stands for
Target Option and value of this parameter specifies the address of
the target node. The parameter 'pathseq', which specifies the Path
Sequence value carried in the Transit Information Option. The
parameter 'I_flag' specifies the 'I' flag in the Transit Information
Option. sequence of actions is as follows:

1. Node D switches its parent from node B to node C
2. D sends a regular DAO(tgt=D,pathseq=x+1,I_flag=1) in the updated
    path to C
3. C checks for a routing entry on behalf of D, since it cannot find
    an entry on behalf of D it creates a new routing entry and
    forwards the reachability information of the target D to H in a
    DAO(tgt=D,pathseq=x+1,I_flag=1).
4. Similar to C, node H checks for a routing entry on behalf of D,
    cannot find an entry and hence creates a new routing entry and
    forwards the reachability information of the target D to A in a
    DAO(tgt=D,pathseq=x+1,I_flag=1).
5. Node A receives the DAO(tgt=D,pathseq=x+1,I_flag=1), and checks
    for a routing entry on behalf of D. It finds a routing entry but
    checks that the next hop for target D is different (i.e., Node
    G). Node A checks the I_flag and generates
    DCO(tgt=D,pathseq=x+1) to previous next hop for target D which is
    G. Subsequently, Node A updates the routing entry and forwards
    the reachability information of target D upstream
    DAO(tgt=D,pathseq=x+1,I_flag=1).
6. Node G receives the DCO(tgt=D,pathseq=x+1). It checks if the
    received path sequence is later than the stored path sequence.
    If it is later, Node G invalidates the routing entry of target D

    and forwards the (un)reachability information downstream to B in
    DCO(tgt=D,pathseq=x+1).
7. Similarly, B processes the DCO(tgt=D,pathseq=x+1) by invalidating
    the routing entry of target D and forwards the (un)reachability
    information downstream to D.
8. D ignores the DCO(tgt=D,pathseq=x+1) since the target is itself.
9. The propagation of the DCO will stop at any node where the node
    does not have an routing information associated with the target.
    If cached routing information is present and the cached Path
    Sequence is higher than the value in the DCO, then the DCO is
    dropped.

. Example DCO Messaging with multiple preferred parents

```
                          (6LBR)
                            |
                            |
                            |
                          (N11)
                          /  \
                         /    \
                        /      \
                     (N21)    (N22)
                      /        / \
                     /        /   \
                    /        /     \
                 (N31)   (N32)   (N33)
                   :       |     /
                    :      |    /
                     :     |   /
                        (N41)
```
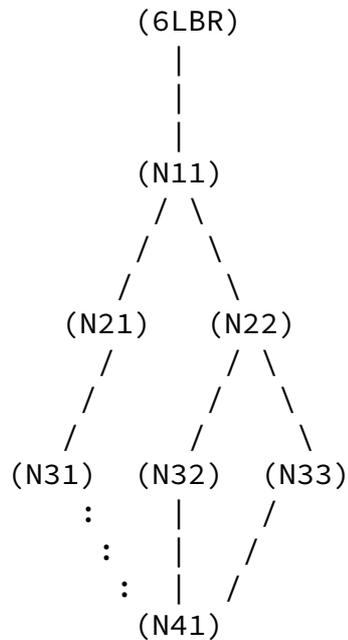
                    Figure 5: Sample topology 2

   In Figure 5, node (N41) selects multiple preferred parents (N32) and
   (N33). The sequence of actions is as follows:

   1.   (N41) sends DAO(tgt=N41,PS=x,I_flag=1) to (N32) and (N33). Here
        I_flag refers to the Invalidation flag and PS refers to Path
        Sequence in Transit Information option.
   2.   (N32) sends DAO(tgt=N41,PS=x,I_flag=1) to (N22). (N33) also
        sends DAO(tgt=N41,PS=x,I_flag=1) to (N22). (N22) learns
        multiple routes for the same destination (N41) through multiple
        next-hops. (N22) may receive the DAOs from (N32) and (N33) in
        any order with the I_flag set. The implementation should use
        the DelayDCO timer to wait to initiate the DCO. If (N22)
        receives an updated DAO from all the paths then the DCO need not

---

        be initiated in this case. Thus the route table at N22 should
        contain (Dst,NextHop,PS): { (N41,N32,x), (N41,N33,x) }.
   3.   (N22) sends DAO(tgt=N41,PS=x,I_flag=1) to (N11).
   4.   (N11) sends DAO(tgt=N41,PS=x,I_flag=1) to (6LBR). Thus the
        complete path is established.

5. (N41) decides to change preferred parent set from { N32, N33 }
    to { N31, N32 }.
6. (N41) sends DAO(tgt=N41,PS=x+1,I_flag=1) to (N32). (N41) sends
    DAO(tgt=N41,PS=x+1,I_flag=1) to (N31).
7. (N32) sends DAO(tgt=N41,PS=x+1,I_flag=1) to (N22). (N22) has
    multiple routes to destination (N41). It sees that a new Path
    Sequence for Target=N41 is received and thus it waits for pre-
    determined time period (DelayDCO time period) to invalidate
    another route {(N41),(N33),x}. After time period, (N22) sends
    DCO(tgt=N41,PS=x+1) to (N33). Also (N22) sends the regular
    DAO(tgt=N41,PS=x+1,I_flag=1) to (N11).
8. (N33) receives DCO(tgt=N41,PS=x+1). The received Path Sequence
    is latest and thus it invalidates the entry associated with
    target (N41). (N33) then sends the DCO(tgt=N41,PS=x+1) to
    (N41). (N41) sees itself as the target and drops the DCO.
9. From Step 6 above, (N31) receives the
    DAO(tgt=N41,PS=x+1,I_flag=1). It creates a routing entry and
    sends the DAO(tgt=N41,PS=x+1,I_flag=1) to (N21). Similarly
    (N21) receives the DAO and subsequently sends the
    DAO(tgt=N41,PS=x+1,I_flag=1) to (N11).
10. (N11) receives DAO(tgt=N41,PS=x+1,I_flag=1) from (N21). It
    waits for DelayDCO timer since it has multiple routes to (N41).
    (N41) will receive DAO(tgt=N41,PS=x+1,I_flag=1) from (N22) from
    Step 7 above. Thus (N11) has received regular
    DAO(tgt=N41,PS=x+1,I_flag=1) from all paths and thus does not
    initiate DCO.
11. (N11) forwards the DAO(tgt=N41,PS=x+1,I_flag=1) to 6LBR and the
    full path is established.

Authors' Addresses

Rahul Arvind Jadhav (editor)
Huawei
Kundalahalli Village, Whitefield,
Bangalore, Karnataka 560037
India

Phone: +91-080-49160700
Email: rahul.ietf@gmail.com

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

Rabi Narayan Sahoo
Huawei
Kundalahalli Village, Whitefield,
Bangalore, Karnataka 560037
India

Phone: +91-080-49160700
Email: rabinarayans@huawei.com

Zhen Cao
Huawei
W Chang'an Ave
Beijing
P.R. China

Email: zhencao.ietf@gmail.com