Stream: Internet Engineering Task Force (IETF)
RFC: 9854
Category: Standards Track
Published: October 2025
ISSN: 2070-1721

Authors:
C.E. Perkins — *Blue Meadow Networks*
S.V.R. Anand — *Indian Institute of Science*
S. Anamalamudi — *SRM University-AP*
B. Liu — *Huawei Technologies*

RFC 9854
AODV-RPL: The Routing Protocol for Low-Power and Lossy Networks (RPL) Based on Ad Hoc On-Demand Distance Vector (AODV) Routing

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 AODV-RPL -- the Routing Protocol for Low-Power and Lossy Networks (RPL) based on Ad hoc On-demand Distance Vector (AODV) routing. AODV-RPL is a reactive P2P route discovery mechanism for both hop-by-hop routes and source routing. 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

The 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 destinations attached to most other routers. Consequently, for traffic between routers within the DODAG (i.e., 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], and [RFC9010]). The network environment that is considered in this document is assumed to be the same as that 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 link symmetry apply independently for each of these.

The route discovery process in AODV-RPL is modeled on the analogous P2P procedure specified in AODV [RFC3561]. The on-demand property of AODV route discovery is useful for the needs of routing in RPL-based LLNs when routes are needed but aren't yet established. P2P routing is desirable to discover shorter routes, 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 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, blocking the use of 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 (OF) (defined in [RFC6551]), Trickle timers, and support for storing and non-storing modes. AODV-RPL adds the basic messages RREQ and RREP as part of the 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 discovering P2P routes. These P2P routes may differ from routes discoverable by RPL [RFC6550]. 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 RPL [RFC6550] is also running. AODV-RPL could be used for networks in which routes are needed with OFs that cannot be satisfied by routes that are constrained to traverse the root of the network or other common ancestors. P2P routes often require fewer hops and therefore consume less resources than routes that traverse the root or other common ancestors. Similar in cost to base RPL [RFC6550], the cost will depend on many factors such as the proximity of the OrigNode and TargNodes and distribution of symmetric/asymmetric P2P links. Experience with AODV [aodv-tot] suggests that AODV-RPL will often find routes with improved Rank compared to routes constrained to traverse a common ancestor of the source and destination nodes.

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].

This document also uses the following terms:

AODV
: Ad hoc On-Demand Distance Vector [RFC3561].

ART option
: The AODV-RPL 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 OF during route discovery.

Bidirectional 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)
: A 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)
: A 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".

OF
: Objective Function (as defined in [RFC6550]).

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).

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_REENABLE is 15 minutes.

RREQ
: Route Request.

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
: Route Reply.

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. The InstanceIDs are matched by the mechanism explained in Section 6.3.3.

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.

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 do not become established until they are needed. The route discovery mechanism in AODV-RPL is invoked when OrigNode has data for delivery to a TargNode, but existing routes do not satisfy the application's requirements. For this reason, AODV-RPL is considered to be an example of an "on-demand" routing protocol. Such protocols are also known as "reactive" routing protocols since their operations are triggered in reaction to a determination that a new route is needed. 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 Directed Acyclic Graph (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 messages. AODV-RPL uses the same notion of Rank as defined in [RFC6550]:

> The Rank is the expression of a relative position within a DODAG Version with regard to neighbors, and it is not necessarily a good indication or a proper expression of a distance or a path cost to the root.

The Rank measurements provided in AODV messages do not indicate a distance or a path cost to the root.

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. The D bit of the RPLInstanceID field is set to 0 to indicate that the source address of the IPv6 packet is the DODAGID.

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 as above.

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); otherwise, the route discovery will fail. Exactly one RREQ option **MUST** be present in an 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
: 8-bit unsigned integer specifying the type of the option (0x0B).

Option Length
: 8-bit unsigned integer specifying the length of the option in octets, excluding the Option Type and Option Length fields. It is 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. This field **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 **SHOULD** leave the RREQ-Instance and stop sending or receiving any more DIOs for the RREQ-Instance; otherwise, memory and network resources are likely to be consumed unnecessarily. 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
: 8-bit unsigned integer specifying 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
: 8-bit unsigned integer specifying the 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 an RREQ-Instance if its own Rank would be equal to or higher than the 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 an 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
: 8-bit unsigned integer specifying the type of the option (0x0C).

Option Length
: 8-bit unsigned integer specifying the length of the option in octets, excluding the Option Type and Option Length fields. It is 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 the RREQ option. It requests either source routing (H=0) or hop-by-hop (H=1) for the downstream route.

X
: 1-bit Reserved field. This field **MUST** be initialized to zero and ignored upon reception.

Compr
: 4-bit unsigned integer. This field has the same definition as in the RREQ option.

L
: 2-bit unsigned integer defined as in the RREQ option. The lifetime of the RREP-Instance **SHOULD** be no greater than the lifetime of the RREQ-Instance to which it is paired, so that the memory required to store the RREP-Instance can be reclaimed when no longer needed.

RankLimit
: 8-bit unsigned integer specifying the upper limit on the integer portion of the Rank, similarly to RankLimit in the RREQ message. A value of 0 in this field indicates the limit is infinity.

Delta
: 6-bit unsigned integer. TargNode uses the Delta field so that nodes receiving its RREP message can identify the RREQ-InstanceID of the RREQ message that triggered the transmission of the RREP (see Section 6.3.3).

X X
: 2-bit Reserved field. This field **MUST** be initialized to zero and ignored upon reception.

Address Vector
: Only present when the H bit is set to 0. The prefix of each address is elided according to the Compr field. For an asymmetric route, the Address Vector represents the IPv6 addresses of the path through the network the RREP-DIO has passed. In contrast, 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 the core RPL specification [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.

An RREQ-DIO message **MUST** carry at least one ART option. An RREP-DIO message **MUST** carry exactly one ART option. Otherwise, the message **MUST** be dropped.

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

```
     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
: 8-bit unsigned integer specifying the type of the option (0x0D).

Option Length
: 8-bit unsigned integer specifying the length of the option in octets, excluding the Option Type and Option Length fields.

Dest SeqNo
: 8-bit unsigned integer. 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

1-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. The Prefix Length field contains the number of valid leading bits in the 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

A variable-length field with an IPv6 destination address or prefix. The length of the Target Prefix / Address field is the least number of octets that can represent all of the bits of the Prefix, in other words, Ceil(Prefix Length/8) octets. When Prefix Length is not equal to 8*Ceil(Prefix Length/8) and nonzero, the Target Prefix / Address field will contain some initial bits that are not part of the Target Prefix. Those initial bits (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.

5. Symmetric and Asymmetric Routes

Links are considered symmetric until indication to the contrary is received. In Figures 4 and 5, BR is the Border Router, O is the OrigNode, each R is an intermediate router, and T is the TargNode. In these examples, 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.

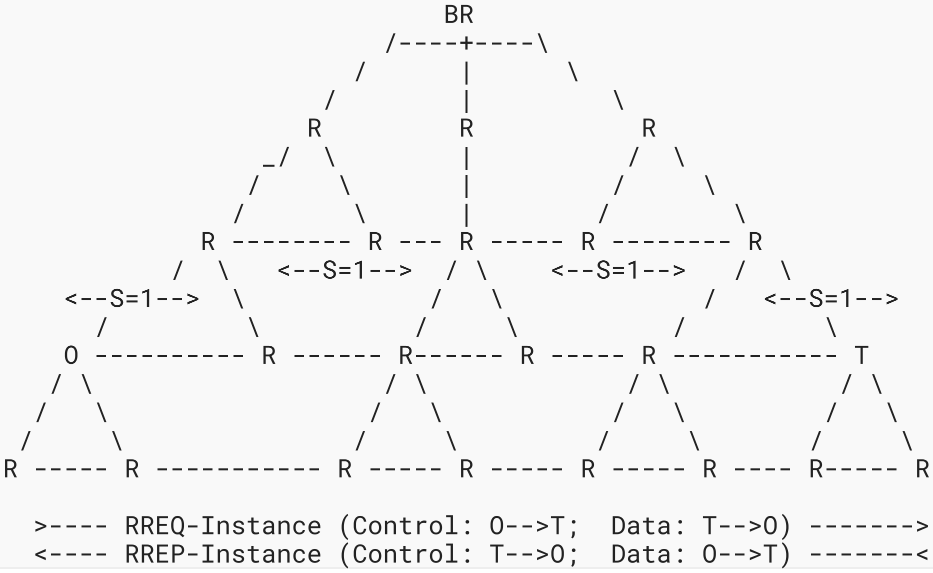
BR
R
R
R
R
R
R
R
R
<--S=1-->
<--S=1-->
<--S=1-->
<--S=1-->
O
R
R
R
R
T
R
R
R
R
R
R
R
R
>---- RREQ-Instance (Control: O-->T; Data: T-->O) ------->
<---- RREP-Instance (Control: T-->O; Data: O-->T) -------<

Figure 4: AODV-RPL with Symmetric Instances

Upon receiving an RREQ-DIO with the S bit set to 1, a node determines whether the link over which it was received 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, then it is set to be 0 when the RREQ-DIO is propagated (Figure 5). For an asymmetric route, there is at least one hop that doesn't satisfy the OF. 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 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, Operations, Administration, and Maintenance (OAM) techniques for evaluating link state (see [RFC7548], [RFC7276], and [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 Transmission Count (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.

See Appendix B for examples illustrating RREQ and RREP transmissions in some networks with symmetric and asymmetric links.

6. AODV-RPL Operation

6.1. Generating RREQ

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 OF for the target of the application's data. In this case, the OrigNode builds a local RPL Instance 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 an RREQ-DIO, the OF for the routes to different TargNodes **MUST** be the same.

OrigNode can maintain different RPL Instances 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 RPL Instances 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 RPL Instance. OrigNode needs to set the 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 the 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 an RREQ message over a link from a neighbor Y, X first determines whether or not the RREQ is valid. If valid, X then determines whether or not it has sufficient resources available to maintain the RREQ-Instance and the value of the S bit needed to process an eventual RREP, if the RREP were to be received. If not valid, then X **MUST** either free up sufficient resources (the means for this are beyond the scope of this document), or 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 transmit 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 the 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 in the RREQ-Instance.

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, 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 an 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 RPL Instance 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.

The reasoning for using the intersection of the lists in the RREQs is as follows. When two or more RREQs are received with the same Orig SeqNo, they were transmitted by OrigNode with the same destinations and OF. When an intermediate node receives two RREQs with the same Orig SeqNo but different lists of destinations, that means that some intermediate nodes retransmitting the RREQs have already deleted themselves from the list of destinations before they retransmitted the RREQ. Those deleted nodes are not to be reinserted back into the list of destinations.

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 can be learned from the DODAGID and the RPLInstanceID of the RREQ-DIO, respectively. 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 and the same RPLInstanceID, but a stale Sequence Number (i.e., incoming Sequence Number is less than the currently stored Sequence Number of the route entry), **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 direction (i.e., towards the TargNode) of the incoming link satisfies the OF. If it does, 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 (see Section 6.3.2).

Suppose a router has joined the RREQ-Instance, the H bit is set to 0, and the S bit of the RREQ-Instance is set to 1. In this case, the router **MAY** optionally include the Address Vector of the symmetric route back to OrigNode as part of the RREQ-Instance data. This is useful if the router later receives an RREP-DIO that is paired with the RREQ-Instance. If the router does NOT include the Address Vector, then it has to rely on multicast for the RREP. The multicast 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. In addition, if the address of the router's interface 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 an RREP-DIO, possibly after waiting for RREP_WAIT_TIME, as detailed in (Section 6.3).

6.3. Generating RREP at TargNode

When a TargNode receives an 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 transmit packets to OrigNode, TargNode generates an RREP according to Sections 6.3.1 and 6.3.2. Otherwise, TargNode drops the RREQ and does not generate an RREP.

If the L field is not 0, the TargNode **MAY** delay transmitting the RREP-DIO for the 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 OF. 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. If 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 an 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 OFs 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 that 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 reused for two distinct DODAGs originated with the same DODAGID (TargNode address), intermediate routers could not distinguish between these DODAGs (and their associated OFs). 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 value that TargNode adds to the RPLInstanceID in the RREQ-DIO that it received, to obtain the value of the RPLInstanceID it uses in the RREP-DIO message. 0 indicates that the RREQ-InstanceID has the same value as the RPLInstanceID of the RREP 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 RREP

Upon receiving an RREP-DIO, a router that 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 OF 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 an 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 OF and whether the router's Rank would not exceed the RankLimit. If these are true, 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 it is included, 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 that 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 the same source and destination address and the same RPLInstanceID, but a stale Sequence Number, **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 previous 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 the 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 the 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 an RREQ-DIO message **MAY** unicast a Gratuitous RREP-DIO (G-RREP-DIO) message back to OrigNode before continuing the transmission of the RREQ-DIO towards TargNode. The Gratuitous RREP (G-RREP) allows the OrigNode to start transmitting data to TargNode sooner. The G bit of the RREP option is provided to distinguish the G-RREP-DIO (G=1) sent by the intermediate router from the RREP-DIO sent by TargNode (G=0).

The G-RREP-DIO **MAY** be sent out when the intermediate router receives an RREQ-DIO for a TargNode and the router has a pair of downward and upward routes to the TargNode that also satisfy the OF and for which the Destination Sequence Number is at least as large as the Sequence Number in the RREQ-DIO message. After unicasting the G-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. An upstream intermediate router that receives such a G-RREP **MUST** also generate a G-RREP and send it further upstream towards OrigNode.

In case of source routing, the intermediate router **MUST** include the Address Vector between the OrigNode and itself in the G-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 G-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 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

AODV-RPL uses the "P2P Route Discovery Mode of Operation" (MOP == 4), with new options as specified in this document. This document has been added as an additional reference for "P2P Route Discovery Mode of Operation" in the "Mode of Operation" registry within the "Routing Protocol for Low Power and Lossy Networks (RPL)" registry group.

IANA has assigned the three new AODV-RPL options described in Table 1 in the "RPL Control Message Options" registry within the "Routing Protocol for Low Power and Lossy Networks (RPL)" registry group.

Value	Meaning	Reference
0x0B	RREQ	RFC 9854
0x0C	RREP	RFC 9854
0x0D	ART	RFC 9854

Table 1: AODV-RPL Options

IANA has allocated the permanent multicast address with link-local scope in Table 2 for nodes implementing this specification. This allocation has been made in the "Local Network Control Block (224.0.0.0 - 224.0.0.255 (224.0.0/24))" registry within the "IPv4 Multicast Address Space Registry" registry group.

Address(es)	Description	References
224.0.0.69	all-AODV-RPL-nodes	RFC 9854

Table 2: Permanent Multicast Address with Link-Local Scope

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 an 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 G-RREP, it could mount denial-of-service attacks.

11. References

11.1. 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>.

11.2. Informative References

[aodv-tot] Perkins, C.E. and E.M. Royer, "Ad-hoc On-demand Distance Vector Routing", Proceedings WMCSA'99. Second IEEE Workshop on Mobile Computing Systems and Applications, pp. 90-100, DOI 10.1109/MCSA.1999.749281, February 1999, <https://ieeexplore.ieee.org/document/749281>.

[co-ioam] Ballamajalu, R., Anand, S.V.R., and M. 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, DOI 10.1109/COMSNETS.2018.8328276, January 2018, <https://ieeexplore.ieee.org/document/8328276>.

[contiki] "The Contiki Open Source OS for the Internet of Things (Contiki Version 2.7)", commit 7635906, November 2013, <https://github.com/contiki-os/contiki>.

[Contiki-ng] "Contiki-NG: The OS for Next Generation IoT Devices (Contiki-NG Version 4.6)", commit 3b0bc6a, December 2020, <https://github.com/contiki-ng/contiki-ng>.

[cooja] "Cooja Simulator for Wireless Sensor Networks (Contiki/Cooja Version 2.7)", commit 7635906, November 2013, <https://github.com/contiki-os/contiki/tree/master/tools/cooja>.

[empirical-study] Misra, P., Ahmed, N., and S. Jha, "An empirical study of asymmetry in low-power wireless links", IEEE Communications Magazine, vol. 50, no. 7, pp. 137-146, DOI 10.1109/MCOM.2012.6231290, July 2012, <https://ieeexplore.ieee.org/document/6231290>.

[Link_Asymmetry] Sang, L., Arora, A., and H. Zhang, "On Link Asymmetry and One-way Estimation in Wireless Sensor Networks", ACM Transactions on Sensor Networks, vol. 6, no. 2, pp. 1-25, DOI 10.1145/1689239.1689242, March 2010, <https://doi.org/10.1145/1689239.1689242>.

[low-power-wireless] Srinivasan, K., Dutta, P., Tavakoli, A., and P. Levis, "An empirical study of low-power wireless", ACM Transactions on Sensor Networks, vol. 6, no. 2, pp. 1-49, DOI 10.1145/1689239.1689246, March 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 the downstream Received Signal Strength Indicator (RSSI) and the upstream Expected Transmission Count (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 versus ETX using the Cooja simulation [cooja] setup in the Contiki OS environment [contiki]. We used Contiki-2.7 running the 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 predefined 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, in general, it is highly recommended to conduct physical radio measurement experiments. 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 that is part of RPL OF implementation. Since the ETX value is reflective of the extent of packet drops, it allowed us to prepare a useful table correlating ETX and RSSI values (see Table 3). ETX and RSSI values obtained in this way may be used as explained below:

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

Figure 6: 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 3: 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 an 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 6) 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 the ETX value at NodeA for link NodeB->NodeA from the received RSSI from NodeB. Whenever NodeA determines that the link towards the NodeB is bidirectional 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. Some Example AODV-RPL Message Flows

This appendix provides some example message flows showing RREQ and RREP establishing symmetric and asymmetric routes. Also, examples for the use of RREP_WAIT and G-RREP are included. In the examples, router (O) is to be understood as performing the role of OrigNode. Router (T) is to be understood as performing the role of TargNode. Routers (R) are intermediate routers that are performing AODV-RPL functions in order to discover one or more suitable routes between (O) and (T).

B.1. Example Control Message Flows in Symmetric and Asymmetric Networks

In the following diagram, RREQ messages are multicast from router (O) in order to discover routes to and from router (T). The RREQ control messages flow outward from (O). Each router along the way establishes a single RREQ-Instance identified by RREQ-InstanceID even if multiple RREQs are received with the same RREQ-InstanceID. In the top half of the diagram, the routers are able to offer a symmetric route at each hop of the path from (O) to (T). When (T) receives an RREQ, it is then able to transmit data packets to (O). Router (T) then prepares to send an RREP along the symmetric path that would enable router (O) to send packets to router (T).

```
      (R) ---RREQ(S=1)--->(R) ---RREQ(S=1)--->(R)
       ^                                       |
       |                                       |
  RREQ(S=1)                                RREQ(S=1)
       |                                       |
       |                                       v
      (O) --------->(R) --------->(R)-------->(T)
      / \    RREQ         RREQ         RREQ    ^
      |  \    (S=1)      (S=0)         (S=0)   |
      |   \                                   /
 RREQ |    \ RREQ (S=1)                    RREQ (S=0)
(S=0) |     \                               /
      v      \              RREQ (S=0)     /
     (R) ---->(R)------>(R)----.....--->(R)
```

Figure 7: AODV-RPL RREQ Message Flow Example When Symmetric Path Available

In the following diagram, which results from the above RREQ message transmission, a symmetric route is available from (T) to router (O) via the routers in the top half of the diagram. RREP messages are sent via unicast along the symmetric route. Since the RREP message is transmitted via unicast, no RREP messages are sent by router (T) to the routers in the bottom half of the diagram.

```
 (R)<------RREP----- (R)<------RREP----- (R)
  |                                       ^
  |                                       |
 RREP                                    RREP
  |                                       |
  v                                       |
 (O) ----------(R) ----------(R) --------(T)
 / \                                      |
 |  \                                     |
 |   \     (no RREP messages sent)       /
 |    \                                 /
 |     \                               /
 |      \                             /
(R) -----(R)-------(R)----.....----(R)
```

Figure 8: AODV-RPL RREP Message Flow Example When Symmetric Path Available

In the following diagram, RREQ messages are multicast from router (O) in order to discover routes to and from router (T) as before. As shown, no symmetric route is available from (O) to (T).

```
   (R) ---RREQ(S=0)--->(R) ---RREQ(S=0)--->(R)
    ^                                       |
    |                                       |
RREQ(S=1)                                RREQ(S=0)
    |                                       |
    |                                       v
   (O) --------->(R) --------->(R)-------->(T)
   ^ \    RREQ          RREQ      RREQ      |  \
   |  \    (S=1)        (S=0)       (S=0)  |   |
   |   \                                   /   |
   |  RREQ (S=1)              RREQ (S=0)  /   (R)
   |     \                               /     |
   |      \               RREQ (S=0)    /     /
  (R) ---->(R)------>(R)----.....----->(R)---
```

Figure 9: AODV-RPL RREQ Message Flow When Symmetric Path Unavailable

Upon receiving the RREQ in Figure 9, router (T) then prepares to send an RREP that would enable router (O) to send packets to router (T). In Figure 9, since no symmetric route is available from (T) to router (O), RREP messages are sent via multicast to all neighboring routers.

```
      (R)<------RREP----- (R)<------RREP----- (R)
       |                                       |
       |                                       |
      RREP                                    RREP
       |                                       |
       |                                       |
       v                                       v
      (O)<--------- (R)<--------- (R)<------- (T)
      ^ \    RREP         RREP        RREP    |  \
      |  \                                    |   |RREP
      |   \                                   /   |
RREP  |    \ RREP                  RREP      /   (R)
      |     \                               /     |
      |      \                             /     /
    (R)<----- (R)<----- (R)<---.....---- (R)< - RREP
         RREP       RREP         RREP
```

Figure 10: AODV-RPL RREQ and RREP-Instances for Asymmetric Links

B.2. Example RREP_WAIT Handling

In Figure 11, the first RREQ arrives at (T). This triggers TargNode to start the RREP_WAIT_TIME timer.

```
(O) --------->(R) --------->(R)-------->(T)
        RREQ         RREQ         RREQ
       (S=1)        (S=0)        (S=0)
```

Figure 11: TargNode Starts RREP_WAIT

In Figure 12, another RREQ arrives before the RREP_WAIT_TIME timer is expired. It could be preferable compared the previously received RREP that caused the RREP_WAIT_TIME timer to be set.

```
      (O)                                     (T)
      / \                                      ^
      |  \                                     |
      |   \                                   /
 RREQ |    \ RREQ (S=1)                    RREQ (S=0)
(S=0) |     \                               /
      v      \              RREQ (S=0)     /
     (R) ---->(R)------>(R)----.....--->(R)
```

Figure 12: Waiting TargNode Receives Preferable RREQ

In Figure 13, the RREP_WAIT_TIME timer expires. TargNode selects the path with S=1.

```
    (R) ---RREQ(S=1)--->(R) ---RREQ(S=1)--->(R)
     ^                                       |
     |                                       |
RREQ(S=1)                                RREQ(S=1)
     |                                       |
     |                                       v
    (O)                                     (T)
```

Figure 13: RREP_WAIT Expires at TargNode

B.3. Example G-RREP Handling

In Figure 14, R* has upward and downward routes to TargNode (T) that satisfy the OF of the RPL Instance originated by OrigNode (O), and the Destination Sequence Number is at least as large as the Sequence Number in the RREQ message.

```
      (R) ---RREQ(S=1)--->(R) ---RREQ(S=0)--->(R)
       ^                                       |
       |                                       |
  RREQ(S=1)                                RREQ(S=0)
       |                                       |
       |                                       v
      (O) --------->(R) --------->(R)-------->(T)
      / \    RREQ         RREQ         RREQ    ^
      |  \    (S=1)      (S=0)         (S=0)   |
      |   \                                   /
 RREQ |    \ RREQ (S=1)                      /
(S=0) |     \                               /
      v      \                             v
     (R) ---->(R*)<------>(R)<----....--->(R)
```

Figure 14: RREP Triggers G-RREP at Intermediate Node

In Figure 15, R* transmits the G-RREP-DIO back to OrigNode (O) and forwards the incoming RREQ towards (T).

```
(O)                                     (T)
  \                                      ^
   \                                     |
    \                            (RREQ) /
     \ G-RREP-DIO                      /
      \                               /
       \   (RREQ)       (RREQ)       /
       (R*)------>(R)----....--->(R)
```

Figure 15: Intermediate Node Initiates G-RREP

Acknowledgements

The authors thank Pascal Thubert, Rahul Jadhav, and Lijo Thomas for their support and valuable input. 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 reviews, 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, Éric Vyncke, and Robert Wilton.

Contributors

Abdur Rashid Sangi
Wenzhou-Kean University
88 Daxue Rd, Ouhai
Wenzhou
Zhejiang Province, 325060
Kean University
1000 Morris Avenue
Union, New Jersey 07083
United States of America
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
China
Email: zhangmingui@huawei.com

Authors' Addresses

Charles E. Perkins
Blue Meadow Networks
Saratoga, CA 95070
United States of America
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