ROLL R. Jadhav, Ed.
Internet-Draft R. Sahoo
Intended status: Standards Track Z. Cao
Expires: December 28, 2017 Huawei Tech
June 26, 2017

No-Path DAO modifications
draft-ietf-roll-efficient-npdao-00

Abstract

This document describes the problems associated with the use of No-Path DAO messaging in RPL and a signaling changes to improve route invalidation efficiency.

Table of Contents

1. Introduction

RPL [RFC6550] specifies a proactive distance-vector based routing scheme. The specification has an optional messaging in the form of DAO messages using which the 6LBR can learn route towards any of the nodes. In storing mode, DAO messages would result in routing entries been created on all intermediate hops 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 No-Path DAO is a DAO message with

route lifetime of zero, signaling route invalidation for the given target. This document explains the problems associated with the current use of NPDAO messaging and also discusses the requirements for an optimized No-Path DAO messaging scheme. The signalling change specified fulfills all mentioned requirements of an optimized NPDAO messaging.

6TiSCH architecture [I-D.ietf-6tisch-architecture] leverages RPL and specifies use of non-storing and storing MOP for its routing operation. Thus an improvement in NPDAO messaging will help optimize 6TiSCH based networks.

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].

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.

Common Ancestor node: 6LR node which is the first common node on the old and new path for the child node.

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

Reverse NPDAO: A No-Path DAO message which traverses downstream in the network.

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

This document also uses terminology described in [RFC6550].

1.2. Current No-Path DAO messaging

RPL introduced No-Path DAO messaging in the storing mode so that the node switching its current parent can inform its parents and ancestors to invalidate the existing route. Subsequently parents or ancestors would release any resources (such as the routing entry) it maintains on behalf of target node. The NPDAO message always traverses the RPL tree in upward direction, originating at the target node itself.

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 BR. Node (A) is the common ancestor for (D) for paths through (B)-(G) and (C)-(H). When (D) switches from (B) to (C), [RFC6550] suggests sending No-Path DAO to (B) and regular DAO to (C).

1.3. Cases when No-Path DAO may be used

There are following cases in which a node switches its parent and may employ No-Path DAO messaging:

Case I: Current parent becomes unavailable because of transient or permanent link or parent node failure.

Case II: The node finds a better parent node i.e. the metrics of another parent is better than its current parent.

Case III: The node switches to a new parent whom it "thinks" has a better metric but does not in reality.

The usual steps of operation when the node switches the parent is that the node sends a No-Path DAO message via its current parent to invalidate its current route and subsequently it tries to establish a new routing path by sending a new DAO via its new parent.

1.4. Why No-Path DAO is important?

Nodes in LLNs may be resource constrained. There is limited memory available and routing entry records are the 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 in case of contention. 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 invalidation needs to be done for (D), (E) and (F). Thus without efficient route invalidation, a 6LR may have to hold a lot of unwanted route entries.

2. Problems with current No-Path DAO messaging

2.1. Lost NP-DAO due to link break to the previous parent

When a node switches its parent, the NPDAO is to be sent via its previous parent and a regular DAO via 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. RPL assumes communication link with the previous parent for No-Path DAO messaging.

RPL allows use of route lifetime to remove unwanted routes in case the routes could not be refreshed. But route lifetimes in case of LLNs could be substantially high and thus the route entries would be stuck for long.

2.2. Invalidate routes to dependent nodes of the switching node

No-path DAO is sent by the node who has switched the parent but it does not work for the dependent child nodes below it. The specification does not specify how route invalidation will work for sub-childs, resulting in stale routing entries on behalf of the sub-childs on the previous route. 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. Post switching, Node (D) transmits a DIO with incremented DTSN so that child nodes, node (E) and (F), generate DAOs to trigger route update on the new path for themselves. There is no NPDAO generated by these child nodes through the previous path resulting in stale entries on nodes (B) and (G) for nodes (E) and (F).

2.3. 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 thus impacting downward traffic for the switching node. In the example topology, consider Node (D) switches from parent (B) to (C) because the metrics of the path via (C) are better. Note that the previous path via (B) may still be available (albeit at relatively bad metrics). An NPDAO sent from previous route may invalidate the existing route whereas there is no way to determine whether the new DAO has successfully updated the route entries on the new path.

An implementation technique to avoid this problem is to further delay the route invalidation by a fixed time interval after receiving an NPDAO, considering the time taken for the new path to be established. Coming up with such a time interval is tricky since the new route may also not be available and it may subsequently require more parent switches to establish a new path.

3. Requirements for the No-Path DAO Optimization

3.1. Req#1: Tolerant to the link failures to the previous parents

When the switching node send the NP-DAO message to the previous parent, it is normal that the link to the previous parent is prone to failure. Therefore, it is required that the NP-DAO message MUST be tolerant to the link failure during the switching.

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

While switching the parent node and sending NP-DAO message, it is required that the routing entries to the dependent nodes of the switching node will be updated accordingly on the previous parents and other relevant upstream nodes.

3.3. Req#3: No impact on traffic while NP-DAO operation in progress

While sending the NP-DAO and DAO messages, it is possible that the NP-DAO successfully invalidates the previous path, while the newly sent DAO gets lost (new path not set up successfully). This will result into downstream unreachability to the current switching node. Therefore, it is desirable that the NP-DAO is synchronized with the DAO to avoid the risk of route downtime.

4. Proposed changes to NPDAO signaling

4.1. Change in NPDAO semantics

As described in Section 1.2, currently 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 proposes to change the way NPDAO originates and traverses the network. The proposed NPDAO originates at a common ancestor node between the new and old path. The trigger for the common ancestor node to generate this NPDAO is the change in the next hop for the node on reception of an update message in the form of regular DAO for the target.

In the 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), may generate an NPDAO which traverses downwards in the network. The document in the subsequent section will explain the message format changes to handle this downward flow of NPDAO.

4.2. DAO message format 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 specifies 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 DAO message to contain "Invalidate previous route" (I) bit. This I-bit which is carried in regular DAO message, signals the common ancestor node to generate a downstream NPDAO on behalf of the target node. The I-bit is carried in the transit container option which augments the reachability information for a given set of RPL 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 NPDAO thus generated by the common ancestor node needs to traverse downstream. An additional flag called as "Reverse NPDAO" (R) is added in the base DAO object to signal this change.

```
     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|R| Flags   |   Reserved    | DAOSequence   |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                                                               |
    +                                                               +
    |                                                               |
    +                            DODAGID*                           +
    |                                                               |
    +                                                               +
    |                                                               |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |   Option(s)...
    +-+-+-+-+-+-+-+-+
```

Figure 3: Updated DAO base object (New R flag added)

R (Reverse DAO) bit: 1 bit flag. The 'R' flag is used to signal that the DAO traverses downwards.

4.2.1. Path Sequence number in the reverse NPDAO

Every DAO message may contain a Path Sequence in the transit information option to identify the freshness of the DAO message. The Path Sequence in the downward NPDAO generated by common ancestor should use the same Path Sequence number present in the regular DAO message.

4.3. Example 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 downstream NPDAO(tgt=D,pathseq=x+1,R_flag=1) 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 downstream NPDAO and invalidates routing entry of target D and then checks the reverse (R) flag and forwards the (un)reachability information downstream to B.

7. Similarly, B processes the downstream NPDAO by invalidating the routing entry of target D and then checks the reverse (R) flag and forwards the (un)reachability information downstream to D.

8. D ignores the downstream NPDAO since the target is itself.

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 section describes approaches for invalidating routes of dependent nodes if the implementation chooses to solve this problem. The common ancestor node realizes that the paths for dependent nodes have changed (based on next hop change) when it receives a regular DAO on behalf of the dependent nodes. Thus dependent nodes route invalidation can be handled in the same way as the switching node. Note that there is no way that dependent nodes can set the I_flag in the DAO message selectively since they are unaware that their parent/ grand parent node is switching paths. There are two ways to handle dependent node route invalidation:

1. One way to resolve is that the common ancestor does not depend upon the I_flag to generate the reverse NPDAO. The only factor it makes the decision will be based on next_hop change for an existing target to generate the NPDAO. Thus when the switching nodes and all the below dependent nodes advertise a regular DAO, the common ancestor node will detect a change in next hop and generate NPDAO for the same target as in the regular DAO.

2. Another way is that the nodes always set the I_flag whenever they send regular DAO. Thus common ancestor will first check whether I_flag is set and then check whether the next_hop has changed and subsequently trigger NPDAO if required.

This document recommends the approach in point 2. The advantage with I_flag is that the generation of downstream NPDAO is still controlled by the target node and thus is still in control of its own routing state.

5. Acknowledgements

We would like to thank Cenk Gundogan, Simon Duquennoy and Pascal

Thubert for their review and insightful comments.

6. IANA Considerations

IANA is requested to allocate bit 11 in the DAO base object defined in RPL [RFC6550] section 6.4 for reverse 'R' NPDAO flag.

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

This draft does not add any new messages but extends existing messaging. The seucrity considerations applicable to DAO messaging in RPL is also applicable here.

8. References

8.1. Normative References

[I-D.ietf-6tisch-architecture]
Thubert, P., "An Architecture for IPv6 over the TSCH mode of IEEE 802.15.4", draft-ietf-6tisch-architecture-11 (work in progress), January 2017.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <http://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, <http://www.rfc-editor.org/info/rfc6550>.

8.2. Informative References

[CONTIKI] Thingsquare, "Contiki: The Open Source OS for IoT", 2012, <http://www.contiki-os.org>.

[RFC3552] Rescorla, E. and B. Korver, "Guidelines for Writing RFC Text on Security Considerations", BCP 72, RFC 3552, DOI 10.17487/RFC3552, July 2003, <http://www.rfc-editor.org/info/rfc3552>.

Appendix A. Additional Stuff

This becomes an Appendix.

Authors' Addresses

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

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

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

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

Zhen Cao
Huawei Tech
W Chang'an Ave

Beijing 560037
China

Email: zhencao.ietf@gmail.com