ROLL R. Jadhav, Ed.
Internet-Draft Huawei
Intended status: Standards Track P. Thubert
Expires: April 17, 2019 Cisco
R. Sahoo
Z. Cao
Huawei
October 14, 2018

Efficient Route Invalidation
draft-ietf-roll-efficient-npdao-09

Abstract

This document describes the problems associated with NPDAO messaging used in RPL for route invalidation and signaling changes to improve route invalidation efficiency.

Status of This Memo

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 an optional messaging in the form of DAO (Destination Advertisement Object) messages using which the 6LBR (6Lo Border Router) and 6LR

(6Lo Router) can learn route towards the downstream nodes. In storing mode, DAO messages would result in routing entries been created on all intermediate 6LRs from the node's parent all the way towards the 6LBR.

RPL allows 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 pro-active 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", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

6LR: 6LoWPAN Router. This is an intermediate 6lowpan router which allows traffic routing through itself in a multihop 6lo network.

DAG: Directed Acyclic Graph. A directed graph having the property that all edges are oriented in such a way that no cycles exist.

DODAG: Destination-oriented DAG. A DAG rooted at a single destination, i.e., at a single DAG root with no outgoing edges.

6LBR: 6LoWPAN Border Router. A border router which is a DODAG root and is the edge node for traffic flowing in and out of the 6lo network.

DAO: Destination Advertisement Object. DAO messaging allows downstream routes to the nodes to be established.

DIO: DODAG Information Object. 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.

NPDAO: No-Path DAO. A DAO message which has target with lifetime 0.

DCO: Destination Cleanup Object, A new RPL control message type defined by this draft. DCO messaging improves proactive route invalidation in RPL.

Regular DAO: A DAO message with non-zero lifetime.

LLN: Low Power and Lossy Networks.

Target Node: The node switching its parent whose routing adjacencies are updated (created/removed).

This document also uses terminology described in [RFC6550].

1.2. Current NPDAO messaging

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

For the rest of this document consider the following topology:

```
                       (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 NPDAO to (B) and regular DAO to (C).

1.3. Why NPDAO is 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 achieve resource utilization. Thus it becomes necessary to have 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 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 async 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 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 (thats 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. Proposed changes to RPL signaling

4.1. Change in RPL route invalidation semantics

As described in Section 1.2, the NPDAO originates at the node switching the parent and traverses upstream towards the root. In order to solve the problems as mentioned in Section 2, the draft adds new pro-active route invalidation message called as "Destination Cleanup Object" (DCO) that originates at a common ancestor node

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.

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 target as address of D and a incremented path sequence number. Node C will update the routing table based on the reachability information in 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. the 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.

4.2. Transit Information Option changes

Every RPL message is divided into base message fields and additional Options. 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.

The draft proposes a change in Transit Information option to contain "Invalidate previous route" (I) bit. This I-bit signals the common ancestor node to generate a DCO on behalf of the target node. The I-bit is carried in the transit information option which augments the reachability information for a given set of RPL Target(s). Transit information option should be carried in the DAO message with I-bit set in case route invalidation is sought for the correspondig 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 |                               |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+                               +
     |                                                               |
     +                                                               +
     |                                                               |
     +                        Parent Address*                        +
     |                                                               |
     +                               +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |                               |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 2: Updated Transit Information Option (New I flag added)

I (Invalidate previous route) bit: 1 bit flag. The 'I' flag is set by the target node to indicate that it wishes to invalidate the previous route by a common ancestor node between the two paths.

The common ancestor node SHOULD generate a DCO message in response to this I-bit when it sees that the routing adjacencies have changed for the target. I-bit governs the ownership of the DCO message in a way that the target node is still in control of its own route invalidation.

4.3. Destination Cleanup Object (DCO)

A new ICMPv6 RPL control message type is defined by this specification called 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   |   Reserved    | 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 is expected to send a DCO-ACK back. If the DCO-ACK is not received even after setting the 'K', an implementation may choose to retry the DCO at a later time. The number of retries are implementation and deployment dependent. This document recommends using retries similar to what will be set for DAO-ACK handling.

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.

Reserved: 8-bit unused field. The field MUST be initialized to zero by the sender and MUST be ignored by the receiver.

DCOSequence: 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.

DODAGID (optional): 128-bit unsigned integer set by a DODAG root that uniquely identifies a DODAG. This field is only present when the 'D' flag is set. This field is typically only present when a local RPLInstanceID is in use, in order to identify the DODAGID that is associated with the RPLInstanceID. When a global RPLInstanceID is in use, this field need not be present. Unassigned bits of the DCO Base

are reserved. They MUST be set to zero on transmission and MUST be ignored on reception.

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 MAY carry 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

The DCO carries a 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 DAO message. The DAO and DCO path sequence are picked from the same sequence number set. Thus if a DCO is received by a 6LR and subsequently a DAO is received with old seqeunce number, then the DAO should be ignored.

4.3.4. Destination Cleanup Option Acknowledgement (DCO-ACK)

The DCO-ACK message may be sent as a unicast packet by a DCO recipient in response to a unicast DCO message.

```
     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|  Reserved   |  DCOSequence  |    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.

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

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

Status: Indicates the completion. Status 0 is defined as unqualified acceptance in this specification. 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 is only present when the 'D' flag is set. This field is typically only present when a local RPLInstanceID is in use, in order to identify the DODAGID that is associated with the RPLInstanceID. When a global RPLInstanceID is in use, this field need not be present. Unassigned bits of the DCO-Ack Base are reserved. They MUST be set to zero on transmission and MUST be ignored on reception.

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. Other considerations

4.4.1. 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-bit in the transit information option as part of regular DAO so as to request invalidation of previous route from the common ancestor node.

4.4.2. NPDAO and DCO in the same network

Even with the changed semantics, the current NPDAO mechanism in [RFC6550] can still be used, for example, when the route lifetime expiry of the target happens 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 proposed DCO and the existing NPDAO mechanism.

4.4.3. 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". Thus a DAO message with the same path sequence MUST be sent to all the parents. Subsequently when route invalidation has to be initiated, RPL mentions that an NPDAO must be initiated with updated path sequence to all the routes 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 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. The time period for initiating a DCO could be based on the depth of the network. After timeout, the DCO needs to be generated for all the next-hops for whom the route invalidation needs to be done.

5. Acknowledgements

Many thanks to Cenk Gundogan, Simon Duquennoy, Georgios Papadopoulous, Peter Van Der Stok for their review and comments.

6. IANA Considerations

IANA is requested to allocate new ICMPv6 RPL control codes in RPL [RFC6550] for DCO and DCO-ACK messages.

Code	Description	Reference
0x04	Destination Cleanup Object	This document
0x05	Destination Cleanup Object Acknowledgement	This document
0x84	Secure Destination Cleanup Object	This document
0x85	Secure Destination Cleanup Object Acknowledgement	This document

IANA is requested to allocate bit 18 in the Transit Information Option defined in RPL [RFC6550] section 6.7.8 for Invalidate route 'I' flag.

7. Security Considerations

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.
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

[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>.

Appendix A. Example Messaging

A.1. Example DCO Messaging

In Figure 1, node (D) switches its parent from (B) to (C). 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 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.
4. Similar to C, node H checks for 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 H in a DAO.
5. Node A receives the DAO, and checks for routing entry on behalf of D. It finds a routing entry but checks that the next hop for target D is now changed. Node A checks the I_flag and generates DCO(tgt=D,pathseq=pathseq(DAO)) to previous next hop for target D which is G. Subsequently, A updates the routing entry and forwards the reachability information of target D upstream DAO(tgt=D,pathseq=x+1,I_flag=x) (the I_flag carries no significance henceforth).
6. Node G receives the DCO and invalidates routing entry of target D and forwards the (un)reachability information downstream to B.
7. Similarly, B processes the DCO by invalidating the routing entry of target D and forwards the (un)reachability information downstream to D.

8. D ignores the DCO 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 the routing information is present and the pathseq associated is not older, then still the DCO is dropped.

A.2. 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. 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 to invalidate another route {(N41),(N33),x}. After time period, (N22) sends DCO(tgt=N41,PS=x+1) to (N33).

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
China

Email: zhencao.ietf@gmail.com