Stream: Internet Engineering Task Force (IETF)
RFC: 9009
Category: Standards Track
Published: April 2021
ISSN: 2070-1721
Authors: R.A. Jadhav, Ed. *Huawei* P. Thubert *Cisco* R.N. Sahoo *Huawei* Z. Cao *Huawei*

RFC 9009
Efficient Route Invalidation

Abstract

This document explains the problems associated with the use of No-Path Destination Advertisement Object (NPDAO) messaging in RFC 6550 and also discusses the requirements for an optimized route invalidation messaging scheme. Further, this document specifies a new proactive route invalidation message called the "Destination Cleanup Object" (DCO), which fulfills requirements for optimized route invalidation messaging.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

RPL (the Routing Protocol for Low-Power and Lossy Networks) as defined in [RFC6550] specifies a proactive distance-vector-based routing scheme. RPL has optional messaging in the form of DAO (Destination Advertisement Object) messages, which the 6LBR (6LoWPAN Border Router) and 6LR (6LoWPAN Router) can use to learn a route towards the downstream nodes. ("6LoWPAN" stands for "IPv6 over Low-Power Wireless Personal Area Network".) In Storing mode, DAO messages would result in routing entries being created on all intermediate 6LRs from a 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. An NPDAO is a DAO message with a route lifetime of zero. It originates at the target node and always flows upstream towards the 6LBR. This document explains the problems associated with the use of NPDAO messaging in [RFC6550] and also discusses the requirements for an optimized route invalidation messaging scheme. Further, this document specifies a new proactive route invalidation message called the "Destination Cleanup Object" (DCO), which fulfills requirements for optimized route invalidation messaging.

This document only caters to RPL's Storing Mode of Operation (MOP). The Non-Storing MOP does not require the use of an 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] [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 Network (LLN):
: A network in which both the routers and their interconnects are constrained. LLN routers typically operate with constraints on processing power, memory, and energy (battery 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 that is a DODAG root and is the edge node for traffic flowing in and out of the 6LoWPAN.

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:
: A 6LR/6LBR node that is the first common node between two paths of a target node.

No-Path DAO (NPDAO):
: A DAO message that has a target with a lifetime of 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 a 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. RPL NPDAO Messaging

RPL uses NPDAO messaging in 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 the target node.

Throughout this document, we will refer to the topology shown in Figure 1:

```
                  (6LBR)
                    |
                    |
                    |
                   (A)
                   / \
                  /   \
                 /     \
               (G)     (H)
                |       |
                |       |
                |       |
               (B)     (C)
                 \      ;
                  \    ;
                   \  ;
                    (D)
                    / \
                   /   \
                  /     \
                (E)     (F)
```

Figure 1: Sample Topology

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 an NPDAO to B and a regular DAO to C.

1.3. Why Is NPDAO Messaging Important?

Resources in LLN nodes are typically 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 routing entries can be discarded for better use of the limited resources. Thus, it becomes necessary to have an efficient route invalidation mechanism. Also note that a single parent switch may result in a "subtree" switching from one parent to another. Thus, the route invalidation needs to be done on behalf of the subtree and not the switching node alone. In the above example, when Node D switches its parent, route updates need to be done for the routing table entries of C, H, A, G, and B with destinations D, E, and F. Without efficient route invalidation, a 6LR may have to hold a lot of stale route entries.

2. Problems with the RPL 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, the NPDAO message may not be received by the parent.

2.2. Invalidating Routes of Dependent Nodes

RPL does not specify how route invalidation will work for dependent nodes in the switching node subDAG, resulting in stale routing entries of the dependent nodes. The only way for a 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 own 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 the NPDAO and DAO

A switching node may generate both an NPDAO and a DAO via two different paths at almost the same time. It is possible that the NPDAO 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, say that 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 NPDAO Optimization

3.1. Req. #1: Remove Messaging Dependency on the 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 to here represents the link between the node and its previous parent (from which the node is now disassociating).

3.2. Req. #2: Route Invalidation for Dependent Nodes at the Parent Switching Node

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 discussed in Section 2, this document adds a new proactive route invalidation message called the "Destination Cleanup Object" (DCO), which originates at a common ancestor node and flows downstream the old path. The common ancestor node generates a DCO when removing a next hop to a target -- for instance, as a delayed response to receiving a regular DAO from another child node with a Path Sequence for the target that is the same or newer, in which case the DCO transmission is canceled.

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

In Figure 1, when child Node D decides to switch the path from parent B to parent 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 will in turn generate another DAO with the same reachability information and forward it to H. Node H recursively 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), can generate a DCO that traverses the network downwards over the old path to the target. Node A handles normal DAO forwarding to the 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-specific / use-case-specific attributes. As an example, a DAO message may be attributed by one or more "RPL Target" options that specify that the reachability information is 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 195, 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 one or more RPL Targets. A Transit Information option with the 'I' flag set should be carried in the DAO message when route invalidation is sought for the corresponding target or targets.

Value 195 represents the 'U' and 'A' bits in RPL Status, to be set as per Figure 6 of [RFC9010], with the lower 6 bits set to the 6LoWPAN Neighbor Discovery (ND) Extended Address Registration Option (EARO) Status value of 3 indicating 'Moved' as per Table 1 of [RFC8505].

```
 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 MOP. In the case of the Storing MOP, the field is usually not needed. In the case of a DCO, the Parent Address field **MUST NOT** be included.

Upon receiving a DAO message with a Transit Information option that has the 'I' flag set, and as a delayed response removing a routing adjacency to the target indicated in the Transit Information option, the common ancestor node **SHOULD** generate a DCO message to the next hop associated to that adjacency. 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 the "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. The format of the DCO message is shown 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| 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 a 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 is implementation and deployment dependent and is expected to be kept similar to the number of DAO retries [RFC6550]. Section 4.6.3 specifies the considerations for DCO retries. 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 [RFC9010]. 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 the 'D' flag is not set. The 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 shown in [RFC6550], Figure 7, where the base message format is the DCO message shown in Figure 3 of this document.

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 a 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 in the DCO

A DCO message includes a Transit Information option for each invalidated path. The value of the Path Sequence counter in the Transit Information option allows identification of the freshness of the DCO message versus the newest known to the 6LRs along the path being removed. If the DCO is generated by a common parent in response to a DAO message, then the Transit Information option in the DCO **MUST** use the value of the Path Sequence as found in the newest Transit Information option that was received for that target by the common parent. If a 6LR down the path receives a DCO with a Path Sequence that is not newer than the Path Sequence as known from a Transit Information option in a DAO message, then the 6LR **MUST NOT** remove its current routing state, and it **MUST NOT** forward the DCO down a path where it is not newer. If the DCO is newer, the 6LR may retain a temporary state to ensure that a DAO that is received later with a Transit Information option with an older sequence number is ignored. A Transit Information option in a DAO message that is as new as or newer than that in a DCO wins, meaning that the path indicated in the DAO is installed and the DAO is propagated. When the DCO is propagated upon a DCO from an 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 the 'K' flag set. If the 'K' flag is not set, then the receiver of the DCO message **MAY** send a DCO-ACK, especially to report an error condition. The format of the DCO-ACK message is shown in Figure 4.

```
 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 the DCO-ACK is copied from the DCOSequence received in the DCO message.

DCO-ACK Status: Indicates completion status. The DCO-ACK Status field is defined based on Figure 6 of [RFC9010] defining the RPL Status Format. A StatusValue of 0 along with the 'U' bit set to 0 indicates Success / Unqualified acceptance as per Figure 6 of [RFC9010]. A StatusValue of 1 with the 'U' bit set to 1 indicates 'No routing entry' as defined in Section 5.3 of this document.

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 the 'D' flag is not set. The 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 shown in [RFC6550], Figure 7, where the base message format is the DCO-ACK message shown in Figure 4 of this document.

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** have the same values as those contained in 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 as new as or 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 a 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.

The scope of DCOSequence values is unique to the node that generates them.

4.5. Unsolicited DCO

A 6LR may generate an unsolicited DCO to unilaterally clean up 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 a DCO in its routing table. The conditions under which a 6LR may generate an unsolicited DCO are beyond the scope of this document, but possible reasons could be as follows:

1. On route expiry of an entry, a 6LR may decide to graciously clean up the entry by initiating a DCO.
2. A 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 by using a DCO.

A DCO that is generated asynchronously to a DAO message and is meant to discard all state along the path regardless of the Path Sequence **MUST** use a Path Sequence value of 240 (see Section 7.2 of [RFC6550]). This value allows the DCO to win against any established DAO path but to lose against a DAO path that is being installed. Note that if an ancestor initiates a unilateral path cleanup on an established path using a DCO with a Path Sequence value of 240, the DCO will eventually reach the target node, which will thus be informed of the path invalidation.

4.6. Other Considerations

4.6.1. Invalidation of Dependent Nodes

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

Dependent nodes do not have any indication regarding whether any of their parents have in turn decided to switch their parent. Thus, for route invalidation, the dependent nodes may choose to always set the 'I' flag in all their DAO messages' Transit Information options. 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 NPDAO mechanism provided in [RFC6550] can still be used in the same network where a DCO is used. 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 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 (Section 6.4.3) to that removed DAO parent to invalidate the existing route." This document introduces an alternative and more optimized way to perform 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 an NPDAO to invalidate the previous route, and send a regular DAO on the new path.
2. Send a 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 the reasons specified in Section 3 of this document.

This document assumes that all the 6LRs in the network support this specification. If there are 6LR nodes that do not support this document that are in the path of the DCO message transmission, then the route invalidation for the corresponding targets (targets that are in the DCO message) may not work or may work partially. Alternatively, a node could generate an NPDAO if it does not receive a DCO with itself as the target within a 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 an NPDAO and a DCO for the same operation would not result in unwanted side effects because the acceptability of an NPDAO or a DCO depends upon the Path Sequence freshness.

4.6.3. Considerations for DCO Retries

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. If the latency limits are not known, an implementation **MUST NOT** retry more than once in 3 seconds and **MUST NOT** retry more than three 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 Multi-Point to Point (MP2P) and Point-to-Multipoint (P2MP) flows, such as in Advanced Metering Infrastructure (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 that may have substantial P2P traffic, the 6LRs might be keen to invalidate efficiently because it may additionally impact 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, an NPDAO, which can be initiated with an updated Path Sequence to all the parent nodes through which the route is to be invalidated, can be used; see [RFC6550].

With a DCO, the target node itself does not initiate the route invalidation; this 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. It is recommended that an implementation initiate 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 which the route invalidation needs to be done.

This document recommends using a DelayDCO timer value of 1 second. This value is inspired by the default DelayDAO timer value of 1 second [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; this should not lead to any inconsistent routing states.

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

5. IANA Considerations

IANA has allocated codes for the DCO and DCO-ACK messages from the "RPL Control Codes" registry.

Code	Description	Reference
0x07	Destination Cleanup Object	This document
0x08	Destination Cleanup Object Acknowledgment	This document
0x87	Secure Destination Cleanup Object	This document
0x88	Secure Destination Cleanup Object Acknowledgment	This document

Table 1: New Codes for DCO and DCO-ACK Messages

IANA has allocated bit 1 from the "Transit Information Option Flags" registry for the 'I' flag (Invalidate previous route; see Section 4.2).

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

IANA has created a registry for the 8-bit Destination Cleanup Object (DCO) Flags field. The "Destination Cleanup Object (DCO) Flags" registry is located in the "Routing Protocol for Low Power and Lossy Networks (RPL)" registry.

New bit numbers may be allocated only by IETF Review [RFC8126]. Each bit is tracked with the following qualities:

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

The following bits are currently defined:

Bit number	Description	Reference
0	DCO-ACK request (K)	This document

Bit number	Description	Reference
1	DODAGID field is present (D)	This document

Table 2: DCO Base Flags

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

IANA has created a registry for the 8-bit Destination Cleanup Object (DCO) Acknowledgment Flags field. The "Destination Cleanup Object (DCO) Acknowledgment Flags" registry is located in the "Routing Protocol for Low Power and Lossy Networks (RPL)" registry.

New bit numbers may be allocated only by IETF Review [RFC8126]. Each bit is tracked with the following qualities:

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

The following bit is currently defined:

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

Table 3: DCO-ACK Base Flag

5.3. RPL Rejection Status Values

This document adds a new status value to the "RPL Rejection Status" subregistry initially created per Section 12.6 of [RFC9010].

Value	Meaning	Reference
1	No routing entry	This document

Table 4: Rejection Value of the RPL Status

6. 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 a 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 the RPL Status field to be used as a cover channel. But the channel only works once, since the message destroys its own medium, i.e., 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 routing 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 work only if the Path Sequence condition is also met for the target for which the invalidation is attempted. 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 by using existing mechanisms offered by [RFC6550], a malicious 6LR might also spoof a DAO with a 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; this allows integrity protection using either a Message Authentication Code (MAC) or a signature. Optionally, secured RPL messages also have encryption protection for confidentiality.

This document adds new messages (DCO and DCO-ACK) that are syntactically similar to existing RPL messages such as DAO and DAO-ACK. Secure versions of DCO and DCO-ACK messages are added in a way that is similar to the technique used for other RPL messages (such as DAO and DAO-ACK).

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

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 and DCO-ACK messages, do not have Security sections. Also note that unsecured mode does not imply that all messages are sent without any protection.

Preinstalled: In this mode, RPL uses secure messages. Thus, secure versions of DCO and DCO-ACK messages **MUST** be used in this mode.

Authenticated: In this mode, RPL uses secure messages. Thus, secure versions of DCO and DCO-ACK messages **MUST** be used in this mode.

7. 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>.

[RFC8126] Cotton, M., Leiba, B., and T. Narten, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 8126, DOI 10.17487/RFC8126, June 2017, <https://www.rfc-editor.org/info/rfc8126>.

[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>.

[RFC8505] Thubert, P., Ed., Nordmark, E., Chakrabarti, S., and C. Perkins, "Registration Extensions for IPv6 over Low-Power Wireless Personal Area Network (6LoWPAN) Neighbor Discovery", RFC 8505, DOI 10.17487/RFC8505, November 2018, <https://www.rfc-editor.org/info/rfc8505>.

[RFC9010] Thubert, P., Ed. and M. Richardson, "Routing for RPL (Routing Protocol for Low-Power and Lossy Networks) Leaves", RFC 9010, DOI 10.17487/RFC9010, April 2021, <https://www.rfc-editor.org/info/rfc9010>.

Appendix A. Example Messaging

A.1. Example DCO Messaging

In this example, Node D (Figure 1) 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 conventions and specifies only the required parameters to explain the example, namely, the parameter 'tgt', which stands for "Target option"; the value of this parameter specifies the address of the target node. The parameter 'pathseq' specifies the Path Sequence value carried in the Transit Information option, and the parameter 'I_flag' specifies the 'I' flag in the Transit Information option. The 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 the DCO(tgt=D,pathseq=x+1) to the 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 using the DAO(tgt=D,pathseq=x+1,I_flag=1).
6. Node G receives the DCO(tgt=D,pathseq=x+1). It checks to see 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 the 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 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.

A.2. Example DCO Messaging with Multiple Preferred Parents

As shown in Figure 5, node (N41) selects multiple preferred parents (N32) and (N33). The sequence of actions is listed below the figure.

```
       (6LBR)
         |
         |
         |
       (N11)
        / \
       /   \
      /     \
   (N21)   (N22)
     /      / \
    /      /   \
   /      /     \
(N31)  (N32)  (N33)
    :    |    /
     :   |   /
      :  |  /
       (N41)
```

Figure 5: Sample Topology 2

1. (N41) sends a DAO(tgt=N41,PS=x,I_flag=1) to (N32) and (N33). Here, 'I_flag' refers to the Invalidation flag, and 'PS' refers to the Path Sequence in the Transit Information option.
2. (N32) sends the DAO(tgt=N41,PS=x,I_flag=1) to (N22). (N33) also sends the 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 routing table at N22 should contain (Dst,NextHop,PS): { (N41,N32,x), (N41,N33,x) }.

3. (N22) sends the DAO(tgt=N41,PS=x,I_flag=1) to (N11).
4. (N11) sends the DAO(tgt=N41,PS=x,I_flag=1) to (6LBR). Thus, the complete path is established.
5. (N41) decides to change the preferred parent set from { N32, N33 } to { N31, N32 }.
6. (N41) sends the DAO(tgt=N41,PS=x+1,I_flag=1) to (N32). (N41) sends the DAO(tgt=N41,PS=x+1,I_flag=1) to (N31).
7. (N32) sends the 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 waits for a predetermined time period (the DelayDCO time period) to invalidate another route { (N41),(N33),x }. After the time period, (N22) sends the 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 the DCO(tgt=N41,PS=x+1). The received Path Sequence is the latest and thus invalidates the entry associated with the 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 the DAO(tgt=N41,PS=x+1,I_flag=1) from (N21). It waits for the DelayDCO timer, since it has multiple routes to (N41). (N41) will receive the DAO(tgt=N41,PS=x+1,I_flag=1) from (N22) from Step 7 above. Thus, (N11) has received the regular DAO(tgt=N41,PS=x+1,I_flag=1) from all paths and thus does not initiate the DCO.
11. (N11) forwards the DAO(tgt=N41,PS=x+1,I_flag=1) to (6LBR), and the full path is established.

Acknowledgments

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

Authors' Addresses

Rahul Arvind Jadhav (EDITOR)
Huawei
Whitefield
Kundalahalli Village
Bangalore 560037
Karnataka
India
Phone: +91-080-49160700
Email: rahul.ietf@gmail.com

Pascal Thubert
Cisco Systems, Inc.
Building D
45 Allee des Ormes - BP1200
06254 MOUGINS - Sophia Antipolis
France
Phone: +33-497-23-26-34
Email: pthubert@cisco.com

Rabi Narayan Sahoo
Huawei
Whitefield
Kundalahalli Village
Bangalore 560037
Karnataka
India
Phone: +91-080-49160700
Email: rabinarayans0828@gmail.com

Zhen Cao
Huawei
W Chang'an Ave
Beijing
China
Email: zhencao.ietf@gmail.com