ROLL Working Group
Internet-Draft
Intended status: Informational
Expires: December 29, 2015
M.I. Robles
Ericsson
M. Richardson
SSW
June 27, 2015

When to use RFC 6553, 6554 and IPv6-in-IPv6
draft-robles-roll-useofrplinfo-00

Abstract

This document states different cases where RFC 6553, RFC 6554 and IPv6-in-IPv6 encapsulation is required to set the bases to help defining the compression of RPL routing information in LLN environments.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

RPL [RFC6550] defines RPL Option to transmit routing information. RFC 6553 [RFC6553] defines how to transmit in a Hop-By-Hop Option RPL Information,such as information to avoid and detect loops. RFC 6554 [RFC6554] defines the use of Extension header for Source Routing.

Several discussions in ROLL/6lo/6tisch Mailing Lists took place focusing in the definition how to compress RPL Information in constrained environment. ROLL Virtual Interim Meeting (02-2015) concluded that there is a need to define how to use RFC 6553, RFC6554 and tunneling (IP-in-IP) to be able to set the correct environment for compression.

2. Terminology and Requirements Language

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].

Terminology defined in [RFC7102]

3. Sample/reference topology

In a typical topology we found 6LBR (6LoWPAN Border Router), 6lR (6LoWPAN Router) and 6LN (6LoWPAN Node) as leaf connected in a DODAG (Destination Oriented Directed Acyclic Graph). Between these entities messages such as DIS, DIO and DAO are transmitted. RPL defines the RPL Control message as an ICMPv6 information message with a Type of 155. RPL supports two modes of Downward traffic: Storing, it is fully stateful or Non-Storing it is fully source routed. Any given RPL Instance is either storing or non-storing.

```
+--------------+
| Upper Layers |
|              |
+--------------+
|   RPL        |
|              |
+--------------+
|   ICMPv6     |
|              |
+--------------+
|   IPv6       |
|              |
+--------------+
|   6LoWPAN    |
|              |
+--------------+
|   PHY-MAC    |
|              |
+--------------+
```

Figure 1: RPL Stack

```
                                          +---------+
                                      +---+Internet |
                                      |   +---------+
                                      |
                                 +----+--+
                                 |DODAG  |
                       +---------+Root   +----------+
                       |         |6LBR   |          |
                       |         +----+--+          |
                       |              |             |
                       |              |             |
                       |              |             |
                 +-----+-+         +--+---+      +--+---+
                 |6LR    |         |      |      |      |
           +-----+       |         |      |      |      |
           |     |       |         |      |      |      +------+
           |     +-----+-+         +-+----+      +-+----+      |
           |           |             |             |           |
           |           |             |             |           |
           |           |             |             |           |
         +-+---+     +-+---+      +--+--+       +- --+     +---+-+
         |Leaf |     |     |      |     |       |    |     |     |
         |6LN  |     |     |      |     |       |    |     |     |
         +-----+     +-----+      +-----+       +----+     +-----+
```

Figure 2: A reference RPL Topology

In different scenarios the use of RFC 6553, RFC 6554 and tunneling can take place:

-Flow from leaf to root

-Flow from leaf to Internet

-Flow from leaf to leaf

-Flow from Internet to leaf

-Flow from leaf to root

4. Example flow from leaf to root

A leaf node generates DAO and DIS messages and in general does not accept them. Additionally, this kind of nodes accepts DIO messages, but in general do not generate them. (In inconsistency A leaf node generates DIO with infinite rank, to fix it).

4.1. Non-storing

In non-storing in this case the leaf node uses Hop-By-Hop option (RFC 6553) to indicate the routing information to send messages to the DODAG root, this message is going to be analyzed in each node until arrive the DODAG root.

RFC 6554 was created to strictly send information between RPL routers in the same RPL routing domain. How it would be in 6554?

TBD: Tunneling is necessary in case that there is information to send outside RPL Domain and other cases?

```
                          +------+
                          |      |
                          | 6LBR |
                          |      |
                          +---+--+
                              |
                              |  LoWPAN_HC
                              |  Route= 6LN-6LR-6LBR
                       ^      |
                       |  +---+-+
                       |  |     |
                       |  | 6LR |
                       |  |     |
                       |  +--+--+
                       |     |   LoWPAN_HC
                       |     |   Route= 6LN-6LR-6LBR
                       |     |
                       +     |
                          +--+--+
                          | 6LN |
                          |     |
                          |     |
                          +-----+
```

Figure 3: From leaf to Root - Non-Storing Mode

4.2. Storing

IP6 6553{X,Y] ?ipip payload. In storing mode is suitable the use of RFC 6553 to send RPL Information through HBH field checking the routing table to find out where to send the message. It may include IP-in-IP encapsulation to transmit information not related with the RPL domain.

```
                      +------+
                      |      |
                      | 6LBR |
                      |      |
                      +---+--+
                          |
                          |  LoWPAN_HC
                          |  0x63|HBH Data
                   ^      |
                   |  +---+-+
                   |  |     |
                   |  | 6LR | 6LR check in routing table
                   |  |     |
                   |  +--+--+
                   |     |   LoWPAN_HC
                   |     |   0x63|HBH Data
                   |     |
                   +     |
                      +--+--+
                      | 6LN |
                      |     |
                      |     |
                      +-----+
```

Figure 4: From leaf to Root - Storing Mode

5. Example flow from leaf to Internet

5.1. Non-storing

In this case the IP-in-IP encapsulation should take place to send information not related to the RPL domain inside of the RPL domain.

RPL information from RFC 6553 should not go out to Internet. The router sould take this information out before send the packet to Internet. The HBH Option is going to be analyzed in each node to the root.

Related to RFC 6554 the Source Header route is added and removed by DODAG root. However, RFC 6554 was created to strictly send information between RPL routers in the same RPL routing domain. How it would be in 6554?

5.2. Storing

In storing the information of RFC 6553 should take away by DODAG root before go to Internet.

6. Example flow from leaf to leaf

can leafs insert appropriate headers, without ipip? In [RFC6550] RPL allows a simple one-hop P2P optimization for both storing and non-storing networks. A node may send a P2P packet destined to a one-hop neighbor direclty to that node. Section 9 in [RFC6550].

6.1. Traditional storing

The route go through an ancestor that knows the route to the destination, using HBH [RFC6553] to carry RPL Information.

6.2. Traditional non-storing

The route go through the DODAG root, using source routing [RFC6554].

6.3. P2P non-storing

(p2p storing? TBD)

7. Example flow from Internet to leaf

A DODAG root do not add routing extension to incoming packets, it instead uses tunneling.

7.1. Storing

DODAG root adds the HBH header [RFC6553] and send the packet downward to the destination.

7.2. Non-storing

DODAG root is going to add the source route header [RFC6554]

8. Example flow from root to leaf

8.1. Storing

DODAG root adds the HBH header [RFC6553] and send the packet downward to the destination.

8.2. Non-storing

DODAG root is going to add the source route header [RFC6554]

9. IANA Considerations

There are no IANA considerations related to this document.

10. Security Considerations

TBD.

11. Acknowledgements

This work is partially funded by the FP7 Marie Curie Initial Training Network (ITN) METRICS project (grant agreement No. 607728)

12. References

12.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC6550] Winter, T., Thubert, P., 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, March 2012.

[RFC6553] Hui, J. and JP. Vasseur, "The Routing Protocol for Low-Power and Lossy Networks (RPL) Option for Carrying RPL Information in Data-Plane Datagrams", RFC 6553, March 2012.

[RFC6554] Hui, J., Vasseur, JP., Culler, D., and V. Manral, "An IPv6 Routing Header for Source Routes with the Routing Protocol for Low-Power and Lossy Networks (RPL)", RFC 6554, March 2012.

12.2. Informative References

[RFC7102] Vasseur, JP., "Terms Used in Routing for Low-Power and Lossy Networks", RFC 7102, January 2014.

Authors' Addresses

Maria Ines Robles
Ericsson
Hirsalantie 11
Jorvas 02420
Finland

Email: maria.ines.robles@ericsson.com

Michael C. Richardson
Sandelman Software Works
470 Dawson Avenue
Ottawa, ON K1Z 5V7
CA

Email: mcr+ietf@sandelman.ca
URI: http://www.sandelman.ca/

ROLL P. Thubert, Ed.
Internet-Draft J. Pylakutty
Intended status: Standards Track Cisco
Expires: December 31, 2015 June 29, 2015

Root initiated routing state in RPL
draft-thubert-roll-dao-projection-00

Abstract

This document proposes a root-initiated protocol extension to RPL that enables to install a limited amount of downward routes in non-storing mode. This enables loose source routing down the DODAG.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

The Routing Protocol for Low Power and Lossy Networks [RFC6550] (LLN)(RPL) specification defines a generic Distance Vector protocol that is indeed designed for very low energy consumption and adapted to a variety of LLNs. RPL forms Destination Oriented Directed Acyclic Graphs (DODAGs) which root often acts as the Border Router to connect the RPL domain to the Internet. The root is responsible to select the RPL Instance that is used to forward a packet coming from the Internet into the RPL domain and set the related RPL information in the packets.

A classical RPL implementation in a very constrained LLN uses the non-storing mode of operation whereby a RPL node indicates a parent-child relationship to the root, using a Destination Advertisement Object (DAO) that is unicast from the node directly to the root, and the root builds a path to a destination down the DODAG by concatenating this information.

```
              ------+---------
                    |          Internet
                    |
                 +-----+
                 |     | Border Router
                 |     |  (RPL Root)
                 +-----+                      ^     |        |
                    |                         | DAO | ACK    |
              o    o   o    o                 |     |        | Strict
          o o   o  o   o  o  o o   o          |     |        | Source
         o  o o  o o    o   o   o  o  o       |     |        | Route
         o   o    o  o     o  o    o  o  o    |     |        |
        o  o   o  o   o         o   o o       |     v        v
        o          o             o     o
                          LLN
```

Figure 1: RPL non-storing operation

Nodes are not expected to store downward routing state via their children, and the routing operates in strict source routing mode as detailed in An IPv6 Routing Header for Source Routes with RPL [RFC6554]

The non-storing mode of operation (MOP) is largely utilized because networks can get very large and the amount of memory in nodes close to the root may become prohibitive in storing mode. But as a network gets deep, the size of the source routing header that the root must add to all the downward packets may also become an issue as well. In some cases, RPL network form long lines and a limited number of well-targeted routes would enable a loose source routing operation and save packet size, energy, and eventually fragmentation which is highly detrimental to the LLN operation.

This draft proposes an addition whereby the root projects a route through an extended DAO to an arbitrary node down the DODAG, indicating a child or a direct sequence of children via which a certain destination (target) may be reached. The root is expected to use the mechanism optimally and with required parsimony to fit within the device resources, but how the root figures the amount of resources that are available is out of scope.

```
              ------+---------
                    |          Internet
                    |
                 +-----+
                 |     | Border Router
                 |     |  (RPL Root)
                 +-----+                      |     ^        |
                    |                         | DAO | ACK    |
              o    o   o    o                 |     |        | Loose
          o o   o  o   o  o  o o   o          |  ^           | Source
         o  o o  o o    o   o   o  o  o       |  | DAO       | Route
         o   o    o  o     o  o    o  o  o    | ^            |
        o  o   o  o   o         o   o o       v | DAO        v
        o          o             o     o
                         LLN
```

Figure 2: Non-Storing with Projected routes

The 6TiSCH architecture [I-D.ietf-6tisch-architecture] leverages the Deterministic Networking Architecture [I-D.finn-detnet-architecture] as one possible model whereby the device resources and capabilities are exposed to an external entity (a Path Computation Element [PCE]), which installs routing states into the network based on some objective functions that reside in that external entity.

Based on non-specified heuristics of usage, path length, and knowledge of device capacity, this draft enables a RPL root to install and maintain a local and temporary storing mode path within the RPL domain, along a selected set of nodes and for a selected duration, thus advantageously modifying the mode of operation of RPL in non-storing mode.

2. 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 [RFC2119].

The Terminology used in this document is consistent with and incorporates that described in ‘Terminology in Low power And Lossy Networks’ [RFC7102] and [RFC6550].

3. New RPL Control Message Options

Section 6.7 of [RFC6550] specifies Control Message Options (CMO) to be placed in RPL messages such as the DAO message. The RPL Target option indicates a node to be reached and the Transit Information Option specifies a parent that can be used to reach that node.

This specification introduces a new Control Message Option, the Via Information option. One or more Via Information options MUST be preceded by one or more RPL Target options, and the Via options indicate an ordered sequence of hops to reach the target(s), presented in the same order as they would in a routing header.

3.1. Via Information

The Via Information option MAY be present in DAO messages, and its format is as follows:

```
 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 = 0x0A | Option Length | Path Sequence | Path Lifetime |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
+                                                               +
.                                                               .
.                        Child Address                          .
.                                                               .
+                                                               +
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 1: Eliding the RPLInstanceID

Option Type: 0x0A (to be confirmed by IANA)

Option Length: Variable, depending on whether or not Parent Address is present.

Path Sequence: 8-bit unsigned integer. When a RPL Target option is issued by the root of the DODAG (i.e. in a DAO message), that root sets the Path Sequence and increments the Path Sequence each time it issues a RPL Target option with updated information. The indicated sequence deprecates any state for a given Target that was learned from a previous sequence and adds to any state that was learned for that sequence.

Path Lifetime: 8-bit unsigned integer. The length of time in Lifetime Units (obtained from the Configuration option) that the prefix is valid for route determination. The period starts when a new Path Sequence is seen. A value of all one bits (0xFF) represents infinity. A value of all zero bits (0x00) indicates a loss of reachability. A DAO message that contains a Via Information option with a Path Lifetime of 0x00 for a

Target is referred as a No-Path (for that Target) in this document.

Child Address: 8 or 16 bytes. IPv6 Address of the child of the node that is a next hop towards the destination(s) indicated in the target option. If the /64 prefix is the same as that of (all of) the target(s) then the prefix can be elided and the address is expressed as the 8-bytes suffix only.

4. Operation

When a RPL domain operates in non-storing Mode of Operation (NS-MOP), only the root possesses routing information about the whole network. A packet that is generated within the domain first reaches the root, which can then apply a source routing information to reach the destination. Similarly, a packet coming from the outside of the domain for a destination that is expected to be in a RPL domain reaches the root.

In NS-MOP, the root, or some associated centralized computation engine, can thus determine the amount of packets that reach a destination in the RPL domain, and thus the amount of energy and bandwidth that is wasted for transmission, between itself and the destination, as well as the risk of fragmentation, any potential delays because of a paths longer than necessary (shorter paths exist that would not traverse the root).

Additionally, the DAG root knows the whole DAG topology, so when the source of a packet is also in the RPL domain, the root can determine the common parent that would have been used in storing mode, and thus the list of nodes in the path between the common parent and the destination. For instance in the below diagram, if the source is 41 and the destination 52, the common parent is the node 22.

```
              ------+---------
                    |          Internet
                    |
                 +-----+
                 |     | Border Router
                 |     |  (RPL Root)
                 +-----+
                  | \  \____
                 /   \       \
               o 11   o 12     o  13
              /       |       /  \
            o 22      o 23   o 24  o 25
           /  \       | \      \
         o 31   o 32  o   o     o 35
        /      /      |    \    |     \
       o 41   o 42    o     o   o 45   o 46
       |      |       |     |    \     |
       o 51   o 52    o 53  o     o 55 o 56
                          LLN
```

Figure 2: Non-Storing with Projected routes

With this draft, the root can install routing states along a segment that is either itself to the destination, or from one or more common parents for a particular source/destination pair towards that destination (in our example, this would be the segment made of nodes 22, 32, 42).

The draft expects that the root has enough information about the capability for each node to store a number of routes, which can be discovered for instance using a Network Management System (NMS) and/or the RPL routing extensions specified in Routing for Path Calculation in LLNs [RFC6551]. Based on that information, the root computes which segment should be routed and which relevant state should be installed in which nodes. The algorithm is out of scope but it is envisaged that the root could compute the ratio between the optimal path (existing path not traversing the root, and the current path), the application SLA for specific flows that could benefit from shorter paths, the energy wasted in the network, local congestion on various links that would benefit from having flows routed along other paths.

This draft introduces a new mode of operation for loose source routing in the LLN, the Non-Storing with Projected routes MOP. With this new MOP, the root sends a unicast DAO message to the last node of the routing segment that must be installed. The DAO message contains the ordered list of hops along the segment as a list of Via Information options that are preceded by one or more RPL Target options to which they relate. Each Via Information option contains a lifetime for which state is to be maintained.

The root sends the DAO directly to the last node in the segment, which is expected to be able to route to the targets on its own.

The last node in the segment may have another information to reach the target(s), such as a connected route or an already installed projected route. If it does not have such a route then the node should lookup the address on the relevant interfaces. If one of the targets cannot be located, the node MUST answer to the root with a negative DAO-ACK listing the target(s) that could not be located (suggested status 10), and continue the process for those targets that could be located if any.

For the targets that could be located, last node in the segment generates a DAO to its loose predecessor in the segment as indicated in the list of Via Information options.

The node strips the last Via Information option which corresponds to self, and uses it as source address for the DAO to the predecessor. The address of the predecessor to be used as destination for the DAO message is found in the now last Via Information option. The predecessor is expected to have a route to the address used as source, either connected, installed previously as another DAO, or from other means.

The predecessor is expected to have a route to the address used as source and that is his successor. If it does not and cannot locate the successor, the predecessor node MUST answer to the root with a negative DAO-ACK indicating the successor that could not be located. The DAO-ACK contains the list of targets that could not be routed to (suggested status 11).

If the predecessor can route to the successor node, then it installs a route to the targets via the successor. If that route is not connected then a recursive lookup will take place to reach the target(s). From there, the node strips the last Via Information option and either answers to the root with a positive DAO-ACK that contains the list of targets that could be routed to, or propagates the DAO to its own predecessor.

A NULL lifetime in the Via Information option along the segment is used to clean up the state.

In the example below, say that there is a lot of traffic to nodes 55 and 56 and the root decides to reduce the size of routing headers to those destinations. The root can first send a DAO to node 45 indicating target 55 and a Via segment (35, 45), as well as another DAO to node 46 indicating target 56 and a Via segment (35, 46). This will save one entry in the routing header on both sides. The root may then send a DAO to node 35 indicating targets 55 and 56 a Via segment (13, 24, 35) to fully optimize that path.

Alternatively, the root may send a DAO to node 45 indicating target 55 and a Via segment (13, 24, 35, 45) and then a DAO to node 46 indicating target 56 and a Via segment (13, 24, 35, 46), indicating the same DAO Sequence.

5. Security Considerations

This draft uses messages that are already present in [RFC6550] with optional secured versions. The same secured versions may be used with this draft, and whatever security is deployed for a given network also applies to the flows in this draft.

6. IANA Considerations

This document updates the IANA registry for the Mode of Operation (MOP)

4: Non-Storing with Projected routes [this]

This document updates IANA registry for the RPL Control Message Options

0x0A: Via descriptor [this]

7. Acknowledgements

The authors wish to acknowledge JP Vasseur and Patrick Wetterwald for their contributions to the ideas developed here.

8. References

8.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC6550] Winter, T., Thubert, P., 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, March 2012.

[RFC6551] Vasseur, JP., Kim, M., Pister, K., Dejean, N., and D. Barthel, "Routing Metrics Used for Path Calculation in Low-Power and Lossy Networks", RFC 6551, March 2012.

[RFC6554] Hui, J., Vasseur, JP., Culler, D., and V. Manral, "An IPv6 Routing Header for Source Routes with the Routing Protocol for Low-Power and Lossy Networks (RPL)", RFC 6554, March 2012.

[RFC7102] Vasseur, JP., "Terms Used in Routing for Low-Power and Lossy Networks", RFC 7102, January 2014.

8.2. Informative References

[I-D.finn-detnet-architecture]
Finn, N., Thubert, P., and M. Teener, "Deterministic Networking Architecture", draft-finn-detnet-architecture-01 (work in progress), March 2015.

[I-D.ietf-6tisch-architecture]
Thubert, P., "An Architecture for IPv6 over the TSCH mode of IEEE 802.15.4", draft-ietf-6tisch-architecture-08 (work in progress), May 2015.

[PCE]   IETF, "Path Computation Element", <https://datatracker.ietf.org/doc/charter-ietf-pce/>.

Authors' Addresses

Pascal Thubert (editor)
Cisco Systems
Village d'Entreprises Green Side
400, Avenue de Roumanille
Batiment T3
Biot - Sophia Antipolis 06410
FRANCE

Phone: +33 4 97 23 26 34
Email: pthubert@cisco.com

James Pylakutty
Cisco Systems
Cessna Business Park
Kadubeesanahalli
Marathalli ORR
Bangalore, Karnataka 560087
INDIA

Phone: +91 80 4426 4140
Email: mundenma@cisco.com