Workgroup: ROLL
Internet-Draft: draft-ietf-roll-aodv-rpl-15
Published: 30 September 2022
Intended Status: Standards Track
Expires: 3 April 2023
Authors:
C.E. Perkins, Lupin Lodge
S.V.R.Anand, Indian Institute of Science
S. Anamalamudi, SRM University-AP
B. Liu, Huawei Technologies

Supporting Asymmetric Links in Low Power Networks: AODV-RPL

Abstract

Route discovery for symmetric and asymmetric Peer-to-Peer (P2P) traffic flows is a desirable feature in Low power and Lossy Networks (LLNs). For that purpose, this document specifies a reactive P2P route discovery mechanism for both hop-by-hop routes and source routing: Ad Hoc On-demand Distance Vector Routing (AODV) based RPL protocol (AODV-RPL). Paired Instances are used to construct directional paths, for cases where there are asymmetric links between source and target nodes.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

Routing Protocol for Low-Power and Lossy Networks (RPL) [RFC6550] is an IPv6 distance vector routing protocol designed to support multiple traffic flows through a root-based Destination-Oriented Directed Acyclic Graph (DODAG). Typically, a router does not have routing information for most other routers. Consequently, for traffic between routers within the DODAG (i.e., Peer-to-Peer (P2P) traffic) data packets either have to traverse the root in non-storing mode, or traverse a common ancestor in storing mode. Such P2P traffic is thereby likely to traverse longer routes and may suffer severe congestion near the root (for more information see [RFC6687], [RFC6997], [RFC6998], [RFC9010]). The network environment that is considered in this document is assumed to be the same as described in Section 1 of [RFC6550]. Each radio interface/link and the associated address should be treated as an independent intermediate router. Such routers have different links and the rules for the link symmetry apply independently for each of these.

The route discovery process in AODV-RPL is modeled on the analogous peer-to-peer procedure specified in AODV [RFC3561]. The on-demand nature of AODV route discovery is natural for the needs of routing in RPL-based LLNs when routes are needed but aren't yet established. Peer-to-peer routing is desirable to discover shorter routes, and especially when it is desired to avoid directing additional traffic through a root or gateway node of the network. It may happen that some routes need to be established proactively when known beforehand and when AODV-RPL's route discovery process introduces unwanted delay at the time when the application is launched.

AODV terminology has been adapted for use with AODV-RPL messages, namely RREQ for Route Request, and RREP for Route Reply. AODV-RPL currently omits some features compared to AODV -- in particular, flagging Route Errors, "blacklisting" unidirectional links ([RFC3561]), multihoming, and handling unnumbered interfaces.

AODV-RPL reuses and extends the core RPL functionality to support routes with bidirectional asymmetric links. It retains RPL's DODAG formation, RPL Instance and the associated Objective Function (defined in [RFC6551]), trickle timers, and support for storing and non-storing modes. AODV-RPL adds basic messages RREQ and RREP as part of RPL DODAG Information Object (DIO) control message, which go in separate (paired) RPL instances. AODV-RPL does not utilize the Destination Advertisement Object (DAO) control message of RPL. AODV-RPL uses the "P2P Route Discovery Mode of Operation" (MOP == 4) with three new Options for the DIO message, dedicated to discover P2P routes. These P2P routes may differ from routes discoverable by native RPL. Since AODV-RPL uses newly defined Options and a newly allocated multicast group (see Section 9), there is no conflict with P2P-RPL [RFC6997], a previous document using the same MOP. AODV-RPL can be operated whether or not P2P-RPL or native RPL is running otherwise. For many networks AODV-RPL could be a replacement for RPL, since it can find better routes at very moderate cost. Consequently, it seems unlikely that RPL would be needed in a network that is running AODV-RPL, even though it would be possible to run both protocols at the same time.

2. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

AODV-RPL reuses names for messages and data structures, including Rank, DODAG and DODAGID, as defined in RPL [RFC6550].

AODV Ad Hoc On-demand Distance Vector Routing [RFC3561].

ART option AODV-RPL Target option: a target option defined in this document.

Asymmetric Route The route from the OrigNode to the TargNode can traverse different nodes than the route from the TargNode to the OrigNode. An asymmetric route may result from the asymmetry of links, such that only one direction of the series of links satisfies the Objective Function during route discovery.

Bi-directional Asymmetric Link A link that can be used in both directions but with different link characteristics.

DIO DODAG Information Object (as defined in [RFC6550])

DODAG RREQ-Instance (or simply RREQ-Instance) RPL Instance built using the DIO with RREQ option; used for transmission of control messages from OrigNode to TargNode, thus enabling data transmission from TargNode to OrigNode.

DODAG RREP-Instance (or simply RREP-Instance) RPL Instance built using the DIO with RREP option; used for transmission of control messages from TargNode to OrigNode thus enabling data transmission from OrigNode to TargNode.

Downward Direction The direction from the OrigNode to the TargNode.

Downward Route A route in the downward direction.

hop-by-hop route A route for which each router along the routing path stores routing information about the next hop. A hop-by-hop route is created using RPL's "storing mode".

on-demand routing Routing in which a route is established only when needed.

OrigNode The IPv6 router (Originating Node) initiating the AODV-RPL route discovery to obtain a route to TargNode.

Paired DODAGs Two DODAGs for a single route discovery process between OrigNode and TargNode.

P2P Peer-to-Peer -- in other words, not constrained a priori to traverse a common ancestor.

reactive routing Same as "on-demand" routing.

REJOIN_REENABLE
The duration during which a node is prohibited from joining a DODAG with a particular RREQ-InstanceID, after it has left a DODAG with the same RREQ-InstanceID. The default value of REJOIN_REENQBLE is 15 minutes.

RREQ-DIO message A DIO message containing the RREQ option. The RPLInstanceID in RREQ-DIO is assigned locally by the OrigNode. The RREQ-DIO message has a secure variant as noted in [RFC6550].

RREQ-InstanceID The RPLInstanceID for the RREQ-Instance. The RREQ-InstanceID is formed as the ordered pair (Orig_RPLInstanceID, OrigNode-IPaddr), where Orig_RPLInstanceID is the local RPLInstanceID allocated by OrigNode, and OrigNode-IPaddr is an IP address of OrigNode. The RREQ-InstanceID uniquely identifies the RREQ-Instance.

RREP-DIO message A DIO message containing the RREP option. OrigNode pairs the RPLInstanceID in RREP-DIO to the one in the associated RREQ-DIO message (i.e., the RREQ-InstanceID) as described in Section 6.3.2. The RREP-DIO message has a secure variant as noted in [RFC6550].

RREP-InstanceID The RPLInstanceID for the RREP-Instance. The RREP-InstanceID is formed as the ordered pair (Targ_RPLInstanceID, TargNode-IPaddr), where Targ_RPLInstanceID is the local RPLInstanceID allocated by TargNode, and TargNode-IPaddr is an IP address of TargNode. The RREP-InstanceID uniquely identifies the RREP-Instance. The RPLInstanceID in the RREP message along with the Delta value indicates the associated RREQ-InstanceID.

Source routing A mechanism by which the source supplies a vector of addresses towards the destination node along with each data packet [RFC6550].

Symmetric route The upstream and downstream routes traverse the same routers and over the same links.

TargNode The IPv6 router (Target Node) for which OrigNode requires a route and initiates Route Discovery within the LLN network.

Upward Direction The direction from the TargNode to the OrigNode.

Upward Route A route in the upward direction.

3. Overview of AODV-RPL

With AODV-RPL, routes from OrigNode to TargNode within the LLN network are established "on-demand". In other words, the route discovery mechanism in AODV-RPL is invoked reactively when OrigNode has data for delivery to the TargNode but existing routes do not satisfy the application's requirements. AODV-RPL works without requiring the use of RPL or any other routing protocol.

The routes discovered by AODV-RPL are not constrained to traverse a common ancestor. AODV-RPL can enable asymmetric communication paths in networks with bidirectional asymmetric links. For this purpose, AODV-RPL enables discovery of two routes: namely, one from OrigNode to TargNode, and another from TargNode to OrigNode. AODV-RPL also enables discovery of symmetric routes along Paired DODAGs, when symmetric routes are possible (see Section 5).

In AODV-RPL, routes are discovered by first forming a temporary DAG rooted at the OrigNode. Paired DODAGs (Instances) are constructed during route formation between the OrigNode and TargNode. The RREQ-Instance is formed by route control messages from OrigNode to TargNode whereas the RREP-Instance is formed by route control messages from TargNode to OrigNode. The route discovered in the RREQ-Instance is used for transmitting data from TargNode to OrigNode, and the route discovered in RREP-Instance is used for transmitting data from OrigNode to TargNode.

Intermediate routers join the DODAGs based on the Rank [RFC6550] as calculated from the DIO message. Henceforth in this document, "RREQ-DIO message" means the DIO message from OrigNode toward TargNode, containing the RREQ option as specified in Section 4.1. The RREQ-InstanceID is formed as the ordered pair (Orig_RPLInstanceID, OrigNode-IPaddr), where Orig_RPLInstanceID is the local RPLInstanceID allocated by OrigNode, and OrigNode-IPaddr is the IP address of OrigNode. A node receiving the RREQ-DIO can use the RREQ-InstanceID to identify the proper OF whenever that node receives a data packet with Source Address == OrigNode-IPaddr and IPv6 RPL Option having the RPLInstanceID == Orig_RPLInstanceID along with 'D' == 0.

Similarly, "RREP-DIO message" means the DIO message from TargNode toward OrigNode, containing the RREP option as specified in Section 4.2. The RREP-InstanceID is formed as the ordered pair (Targ_RPLInstanceID, TargNode-IPaddr), where Targ_RPLInstanceID is the local RPLInstanceID allocated by TargNode, and TargNode-IPaddr is the IP address of TargNode. A node receiving the RREP-DIO can use the RREP-InstanceID to identify the proper OF whenever that node receives a data packet with Source Address == TargNode-IPaddr and IPv6 RPL Option having the RPLInstanceID == Targ_RPLInstanceID along with 'D' == 0.

4. AODV-RPL DIO Options

4.1. AODV-RPL RREQ Option

OrigNode selects one of its IPv6 addresses and sets it in the DODAGID field of the RREQ-DIO message. The address scope of the selected address must encompass the domain where the route is built (e.g, not link-local). Exactly one RREQ option MUST be present in a RREQ-DIO message, otherwise the message MUST be dropped.

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Option Type  | Option Length |S|H|X| Compr | L |  RankLimit  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Orig SeqNo   |                                               |
+-+-+-+-+-+-+-+-+                                               |
|                                                               |
|                                                               |
|           Address Vector (Optional, Variable Length)          |
|                                                               |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 1: Format for AODV-RPL RREQ Option

OrigNode supplies the following information in the RREQ option:

Option Type TBD2

Option Length The length of the option in octets, excluding the Type and Length fields. Variable due to the presence of the address vector and the number of octets elided according to the Compr value.

S Symmetric bit indicating a symmetric route from the OrigNode to the router transmitting this RREQ-DIO. See Section 5.

H Set to one for a hop-by-hop route. Set to zero for a source route. This flag controls both the downstream route and upstream route.

X Reserved; MUST be initialized to zero and ignored upon reception.

Compr 4-bit unsigned integer. When Compr is nonzero, exactly that number of prefix octets MUST be elided from each address before storing it in the Address Vector. The octets elided are shared with the IPv6 address in the DODAGID. This field is only used in source routing mode (H=0). In hop-by-hop mode (H=1), this field MUST be set to zero and ignored upon reception.

L 2-bit unsigned integer determining the time duration that a node is able to belong to the RREQ-Instance (a temporary DAG including the OrigNode and the TargNode). Once the time is reached, a node MUST leave the RREQ-Instance and stop sending or receiving any more DIOs for the RREQ-Instance. This naturally depends on the node's ability to keep track of time. Once a node leaves an RREQ-Instance, it MUST NOT rejoin the same RREQ-Instance for at least the time interval specified by the configuration variable REJOIN_REENABLE. L is independent from the route lifetime, which is defined in the DODAG configuration option.

- *0x00: No time limit imposed.
- *0x01: 16 seconds
- *0x02: 64 seconds
- *0x03: 256 seconds

RankLimit This field indicates the upper limit on the integer portion of the Rank (calculated using the DAGRank() macro defined in [RFC6550]). A value of 0 in this field indicates the limit is infinity.

Orig SeqNo Sequence Number of OrigNode. See Section 6.1.

Address Vector A vector of IPv6 addresses representing the route that the RREQ-DIO has passed. It is only present when the H bit is set to 0. The prefix of each address is elided according to the Compr field.

TargNode can join the RREQ instance at a Rank whose integer portion is less than or equal to the RankLimit. Any other node MUST NOT join a RREQ instance if its own Rank would be equal to or higher than RankLimit. A router MUST discard a received RREQ if the integer part of the advertised Rank equals or exceeds the RankLimit.

4.2. AODV-RPL RREP Option

TargNode sets one of its IPv6 addresses in the DODAGID field of the RREP-DIO message. The address scope of the selected address must encompass the domain where the route is built (e.g, not link-local). Exactly one RREP option MUST be present in a RREP-DIO message, otherwise the message MUST be dropped. TargNode supplies the following information in the RREP option:

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Option Type  | Option Length |G|H|X| Compr | L |  RankLimit  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   Delta   |X X|                                               |
+-+-+-+-+-+-+-+-+                                               |
|                                                               |
|                                                               |
|           Address Vector (Optional, Variable Length)          |
.                                                               .
.                                                               .
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 2: Format for AODV-RPL RREP option

Option Type TBD3

Option Length The length of the option in octets, excluding the Type and Length fields. Variable due to the presence of the address vector and the number of octets elided according to the Compr value.

G Gratuitous RREP (see Section 7).

H The H bit in the RREP option MUST be set to be the same as the H bit in RREQ option. It requests either source routing (H=0) or hop-by-hop (H=1) for the downstream route.

X Reserved; MUST be initialized to zero and ignored upon reception.

Compr 4-bit unsigned integer. Same definition as in RREQ option.

L
2-bit unsigned integer defined as in RREQ option. The lifetime of the RREP-Instance MUST be no greater than the lifetime of the RREQ-Instance to which it is paired.

RankLimit Similarly to RankLimit in the RREQ message, this field indicates the upper limit on the integer portion of the Rank. A value of 0 in this field indicates the limit is infinity.

Delta 6-bit unsigned integer. This field is used to recover the RREQ-InstanceID (see Section 6.3.3); 0 indicates that the RREQ-InstanceID has the same value as the RPLInstanceID of the RREP message.

X X Reserved; MUST be initialized to zero and ignored upon reception.

Address Vector Only present when the H bit is set to 0. For an asymmetric route, the Address Vector represents the IPv6 addresses of the path through the network the RREP-DIO has passed. For a symmetric route, it is the Address Vector when the RREQ-DIO arrives at the TargNode, unchanged during the transmission to the OrigNode.

4.3. AODV-RPL Target Option

The AODV-RPL Target (ART) Option is based on the Target Option in core RPL [RFC6550]. The Flags field is replaced by the Destination Sequence Number of the TargNode and the Prefix Length field is reduced to 7 bits so that the value is limited to be no greater than 127.

A RREQ-DIO message MUST carry at least one ART Option. A RREP-DIO message MUST carry exactly one ART Option. Otherwise, the message MUST be dropped.

OrigNode can include multiple TargNode addresses via multiple AODV-RPL Target Options in the RREQ-DIO, for routes that share the same requirement on metrics. This reduces the cost to building only one DODAG.

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Option Type  | Option Length |  Dest SeqNo   |X|Prefix Length|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
+                                                               |
|            Target Prefix / Address (Variable Length)          |
.                                                               .
.                                                               .
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 3: ART Option format for AODV-RPL

Option Type
TBD4

Option Length Length of the option in octets excluding the Type and Length fields.

Dest SeqNo In RREQ-DIO, if nonzero, it is the Sequence Number for the last route that OrigNode stored to the TargNode for which a route is desired. In RREP-DIO, it is the destination sequence number associated to the route. Zero is used if there is no known information about the sequence number of TargNode, and not used otherwise.

X A one-bit reserved field. This field MUST be initialized to zero by the sender and MUST be ignored by the receiver.

Prefix Length 7-bit unsigned integer. Number of valid leading bits in the IPv6 Prefix. If Prefix Length is 0, then the value in the Target Prefix / Address field represents an IPv6 address, not a prefix.

Target Prefix / Address (variable-length field) An IPv6 destination address or prefix. The Prefix Length field contains the number of valid leading bits in the prefix. The Target Prefix / Address field contains the least number of octets that can represent all of the bits of the Prefix, in other words Ceil(Prefix Length/8) octets. The initial bits in the Target Prefix / Address field preceding the prefix length (if any) MUST be set to zero on transmission and MUST be ignored on receipt. If Prefix Length is zero, the Address field is 128 bits for IPv6 addresses.

5. Symmetric and Asymmetric Routes

Links are considered symmetric until indication to the contrary is received. In Figure 4 and Figure 5, BR is the Border Router, O is the OrigNode, each R is an intermediate router, and T is the TargNode. In this example, the use of BR is only for illustrative purposes; AODV does not depend on the use of border routers for its operation. If the RREQ-DIO arrives over an interface that is known to be symmetric, and the S bit is set to 1, then it remains as 1, as illustrated in Figure 4. If an intermediate router sends out RREQ-DIO with the S bit set to 1, then each link en route from the OrigNode O to this router has met the requirements of route discovery, and the route can be used symmetrically.

```
                                BR
                            /----+----\
                          /      |      \
                        /        |         \
                       R         R           R
                    _/  \        |          /  \
                   /     \       |         /     \
                  /       \      |        /        \
                R -------- R --- R ----- R -------- R
              /  \   <--S=1-->  / \    <--S=1-->   /  \
       <--S=1-->  \            /   \             /   <--S=1-->
         /         \          /     \          /         \
       O ---------- R ------ R------ R ----- R ----------- T
      / \                   / \             / \           / \
     /   \                 /   \           /   \         /   \
    /     \               /     \         /     \       /     \
   R ----- R ----------- R ----- R ----- R ----- R ---- R----- R

     >---- RREQ-Instance (Control: O-->T;  Data: T-->O) ------->
     <---- RREP-Instance (Control: T-->O;  Data: O-->T) -------<
```

Figure 4: AODV-RPL with Symmetric Instances

Upon receiving a RREQ-DIO with the S bit set to 1, a node determines whether this link can be used symmetrically, i.e., both directions meet the requirements of data transmission. If the RREQ-DIO arrives over an interface that is not known to be symmetric, or is known to be asymmetric, the S bit is set to 0. If the S bit arrives already set to be '0', it is set to be '0' when the RREQ-DIO is propagated (Figure 5). For an asymmetric route, there is at least one hop which doesn't satisfy the Objective Function. Based on the S bit received in RREQ-DIO, TargNode T determines whether or not the route is symmetric before transmitting the RREP-DIO message upstream towards the OrigNode O.

It is beyond the scope of this document to specify the criteria used when determining whether or not each link is symmetric. As an example, intermediate routers can use local information (e.g., bit rate, bandwidth, number of cells used in 6tisch [RFC9030]), a priori knowledge (e.g., link quality according to previous communication) or use averaging techniques as appropriate to the application. Other link metric information can be acquired before AODV-RPL operation, by executing evaluation procedures; for instance test traffic can be generated between nodes of the deployed network. During AODV-RPL operation, OAM techniques for evaluating link state (see [RFC7548], [RFC7276], [co-ioam]) MAY be used (at regular intervals appropriate for the LLN). The evaluation procedures are out of scope for AODV-RPL. For further information on this topic, see [Link_Asymmetry], [low-power-wireless], and [empirical-study].

Appendix A describes an example method using the upstream Expected Number of Transmissions (ETX) and downstream Received Signal Strength Indicator (RSSI) to estimate whether the link is symmetric in terms of link quality using an averaging technique.

```
                                    BR
                                /----+----\
                              /      |      \
                            /        |        \
                          R          R          R
                        / \          |        /   \
                      /     \        |       /      \
                    /         \      |      /         \
                   R --------- R --- R ---- R --------- R
                 /  \   --S=1-->   / \    --S=0-->   /   \
           --S=1-->   \           /    \            /   --S=0-->
            /          \        /       \         /         \
          O ---------- R ------ R------ R ----- R ----------- T
         / \                   / \             / \           / \
        /  <--S=0--           /   \           /   \         / <--S=0--
       /     \               /     \         /     \       /     \
      R ----- R ----------- R ----- R ----- R ----- R ---- R----- R
                  <--S=0--   <--S=0-- <--S=0-- <--S=0--    <--S=0--

      >---- RREQ-Instance (Control: O-->T;  Data: T-->O) ------->
      <---- RREP-Instance (Control: T-->O;  Data: O-->T) -------<
```

Figure 5: AODV-RPL with Asymmetric Paired Instances

As illustrated in Figure 5, an intermediate router determines the S bit value that the RREQ-DIO should carry using link asymmetry detection methods as discussed earlier in this section. In many cases the intermediate router has already made the link asymmetry decision by the time RREQ-DIO arrives.

6. AODV-RPL Operation

6.1. Route Request Generation

The route discovery process is initiated when an application at the OrigNode has data to be transmitted to the TargNode, but does not have a route that satisfies the Objective Function for the target of the application's data. In this case, the OrigNode builds a local RPLInstance and a DODAG rooted at itself. Then it transmits a DIO message containing exactly one RREQ option (see Section 4.1) to multicast group all-AODV-RPL-nodes. The RREQ-DIO MUST contain at least one ART Option (see Section 4.3), which indicates the TargNode. The S bit in RREQ-DIO sent out by the OrigNode is set to 1.

Each node maintains a sequence number; the operation is specified in section 7.2 of [RFC6550]. When the OrigNode initiates a route discovery process, it MUST increase its own sequence number to avoid conflicts with previously established routes. The sequence number is carried in the Orig SeqNo field of the RREQ option.

The Target Prefix / Address in the ART Option can be a unicast IPv6 address or a prefix. The OrigNode can initiate the route discovery process for multiple targets simultaneously by including multiple ART Options. Within a RREQ-DIO the Objective Function for the routes to different TargNodes MUST be the same.

OrigNode can maintain different RPLInstances to discover routes with different requirements to the same targets. Using the RPLInstanceID pairing mechanism (see Section 6.3.3), route replies (RREP-DIOs) for different RPLInstances can be generated.

The transmission of RREQ-DIO obeys the Trickle timer [RFC6206]. If the duration specified by the L field has elapsed, the OrigNode MUST leave the DODAG and stop sending RREQ-DIOs in the related RPLInstance. OrigNode needs to set L field such that the DODAG will not prematurely timeout during data transfer with the TargNode. For setting this value, it has to consider factors such as Trickle timer, TargNode hop distance, network size, link behavior, expected data usage time, and so on.

6.2. Receiving and Forwarding RREQ messages

6.2.1. Step 1: RREQ reception and evaluation

When a router X receives a RREQ message over a link from a neighbor Y, X first determines whether or not the RREQ is valid. If so, X then determines whether or not it has sufficient resources available to maintain the state needed to process an eventual RREP if the RREP were to be received. If not, then X MUST drop the packet and discontinue processing of the RREQ. Otherwise, X next determines whether the RREQ advertises a usable route to OrigNode, by checking whether the link to Y can be used to tramsmit packets to OrigNode.

When H=0 in the incoming RREQ, the router MUST drop the RREQ-DIO if one of its addresses is present in the Address Vector. When H=1 in the incoming RREQ, the router MUST drop the RREQ message if Orig SeqNo field of the RREQ is older than the SeqNo value that X has stored for a route to OrigNode. Otherwise, the router determines whether to propagate the RREQ-DIO. It does this by determining whether or not a route to OrigNode using the upstream direction of the incoming link satisfies the Objective Function (OF). In order to evaluate the OF, the router first determines the maximum useful rank (MaxUsefulRank). If the router has previously joined the RREQ-Instance associated with the RREQ-DIO, then MaxUsefulRank is set to be the Rank value that was stored when the router processed the best previous RREQ for the DODAG with the given RREQ-Instance. Otherwise, MaxUsefulRank is set to be RankLimit. If OF cannot be satisfied (i.e., the Rank evaluates to a value greater than MaxUsefulRank) the RREQ-DIO MUST be dropped, and the following steps are not processed. Otherwise, the router MUST join the RREQ-Instance and prepare to propagate the RREQ-DIO, as follows. The upstream neighbor router that transmitted the received RREQ-DIO is selected as the preferred parent.

6.2.2. Step 2: TargNode and Intermediate Router determination

After determining that a received RREQ provides a usable route to OrigNode, a router determines whether it is a TargNode, or a possible intermediate router between OrigNode and a TargNode, or both. The router is a TargNode if it finds one of its own addresses in a Target Option in the RREQ. After possibly propagating the RREQ according to the procedures in Steps 3, 4, and 5, the TargNode generates a RREP as specified in Section 6.3. If S=0, the determination of TargNode status and determination of a usable route to OrigNode is the same.

If the OrigNode tries to reach multiple TargNodes in a single RREQ-Instance, one of the TargNodes can be an intermediate router to other TargNodes. In this case, before transmitting the RREQ-DIO to multicast group all-AODV-RPL-nodes, a TargNode MUST delete the Target Option encapsulating its own address, so that downstream routers with higher Rank values do not try to create a route to this TargNode.

An intermediate router could receive several RREQ-DIOs from routers with lower Rank values in the same RREQ-Instance with different lists of Target Options. For the purposes of determining the intersection with previous incoming RREQ-DIOs, the intermediate router maintains a record of the targets that have been requested for a given RREQ-Instance. An incoming RREQ-DIO message having multiple ART Options coming from a router with higher Rank than the Rank of the stored targets is ignored. When transmitting the RREQ-DIO, the intersection of all received lists MUST be included if it is nonempty after TargNode has deleted the Target Option encapsulating its own address. If the intersection is empty, it means that all the targets have been reached, and the router MUST NOT transmit any RREQ-DIO. Otherwise it proceeds to Section 6.2.3.

For example, suppose two RREQ-DIOs are received with the same RPLInstance and OrigNode. Suppose further that the first RREQ has (T1, T2) as the targets, and the second one has (T2, T4) as targets. Then only T2 needs to be included in the generated RREQ-DIO.

6.2.3. Step 3: Intermediate Router RREQ processing

The intermediate router establishes itself as a viable node for a route to OrigNode as follows. If the H bit is set to 1, for a hop-by-hop route, then the router MUST build or update its upward route entry towards OrigNode, which includes at least the following items: Source Address, RPLInstanceID, Destination Address, Next Hop, Lifetime, and Sequence Number. The Destination Address and the RPLInstanceID respectively can be learned from the DODAGID and the RPLInstanceID of the RREQ-DIO. The Source Address is the address used by the router to send data to the Next Hop, i.e., the preferred parent. The lifetime is set according to DODAG configuration (not the L field) and can be extended when the route is actually used. The sequence number represents the freshness of the route entry; it is copied from the Orig SeqNo field of the RREQ option. A route entry with the same source and destination address, same RPLInstanceID, but stale sequence number, MUST be deleted.

6.2.4. Step 4: Symmetric Route Processing at an Intermediate Router

If the S bit of the incoming RREQ-DIO is 0, then the route cannot be symmetric, and the S bit of the RREQ-DIO to be transmitted is set to 0. Otherwise, the router MUST determine whether the downward (i.e., towards the TargNode) direction of the incoming link satisfies the OF. If so, the S bit of the RREQ-DIO to be transmitted is set to 1. Otherwise the S bit of the RREQ-DIO to be transmitted is set to 0.

When a router joins the RREQ-Instance, it also associates within its data structure for the RREQ-Instance the information about whether or not the RREQ-DIO to be transmitted has the S-bit set to 1. This information associated to RREQ-Instance is known as the S-bit of the RREQ-Instance. It will be used later during the RREP-DIO message processing Section 6.3.2.

Suppose a router has joined the RREQ-Instance, and H=0, and the S-bit of the RREQ-Instance is set to 1. In this case, the router MAY optionally associate to the RREQ-Instance, the Address Vector of the symmetric route back to OrigNode. This is useful if the router later receives an RREP-DIO that is paired with the RREQ-Instance. If the router does NOT associate the Address Vector, then it has to rely on multicast for the RREP. This can impose a substantial performance penalty.

6.2.5. Step 5: RREQ propagation at an Intermediate Router

If the router is an intermediate router, then it transmits the RREQ-DIO to the multicast group all-AODV-RPL-nodes; if the H bit is set to 0, the intermediate router MUST append the address of its interface receiving the RREQ-DIO into the address vector. If, in addition, the address of the router's transmitting the RREQ-DIO is not the same as the address of the interface receiving the RREQ-DIO, the router MUST also append the transmitting interface address into the address vector.

6.2.6. Step 6: RREQ reception at TargNode

If the router is a TargNode and was already associated with the RREQ-Instance, it takes no further action and does not send an RREP-DIO. If TargNode is not already associated with the RREQ-Instance, it prepares and transmits a RREP-DIO, possibly after waiting for RREP_WAIT_TIME, as detailed in (Section 6.3).

6.3. Generating Route Reply (RREP) at TargNode

When a TargNode receives a RREQ message over a link from a neighbor Y, TargNode first follows the procedures in Section 6.2. If the link to Y can be used to tramsmit packets to OrigNode, TargNode generates a RREP according to the steps below. Otherwise TargNode drops the RREQ and does not generate a RREP.

If the L field is not 0, the TargNode MAY delay transmitting the RREP-DIO for duration RREP_WAIT_TIME to await a route with a lower Rank. The value of RREP_WAIT_TIME is set by default to 1/4 of the duration determined by the L field. For L == 0, RREP_WAIT_TIME is set by default to 0. Depending upon the application, RREP_WAIT_TIME may be set to other values. Smaller values enable quicker formation for the P2P route. Larger values enable formation of P2P routes with better Rank values.

The address of the OrigNode MUST be encapsulated in the ART Option and included in this RREP-DIO message along with the SeqNo of TargNode.

6.3.1. RREP-DIO for Symmetric route

If the RREQ-Instance corresponding to the RREQ-DIO that arrived at TargNode has the S bit set to 1, there is a symmetric route both of whose directions satisfy the Objective Function. Other RREQ-DIOs might later provide better upward routes. The method of selection between a qualified symmetric route and an asymmetric route that might have better performance is implementation-specific and out of scope.

For a symmetric route, the RREP-DIO message is unicast to the next hop according to the Address Vector (H=0) or the route entry (H=1); the DODAG in RREP-Instance does not need to be built. The RPLInstanceID in the RREP-Instance is paired as defined in Section 6.3.3. In case the H bit is set to 0, the address vector from the RREQ-DIO MUST be included in the RREP-DIO.

6.3.2. RREP-DIO for Asymmetric Route

When a RREQ-DIO arrives at a TargNode with the S bit set to 0, the TargNode MUST build a DODAG in the RREP-Instance corresponding to the RREQ-DIO rooted at itself, in order to provide OrigNode with a downstream route to the TargNode. The RREP-DIO message is transmitted to multicast group all-AODV-RPL-nodes.

6.3.3. RPLInstanceID Pairing

Since the RPLInstanceID is assigned locally (i.e., there is no coordination between routers in the assignment of RPLInstanceID), the tuple (OrigNode, TargNode, RPLInstanceID) is needed to uniquely identify a discovered route. It is possible that multiple route discoveries with dissimilar Objective Functions are initiated simultaneously. Thus between the same pair of OrigNode and TargNode, there can be multiple AODV-RPL route discovery instances. So that OrigNode and Targnode can avoid any mismatch, they MUST pair the RREQ-Instance and the RREP-Instance in the same route discovery by using the RPLInstanceID.

When preparing the RREP-DIO, a TargNode could find the RPLInstanceID candidate for the RREP-Instance is already occupied by another RPL Instance from an earlier route discovery operation which is still active. This unlikely case might happen if two distinct OrigNodes need routes to the same TargNode, and they happen to use the same RPLInstanceID for RREQ-Instance. In such cases, the RPLInstanceID of an already active RREP-Instance MUST NOT be used again for assigning RPLInstanceID for the later RREP-Instance. If the same RPLInstanceID were re-used for two distinct DODAGs originated with the same DODAGID (TargNode address), intermediate routers could not distinguish between these DODAGs (and their associated Objective Functions). Instead, the RPLInstanceID MUST be replaced by another value so that the two RREP-instances can be distinguished. In the RREP-DIO option, the Delta field of the RREP-DIO message (Figure 2) indicates the increment to be applied to the pre-existing RPLInstanceID to obtain the value of the RPLInstanceID that is used in the RREP-DIO message. When the new RPLInstanceID after incrementation exceeds 255, it rolls over starting at 0. For example, if the RREQ-InstanceID is 252, and incremented by 6, the new RPLInstanceID will be 2. Related operations can be found in

Section 6.4. RPLInstanceID collisions do not occur across RREQ-DIOs; the DODAGID equals the OrigNode address and is sufficient to disambiguate between DODAGs.

6.4. Receiving and Forwarding Route Reply

Upon receiving a RREP-DIO, a router which already belongs to the RREP-Instance SHOULD drop the RREP-DIO. Otherwise the router performs the steps in the following subsections.

6.4.1. Step 1: Receiving and Evaluation

If the Objective Function is not satisfied, the router MUST NOT join the DODAG; the router MUST discard the RREP-DIO, and does not execute the remaining steps in this section. An Intermediate Router MUST discard a RREP if one of its addresses is present in the Address Vector, and does not execute the remaining steps in this section.

If the S bit of the associated RREQ-Instance is set to 1, the router MUST proceed to Section 6.4.2.

If the S-bit of the RREQ-Instance is set to 0, the router MUST determine whether the downward direction of the link (towards the TargNode) over which the RREP-DIO is received satisfies the Objective Function, and the router's Rank would not exceed the RankLimit. If so, the router joins the DODAG of the RREP-Instance. The router that transmitted the received RREP-DIO is selected as the preferred parent. Afterwards, other RREP-DIO messages can be received; AODV-RPL does not specify any action to be taken in such cases.

6.4.2. Step 2: OrigNode or Intermediate Router

The router updates its stored value of the TargNode's sequence number according to the value provided in the ART option. The router next checks if one of its addresses is included in the ART Option. If so, this router is the OrigNode of the route discovery. Otherwise, it is an intermediate router.

6.4.3. Step 3: Build Route to TargNode

If the H bit is set to 1, then the router (OrigNode or intermediate) MUST build a downward route entry towards TargNode which includes at least the following items: OrigNode Address, RPLInstanceID, TargNode Address as destination, Next Hop, Lifetime and Sequence Number. For a symmetric route, the Next Hop in the route entry is the router from which the RREP-DIO is received. For an asymmetric route, the Next Hop is the preferred parent in the DODAG of RREP-Instance. The RPLInstanceID in the route entry MUST be the RREQ-InstanceID (i.e., after subtracting the Delta field value from the value of the RPLInstanceID). The source address is learned from the ART Option, and the destination address is learned from the DODAGID. The lifetime is set according to DODAG configuration (i.e., not the L field) and can be extended when the route is actually used. The sequence number represents the freshness of the route entry, and is copied from the Dest SeqNo field of the ART option of the RREP-DIO. A route entry with same source and destination address, same

RPLInstanceID, but stale sequence number (i.e., incoming sequence number is less than the currently stored sequence number of the route entry), MUST be deleted.

6.4.4. Step 4: RREP Propagation

If the receiver is the OrigNode, it can start transmitting the application data to TargNode along the path as provided in RREP-Instance, and processing for the RREP-DIO is complete. Otherwise, the RREP will be propagated towards OrigNode. If H=0, the intermediate router MUST include the address of the interface receiving the RREP-DIO into the address vector. If H=1, according to the last step the intermediate router has set up a route entry for TargNode. If the intermediate router has a route to OrigNode, it uses that route to unicast the RREP-DIO to OrigNode. Otherwise, in case of a symmetric route, the RREP-DIO message is unicast to the Next Hop according to the address vector in the RREP-DIO (H=0) or the local route entry (H=1). Otherwise, in case of an asymmetric route, the intermediate router transmits the RREP-DIO to multicast group all-AODV-RPL-nodes. The RPLInstanceID in the transmitted RREP-DIO is the same as the value in the received RREP-DIO.

7. Gratuitous RREP

In some cases, an Intermediate router that receives a RREQ-DIO message MAY unicast a "Gratuitous" RREP-DIO message back to OrigNode before continuing the transmission of the RREQ-DIO towards TargNode. The Gratuitous RREP allows the OrigNode to start transmitting data to TargNode sooner. The G bit of the RREP option is provided to distinguish the Gratuitous RREP-DIO (G=1) sent by the Intermediate router from the RREP-DIO sent by TargNode (G=0).

The gratuitous RREP-DIO MAY be sent out when the Intermediate router receives a RREQ-DIO for a TargNode, and the router has a pair of downward and upward routes to the TargNode which also satisfy the Objective Function and for which the destination sequence number is at least as large as the sequence number in the RREQ-DIO message. After unicasting the Gratuitous RREP to the OrigNode, the Intermediate router then unicasts the RREQ towards TargNode, so that TargNode will have the advertised route towards OrigNode along with the RREQ-InstanceID for the RREQ-Instance.

In case of source routing, the intermediate router MUST include the address vector between the OrigNode and itself in the Gratuitous RREP. It also includes the address vector in the unicast RREQ-DIO towards TargNode. Upon reception of the unicast RREQ-DIO, the TargNode will have a route address vector from itself to the OrigNode. Then the router MUST include the address vector from the TargNode to the router itself in the gratuitous RREP-DIO to be transmitted.

For establishing hop-by-hop routes, the intermediate router MUST unicast the received RREQ-DIO to the Next Hop on the route. The Next Hop router along the route MUST build new route entries with the related RPLInstanceID and DODAGID in the downward direction. This process repeats at each node until the RREQ-DIO arrives at the TargNode. Then the TargNode and each router along the path towards

OrigNode MUST unicast the RREP-DIO hop-by-hop towards OrigNode as specified in Section 6.3.

8. Operation of Trickle Timer

RREQ-Instance/RREP-Instance multicast uses trickle timer operations [RFC6206] to control RREQ-DIO and RREP-DIO transmissions. The Trickle control of these DIO transmissions follows the procedures described in the Section 8.3 of [RFC6550] entitled "DIO Transmission". If the route is symmetric, the RREP DIO does not need the Trickle timer mechanism.

9. IANA Considerations

Note to RFC editor:

The sentence "The parenthesized numbers are only suggestions." is to be removed prior publication.

A Subregistry in this section refers to a named sub-registry of the "Routing Protocol for Low Power and Lossy Networks (RPL)" registry.

AODV-RPL uses the "P2P Route Discovery Mode of Operation" (MOP == 4) with new Options as specified in this document. Please cite AODV-RPL and this document as one of the protocols using MOP 4.

IANA is asked to assign three new AODV-RPL options "RREQ", "RREP" and "ART", as described in Figure 6 from the "RPL Control Message Options" Subregistry. The parenthesized numbers are only suggestions.

Value	Meaning	Reference
TBD2 (0x0B)	RREQ Option	This document
TBD3 (0x0C)	REP Option	This document
TBD4 (0x0D)	ART Option	This document

Figure 6: AODV-RPL Options

IANA is requested to allocate a new permanent multicast address with link-local scope called all-AODV-RPL-nodes for nodes implementing this specification.

10. Security Considerations

The security considerations for the operation of AODV-RPL are similar to those for the operation of RPL (as described in Section 19 of the RPL specification [RFC6550]). Sections 6.1 and 10 of [RFC6550] describe RPL's optional security framework, which AODV-RPL relies on to provide data confidentiality, authentication, replay protection, and delay protection services. Additional analysis for the security threats to RPL can be found in [RFC7416].

A router can join a temporary DAG created for a secure AODV-RPL route discovery only if it can support the security configuration in use (see Section 6.1 of [RFC6550]), which also specifies the key in use. It does not matter whether the key is preinstalled or dynamically acquired. The router must have the key in use before it can join the DAG being created for secure route discovery.

If a rogue router knows the key for the security configuration in use, it can join the secure AODV-RPL route discovery and cause various types of damage. Such a rogue router could advertise false information in its DIOs in order to include itself in the discovered route(s). It could generate bogus RREQ-DIO, and RREP-DIO messages carrying bad routes or maliciously modify genuine RREP-DIO messages it receives. A rogue router acting as the OrigNode could launch denial-of-service attacks against the LLN deployment by initiating fake AODV-RPL route discoveries. When rogue routers might be present, RPL's preinstalled mode of operation, where the key to use for route discovery is preinstalled, SHOULD be used.

When a RREQ-DIO message uses the source routing option by setting the H bit to 0, a rogue router may populate the Address Vector field with a set of addresses that may result in the RREP-DIO traveling in a routing loop.

If a rogue router is able to forge a gratuitous RREP, it could mount denial-of-service attacks.

11. Acknowledgements

The authors thank Pascal Thubert, Rahul Jadhav, and Lijo Thomas for their support and valuable inputs. The authors specially thank Lavanya H.M for implementing AODV-RPl in Contiki and conducting extensive simulation studies.

The authors would like to acknowledge the review, feedback and comments from the following people, in alphabetical order: Roman Danyliw, Lars Eggert, Benjamin Kaduk, Tero Kivinen, Erik Kline, Murray Kucherawy, Warren Kumari, Francesca Palombini, Alvaro Retana, Ines Robles, John Scudder, Meral Shirazipour, Peter Van der Stok, Eric Vyncke, and Robert Wilton.

12. References

12.1. 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, <https://www.rfc-editor.org/info/rfc2119>.

[RFC6206] Levis, P., Clausen, T., Hui, J., Gnawali, O., and J. Ko, "The Trickle Algorithm", RFC 6206, DOI 10.17487/RFC6206, March 2011, <https://www.rfc-editor.org/info/rfc6206>.

[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>.

[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, <https://www.rfc-editor.org/info/rfc6551>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

12.2. Informative References

[co-ioam] Rashmi Ballamajalu, Anand, S.V.R., and Malati Hegde, "Co-iOAM: In-situ Telemetry Metadata Transport for Resource Constrained Networks within IETF Standards Framework", 2018 10th International Conference on Communication Systems & Networks (COMSNETS) pp.573-576, January 2018.

[contiki] Contiki contributors, "The Contiki Open Source OS for the Internet of Things (Contiki Version 2.7)", November 2013, <https://github.com/contiki-os/contiki>.

[Contiki-ng] Contiki-NG contributors, "Contiki-NG: The OS for Next Generation IoT Devices (Contiki-NG Version 4.6)", December 2020, <https://github.com/contiki-ng/contiki-ng>.

[cooja] Contiki/Cooja contributors, "Cooja Simulator for Wireless Sensor Networks (Contiki/Cooja Version 2.7)", November 2013, <https://github.com/contiki-os/contiki/tree/master/tools/cooja>.

[empirical-study] Prasant Misra, Nadeem Ahmed, and Sanjay Jha, "An empirical study of asymmetry in low-power wireless links", IEEE Communications Magazine (Volume: 50, Issue: 7), July 2012.

[Link_Asymmetry] Lifeng Sang, Anish Arora, and Hongwei Zhang, "On Link Asymmetry and One-way Estimation in Wireless Sensor Networks", ACM Transactions on Sensor Networks, Volume 6 Issue 2 pp.1-25, February 2010, <https://doi.org/10.1145/1689239.1689242>.

[low-power-wireless] Kannan Srinivasan, Prabal Dutta, Arsalan Tavakoli, and Philip Levis, "An empirical study of low-power wireless", ACM Transactions on Sensor Networks (Volume 6 Issue 2 pp.1-49), February 2010, <https://doi.org/10.1145/1689239.1689246>.

[RFC3561] Perkins, C., Belding-Royer, E., and S. Das, "Ad hoc On-Demand Distance Vector (AODV) Routing", RFC 3561, DOI 10.17487/RFC3561, July 2003, <https://www.rfc-editor.org/info/rfc3561>.

[RFC6687] Tripathi, J., Ed., de Oliveira, J., Ed., and JP. Vasseur, Ed., "Performance Evaluation of the Routing Protocol for Low-Power and Lossy Networks (RPL)", RFC 6687, DOI 10.17487/RFC6687, October 2012, <https://www.rfc-editor.org/info/rfc6687>.

[RFC6997] Goyal, M., Ed., Baccelli, E., Philipp, M., Brandt, A., and J. Martocci, "Reactive Discovery of Point-to-Point Routes in Low-Power and Lossy Networks", RFC 6997, DOI 10.17487/RFC6997, August 2013, <https://www.rfc-editor.org/info/rfc6997>.

[RFC6998] Goyal, M., Ed., Baccelli, E., Brandt, A., and J. Martocci, "A Mechanism to Measure the Routing Metrics along a Point-to-Point Route in a Low-Power and Lossy Network", RFC 6998, DOI 10.17487/RFC6998, August 2013, <https://www.rfc-editor.org/info/rfc6998>.

[RFC7276] Mizrahi, T., Sprecher, N., Bellagamba, E., and Y. Weingarten, "An Overview of Operations, Administration, and Maintenance (OAM) Tools", RFC 7276, DOI 10.17487/RFC7276, June 2014, <https://www.rfc-editor.org/info/rfc7276>.

[RFC7416] Tsao, T., Alexander, R., Dohler, M., Daza, V., Lozano, A., and M. Richardson, Ed., "A Security Threat Analysis for the Routing Protocol for Low-Power and Lossy Networks (RPLs)", RFC 7416, DOI 10.17487/RFC7416, January 2015, <https://www.rfc-editor.org/info/rfc7416>.

[RFC7548] Ersue, M., Ed., Romascanu, D., Schoenwaelder, J., and A. Sehgal, "Management of Networks with Constrained Devices: Use Cases", RFC 7548, DOI 10.17487/RFC7548, May 2015, <https://www.rfc-editor.org/info/rfc7548>.

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

[RFC9030] Thubert, P., Ed., "An Architecture for IPv6 over the Time-Slotted Channel Hopping Mode of IEEE 802.15.4 (6TiSCH)", RFC 9030, DOI 10.17487/RFC9030, May 2021, <https://www.rfc-editor.org/info/rfc9030>.

Appendix A. Example: Using ETX/RSSI Values to determine value of S bit

The combination of Received Signal Strength Indication(downstream) (RSSI) and Expected Number of Transmissions(upstream) (ETX) has been tested to determine whether a link is symmetric or asymmetric at intermediate routers. We present two methods to obtain an ETX value from RSSI measurement.

Method 1: In the first method, we constructed a table measuring RSSI vs ETX using the Cooja simulation [cooja] setup in the Contiki OS environment[contiki]. We used Contiki-2.7 running 6LoWPAN/RPL protocol stack for the simulations. For approximating the number of packet drops based on the RSSI values, we implemented simple logic that drops transmitted packets with certain pre-defined ratios before handing over the packets to the receiver. The packet drop ratio is implemented as a table lookup of RSSI ranges mapping to different packet drop ratios with lower RSSI ranges resulting in higher values. While this table has been defined for the purpose of capturing the overall link behavior, it is highly recommended to conduct physical radio measurement experiments, in general. By keeping the receiving node at different distances, we let the packets experience different packet drops as per the described method. The ETX value computation is done by another module which is part of RPL Objective Function implementation. Since ETX value is reflective of the extent of packet drops, it allowed us to prepare a useful ETX vs RSSI table. ETX versus RSSI values obtained in this way may be used as explained below:

```
        Source -------> NodeA -------> NodeB -----> Destination
```

Figure 7: Communication link from Source to Destination

RSSI at NodeA for NodeB	**Expected ETX at NodeA for NodeB->NodeA**
> -60	150
-70 to -60	192
-80 to -70	226
-90 to -80	662
-100 to -90	3840

Table 1: Selection of S bit based on Expected ETX value

Method 2: One could also make use of the function guess_etx_from_rssi() defined in the 6LoWPAN/RPL protocol stack of Contiki-ng OS [Contiki-ng] to obtain RSSI-ETX mapping. This function outputs ETX value ranging between 128 and 3840 for -60 <= rssi <= -89. The function description is beyond the scope of this document.

We tested the operations in this specification by making the following experiment, using the above parameters. In our experiment, a communication link is considered as symmetric if the ETX value of NodeA->NodeB and NodeB->NodeA (see Figure 7) are within, say, a 1:3 ratio. This ratio should be understood as determining the link's symmetric/asymmetric nature. NodeA can typically know the ETX value in the direction of NodeA -> NodeB but it has no direct way of knowing the value of ETX from NodeB->NodeA. Using physical testbed experiments and realistic wireless channel propagation models, one can determine a relationship between RSSI and ETX representable as an expression or a mapping table. Such a relationship in turn can be used to estimate ETX value at nodeA for link NodeB--->NodeA from the received RSSI from NodeB. Whenever nodeA determines that the link towards the nodeB is bi-directional asymmetric then the S bit is set to 0. Afterwards, the link from NodeA to Destination remains designated as asymmetric and the S bit remains set to 0.

Determination of asymmetry versus bidirectionality remains a topic of lively discussion in the IETF.

Appendix B. Changelog

Note to the RFC Editor: please remove this section before publication.

B.1. Changes from version 13 to version 14

*Provided more details about scenarios naturally supporting the choice of AODV-RPL as a routing protocol

*Added new informative references [RFC6687], [RFC9010]) that describe the value provided by peer-to-peer routing.

*Requested IANA to allocate a new multicast group to enable clean separation of AODV-RPL operation from previous routing protocols in the RPL family.

*Cited [RFC6550] as the origination of the definition of DIO

*Defined "hop-by-hop route" as a route created using RPL's storing mode.

*Defined new configuration variable REJOIN_REENABLE.

*Improved definition for RREQ-InstanceID. Created analogous definition for RREP-InstanceID=(RPLInstanceID, TargNode_IPaddr)

*Improved definition of source routing

*Clarified that the Border Router (BR) in Figure 4 does not imply that AODV does not a require a BR as a protocol entity.

*Provided more guidelines about factors to be considered by OrigNode when selecting a value for the 'L' field.

*Described the disadvantage of not keeping track of the Address Vector in the RREQ-Instance.

*Specified that in non-storing mode an intermediate node has to record the IP addresses of both incoming and outgoing interfaces into the Address Vector, when those interfaces have different IP addresses.

*Added three informative references to describe relevant details about evaluating link assymetry.

*Clarified details about Gratuitous RREP.

B.2. Changes from version 12 to version 13

*Changed name of "Shift" field to be the "Delta" field.

*Specified that if a node does not have resources, it MUST drop the RREQ.

*Changed name of MaxUseRank to MaxUsefulRank.

*Revised a sentence that was not clear about when a TargNode can delay transmission of the RREP in response to a RREQ.

*Provided advice about running AODV-RPL at same time as P2P-RPL or native RPL.

*Small reorganization and enlargement of the description of Trickle time operation in Section 8.

*Added definition for "RREQ-InstanceID" to Terminology section.

*Specified that once a node leaves an RREQ-Instance, it MUST NOT rejoin the same RREQ-Instance.

B.3. Changes from version 11 to version 12

*Defined RREP_WAIT_TIME for asymmetric as well as symmetric handling of RREP-DIO.

*Clarifed link-local multicast transmission to use link-local multicast group all-RPL nodes.

*Identified some security threats more explicitly.

*Specified that the pairing between RREQ-DIO and RREP-DIO happens at OrigNode and TargNode. Intermediate routers do not necessarily maintain the pairing.

*When RREQ-DIO is received with H=0 and S=1, specified that intermediate routers MAY store symmetric Address Vector information for possible use when a matchine RREP-DIO is received.

*Specified that AODV-RPL uses the "P2P Route Discovery Mode of Operation" (MOP == 4), instead of requesting the allocation of a new MOP. Clarified that there is no conflict with [RFC6997].

*Fixed several important typos and improved language in numerous places.

*Reorganized the steps in the specification for handling RREQ and RREP at an intermediate router, to more closely follow the order of processing actions to be taken by the router.

B.4. Changes from version 10 to version 11

*Numerous editorial improvements.

*Replace Floor((7+(Prefix Length))/8) by Ceil(Prefix Length/8) for simplicity and ease of understanding.

*Use "L field" instead of "L bit" since L is a two-bit field.

*Improved the procedures in section 6.2.1.

*Define the S bit of the data structure a router uses to represent whether or not the RREQ instance is for a symmetric or an asymmetric route. This replaces text in the document that was a holdover from earlier versions in which the RREP had an S bit for that purpose.

*Quote terminology from AODV that has been identified as possibly originating in language reflecting various kinds of bias against certain cultures.

*Clarified the relationship of AODV-RPL to RPL.

*Eliminated the "Point-to-Point" terminology to avoid suggesting only a single link.

*Modified certain passages to better reflect the possibility that a router might have multiple IP addresses.

*"Rsv" replaced by "X X" for reserved field.

*Added mandates for reserved fields, and replaces some ambiguous language phraseology by mandates.

*Replaced "retransmit" terminology by more correct "propagate" terminology.

*Added text about determining link symmetry near Figure 5.

*Mandated checking the Address Vector to avoid routing loops.

*Improved specification for use of the Delta value in Section 6.3.3.

*Corrected the wrong use of RREQ-Instance to be RREP-Instance.

*Referred to Subregistry values instead of Registry values in Section 9.

*Sharpened language in Section 10, eliminated misleading use of capitalization in the words "Security Configuration".

*Added acknowledgements and contributors.

B.5. Changes from version 09 to version 10

*Changed the title for brevity and to remove acronyms.

*Added "Note to the RFC Editor" in Section 9.

*Expanded DAO and P2MP in Section 1.

*Reclassified [RFC6998] and [RFC7416] as Informational.

*SHOULD changed to MUST in Section 4.1 and Section 4.2.

*Several editorial improvements and clarifications.

B.6. Changes from version 08 to version 09

*Removed section "Link State Determination" and put some of the relevant material into Section 5.

*Cited security section of [RFC6550] as part of the RREP-DIO message description in Section 2.

*SHOULD has been changed to MUST in Section 4.2.

*Expanded the terms ETX and RSSI in Section 5.

*Section 6.4 has been expanded to provide a more precise explanation of the handling of route reply.

*Added [RFC7416] in the Security Considerations (Section 10) for RPL security threats. Cited [RFC6550] for authenticated mode of operation.

*Appendix A has been mostly re-written to describe methods to determine whether or not the S bit should be set to 1.

*For consistency, adjusted several mandates from SHOULD to MUST and from SHOULD NOT to MUST NOT.

*Numerous editorial improvements and clarifications.

B.7. Changes from version 07 to version 08

*Instead of describing the need for routes to "fulfill the requirements", specify that routes need to "satisfy the Objective Function".

*Removed all normative dependencies on [RFC6997]

*Rewrote Section 10 to avoid duplication of language in cited specifications.

*Added a new section "Link State Determination" with text and citations to more fully describe how implementations determine whether links are symmetric.

*Modified text comparing AODV-RPL to other protocols to emphasize the need for AODV-RPL instead of the problems with the other protocols.

*Clarified that AODV-RPL uses some of the base RPL specification but does not require an instance of RPL to run.

*Improved capitalization, quotation, and spelling variations.

*Specified behavior upon reception of a RREQ-DIO or RREP-DIO message for an already existing DODAGID (e.g, Section 6.4).

*Fixed numerous language issues in IANA Considerations Section 9.

*For consistency, adjusted several mandates from SHOULD to MUST and from SHOULD NOT to MUST NOT.

*Numerous editorial improvements and clarifications.

B.8. Changes from version 06 to version 07

*Added definitions for all fields of the ART option (see Section 4.3). Modified definition of Prefix Length to prohibit Prefix Length values greater than 127.

*Modified the language from [RFC6550] Target Option definition so that the trailing zero bits of the Prefix Length are no longer described as "reserved".

*Reclassified [RFC3561] and [RFC6998] as Informative.

*Added citation for [RFC8174] to Terminology section.

B.9. Changes from version 05 to version 06

*Added Security Considerations based on the security mechanisms defined in [RFC6550].

*Clarified the nature of improvements due to P2P route discovery versus bidirectional asymmetric route discovery.

*Editorial improvements and corrections.

B.10. Changes from version 04 to version 05

*Add description for sequence number operations.

*Extend the residence duration L in section 4.1.

*Change AODV-RPL Target option to ART option.

B.11. Changes from version 03 to version 04

*Updated RREP option format. Remove the T bit in RREP option.

*Using the same RPLInstanceID for RREQ and RREP, no need to update [RFC6550].

*Explanation of Delta field in RREP.

*Multiple target options handling during transmission.

B.12. Changes from version 02 to version 03

*Include the support for source routing.

*Import some features from [RFC6997], e.g., choice between hop-by-hop and source routing, the L field which determines the duration of residence in the DAG, RankLimit, etc.

*Define new target option for AODV-RPL, including the Destination Sequence Number in it. Move the TargNode address in RREQ option and the OrigNode address in RREP option into ADOV-RPL Target Option.

*Support route discovery for multiple targets in one RREQ-DIO.

*New RPLInstanceID pairing mechanism.

Appendix C. Contributors

*Abdur Rashid Sangi

Huaiyin Institute of Technology

No.89 North Beijing Road, Qinghe District

Huaian 223001

P.R. China

Email: sangi_bahrian@yahoo.com

*Malati Hegde

Indian Institute of Science

Bangalore 560012

India

Email: malati@iisc.ac.in

*Mingui Zhang

Huawei Technologies

No. 156 Beiqing Rd. Haidian District

Beijing 100095

P.R. China

Email: zhangmingui@huawei.com

Authors' Addresses

Charles E. Perkins
Lupin Lodge
Los Gatos, 95033
United States

Email: charliep@lupinlodge.com

S.V.R Anand
Indian Institute of Science
Bangalore 560012
India

Email: anandsvr@iisc.ac.in

Satish Anamalamudi
SRM University-AP
Amaravati Campus

Amaravati, Andhra Pradesh 522 502
India

Email: satishnaidu80@gmail.com

Bing Liu
Huawei Technologies
No. 156 Beiqing Rd. Haidian District
Beijing
100095
China

Email: remy.liubing@huawei.com