

ROLL WG  
INTERNET-DRAFT  
Intended Status: Informational  
Expires: December 27, 2016

R. Jadhav  
R. Sahoo  
Z. Cao  
Huawei Tech  
H. Deng  
China Mobile  
June 27, 2016

**No-Path DAO Problem Statement**  
**draft-jadhav-roll-no-path-dao-ps-01**

**Abstract**

This document describes the problems associated with the use of No-Path DAO messaging in RPL.

**Status of this Memo**

This Internet-Draft is submitted to IETF in full conformance with the provisions of [BCP 78](#) and [BCP 79](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at <http://www.ietf.org/lid-abstracts.html>

The list of Internet-Draft Shadow Directories can be accessed at <http://www.ietf.org/shadow.html>

**Copyright and License Notice**

Copyright (c) 2016 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](#) and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of

publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

<a href="#">1.</a>	<a href="#">Introduction . . . . .</a>	<a href="#">3</a>
<a href="#">1.1.</a>	<a href="#">Current No-Path DAO messaging . . . . .</a>	<a href="#">3</a>
<a href="#">1.2.</a>	<a href="#">Cases when No-Path DAO may be used . . . . .</a>	<a href="#">4</a>
<a href="#">1.3.</a>	<a href="#">Why No-Path DAO is important? . . . . .</a>	<a href="#">5</a>
<a href="#">1.4.</a>	<a href="#">Terminology . . . . .</a>	<a href="#">5</a>
<a href="#">2.</a>	<a href="#">Problems with current No-Path DAO messaging . . . . .</a>	<a href="#">5</a>
<a href="#">2.1.</a>	<a href="#">Lost NP-DAO due to Link break to the previous parent . . . . .</a>	<a href="#">5</a>
<a href="#">2.2.</a>	<a href="#">Invalidate routes to dependent nodes of the switching node . . . . .</a>	<a href="#">6</a>
<a href="#">2.3.</a>	<a href="#">Route downtime caused by asynchronous operation of NPDAO and DAO . . . . .</a>	<a href="#">6</a>
<a href="#">3.</a>	<a href="#">Requirements for the No-Path DAO Optimization . . . . .</a>	<a href="#">6</a>
<a href="#">3.1.</a>	<a href="#">Req#1: Tolerant to the link failures to the previous parents. . . . .</a>	<a href="#">7</a>
<a href="#">3.2.</a>	<a href="#">Req#2: Support of removal of entries to the dependent nodes of the switching node. . . . .</a>	<a href="#">7</a>
<a href="#">3.3.</a>	<a href="#">Req#3: No disruption of downstream reachability to the node while sending NP-DAO. . . . .</a>	<a href="#">7</a>
<a href="#">4.</a>	<a href="#">Existing Solution . . . . .</a>	<a href="#">7</a>
<a href="#">4.1.</a>	<a href="#">NP-DAO can be generated by the parent node who detects link failure to the child . . . . .</a>	<a href="#">7</a>
<a href="#">4.2.</a>	<a href="#">NP-DAO can be generated once the link is restored to the previous parent. . . . .</a>	<a href="#">8</a>
<a href="#">5.</a>	<a href="#">Security Considerations . . . . .</a>	<a href="#">9</a>
<a href="#">6.</a>	<a href="#">IANA Considerations . . . . .</a>	<a href="#">9</a>
<a href="#">7.</a>	<a href="#">Acknowledgements . . . . .</a>	<a href="#">9</a>
<a href="#">8.</a>	<a href="#">References . . . . .</a>	<a href="#">9</a>
<a href="#">8.1.</a>	<a href="#">Normative References . . . . .</a>	<a href="#">9</a>
<a href="#">8.2.</a>	<a href="#">Informative References . . . . .</a>	<a href="#">9</a>
	<a href="#">Authors' Addresses . . . . .</a>	<a href="#">9</a>



## **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.

[RFC6550] also allows use of No-Path DAO (NPDAO) messaging to invalidate a routing path and thus releasing of any 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 studies the problems associated with the current use of No-Path DAO messaging, which creates route inefficiency and inconsistency. This document also discusses the requirements for an optimized No-Path DAO messaging scheme.

### **1.1. Current No-Path DAO messaging**

[RFC6550] 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 that child node. The No-Path DAO 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:



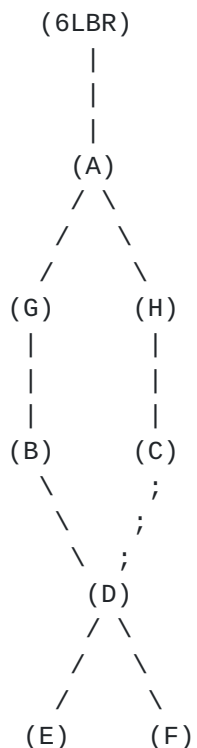


Figure 1: Sample Topology

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).

### **1.2. 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.



### **1.3. 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.

### **1.4. 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](#)].

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

Current parent: Parent 6LR node before switching to the new path

New parent: Parent 6LR node after switching to the new path

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](#)].

## **2. Problems with current No-Path DAO messaging**

We will confront the following problems when using the current NP-DAO messaging.

### **2.1. Lost NP-DAO due to Link break to the previous parent**

When the 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. [[RFC6550](#)]



assumes communication link with the previous parent for No-Path DAO messaging.

[RFC6550] mentions 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-children, resulting in stale routing entries on behalf of the sub-children 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**



We identify the following requirements for the NP-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: Support of removal of entries to the dependent nodes of the switching node.**

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 disruption of downstream reachability to the node while sending NP-DAO.**

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 routing downtime.

## **[4.](#) Existing Solution**

### **[4.1.](#) NP-DAO can be generated by the parent node who detects link failure to the child**

RPL [[RFC6550](#)] states mechanisms which could be utilized to clear DAO states in a sub-DODAG. [[RFC6550](#)] [Section 11.2.2.3](#) states "With DAO inconsistency loop recovery, a packet can be used to recursively explore and clean up the obsolete DAO states along a sub-DODAG."

Thus in the sample topology in Figure 1, when Node (B) detects link failure to (D), (B) has an option of generating an NP-DAO on behalf of Node (D) and its sub-children, (E) and (F).

This section explains why generation of an NP-DAO in such cases may not function as desired. Primarily the DAO state information in the form of Path Sequence plays a major role here. Every target is associated with a Path Sequence number which relates to the latest state of the target. [[RFC6550](#)] [Section 7.1](#) explains the semantics of Path Sequence number. The target node increments the Path Sequence number every time it generates a new DAO. The router nodes en-route



utilize this Path Sequence number to decide the freshness of target information. If a non-target node has to generate an NP-DAO then it could use following two possibilities with Path Sequence number:

Let the Path Sequence number of old regular DAO that flowed through (B) be  $x$ . The subsequent regular DAO generated by Node (D) will have sequence number  $x+1$ .

i. Node (B) uses the previous Path Sequence number from the regular DAO i.e. NP-DAO(pathseq= $x$ )

ii. Node (B) increments the Path Sequence number i.e. NP-DAO(pathseq= $x+1$ )

In case i, the NP-DAO(pathseq= $x$ ) will be dropped by all the intermediate nodes since the semantics of Path Sequence number dictates that any DAO with an older Path Sequence number be dropped.

In case ii, there is a risk that the NP-DAO(pathseq= $x+1$ ) traverses up the DODAG and invalidates all the routes till the root and then the regular DAO(pathseq= $x+1$ ) from the target traverses upwards. In this case the regular DAO(pathseq= $x+1$ ) will be dropped from common ancestor node to the root. This will result in route downtime.

Another problem with this scheme is its dependence on the upstream neighbor to detect that the downstream neighbor is unavailable. There are two possibilities by which such a detection might be put to work:

i. There is P2P traffic from the previous sub-DODAG to any of nodes in the sub-tree which has switched the path. In the above example, lets consider that Node (G) has P2P traffic for either of nodes (D), (E), or (F). In this case, Node (B) will detect forwarding error while forwarding the packets from Node (B) to (D). But dependence on P2P traffic may not be an optimal way to solve this problem considering the reactive approach of the scheme. The P2P traffic pattern might be sparse and thus such a detection might kick-in too late.

ii. The other case is where Node (B) explicitly employs some mechanism to probe directly attached downstream child nodes. Such kind of schemes are seldom used.

#### **4.2. NP-DAO can be generated once the link is restored to the previous parent.**

This scheme solves a specific scenario of transient links. The child node can detect that the connection to previous parent is restored and then transmit an NP-DAO to the previous parent to invalidate the



route. This scheme is stateful, thus requires more memory and solves a specific scenario.

## **5. Security Considerations**

This draft is a problem statement, and therefore, does not introduce any new security risks.

## **6. IANA Considerations**

Not applicable to this document.

## **7. Acknowledgements**

We would like to thank Cenk Gundogan, Simon Duquennoy and Pascal Thubert for their review and insightful comments.

## **8. References**

### **8.1. 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.
- [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.
- [RFC6551] Vasseur, JP., Ed., Kim, M., Ed., Pister, K., Dejean, N., and D. Barthel, "Routing Metrics Used for Path Calculation in Low-Power and Lossy Networks", [RFC 6551](#), DOI 10.17487/RFC6551, March 2012.

### **8.2. Informative References**

- [RFC5548] Dohler, M., Ed., Watteyne, T., Ed., Winter, T., Ed., and D. Barthel, Ed., "Routing Requirements for Urban Low-Power and Lossy Networks", [RFC 5548](#), May 2009.

## **Authors' Addresses**

Rahul Arvind Jadhav  
Huawei Tech,  
Kundalahalli Village,



Bangalore, India

EMail: rahul.jadhav@huawei.com

Rabi Narayan Sahoo  
Huawei Tech,  
Kundalahalli Village,  
Bangalore, India

EMail: rabinarayans@huawei.com

Zhen Cao  
Huawei Tech,  
Beijing, China

EMail: zhen.cao@huawei.com

Hui Deng  
China Mobile,  
Beijing, China

EMail: denghui@chinamobile.com

