Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: May 3, 2012

T. Clausen
A. Colin de Verdiere
J. Yi
LIX, Ecole Polytechnique
A. Niktash
Maxim Integrated Products
Y. Igarashi
SATOH. H.
Hitachi, Ltd., Yokohama Research Laboratory
U. Herberg
Fujitsu Laboratories of America

October 31, 2011

The LLN On-demand Ad hoc Distance-vector Routing Protocol - Next Generation (LOADng)
draft-clausen-lln-loadng-01

Abstract

This document describes the LLN Ad hoc On-Demand - Next Generation (LOADng) distance vector routing protocol, a reactive routing protocol intended for use in Low power and Lossy Networks (LLN). The protocol is derived from AODV and extended for use in LLNs.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

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

This Internet-Draft will expire on May 3, 2012.

Copyright Notice

Table of Contents

1. Introduction

The LLN On-demand Ad hoc Distance-vector Routing Protocol - Next Generation (LOADng) is a routing protocol, derived from AODV [RFC3561] and extended for use in Low power and Lossy Networks (LLNs). As a reactive protocol, the basic operations of LOADng include generation of Route Requests (RREQs) by a router (originator) for when discovering a route to a destination, forwarding of such RREQs until they reach the destination router, generation of Route Replies (RREPs) upon receipt of a RREQ by the indicated destination, and unicast hop-by-hop forwarding of these RREPs towards the originator. If a route is detected broken, i.e., if forwarding of a data packet to the recorded next hop on the route to the destination is detected to fail, local route repair can be attempted, or a Route Error (RERR) message can be returned to the originator of that data packet.

Compared to [RFC3561], LOADng is simplified as follows:

- Only the destination is permitted to respond to a RREQ; intermediate routers are explicitly prohibited from responding to RREQs, even if they may have active routes to the sought destination, and all messages (RREQ or RREPs) generated by a given router share a single unique, monotonically increasing sequence number. This also eliminates Gratuitous RREPs while ensuring loop freedom. The rationale for this simplification is reduced complexity of protocol operation and reduced message sizes.

- A LOADng router does not maintain a precursor list, thus when

forwarding of a data packet to the recorded next hop on the route to the destination fails, a RERR is sent only to the originator of that data packet. The rationale for this simplification is an assumption that few overlapping routes are in use concurrently in a given network.

Compared to [RFC3561], LOADng is extended as follows:

- Optimized Flooding is supported, reducing the overhead incurred by RREQ generation and flooding.

- Different address lengths are supported - from full 16 octet IPv6 addresses over 6 octet Ethernet MAC addresses and 4 octet IPv4 addresses to shorter 1 and 2 octet addresses. The only requirement is, that within a given routing domain, all addresses are of the same address length.

- Control messages can include TLV (Type-Length-Value) elements, permitting protocol extensions to be developed.

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 [RFC2119].

Additionally, this document uses the following terminology:

LOADng Router - A router which implements this routing protocol.

Link Cost - The cost (weight) between a pair of LOADng Routers, determined by a LOADng router upon receipt of a packet.

Route Cost - The sum of the Link Costs for the links that a RREQ or RREP has crossed.

Weak Link - A link which is marginally usable, i.e., which MAY be used if no other links are available, but which SHOULD be avoided if at all possible - even if it entails ultimately longer routes. As an example, a Weak Link might be defined as a link with a nominatively high bit-rate (thus, a priori attractive) while

suffering a significant loss-rate.

This document uses the following notational conventions:

a := b An assignment operator, whereby the left side (a) is assigned the value of the right side (b).

c = d A comparison operator, returning true if the value of the left side (c) is equal to the value of the right side (d).

3. Applicability Statement

This protocol:

- Is a reactive routing protocol for Low power and Lossy Networks (LLNs).
- Supports the use of optimized flooding for RREQs.
- Enables any router in the LLN to discover bi-directional routes to any other router in the LLN.
- Supports addresses of any length, from 16 octets to a single octet.

- Is layer-agnostic, i.e., may be used at layer 3 as a "route over" routing protocol, or at layer 2 as a "mesh under" routing protocol.
- Supports per-route maintenance; if a destination becomes unreachable, rediscovery of that single (bi-directional) route is performed, without need for global topology recalculation.

4. Protocol Overview and Functioning

The objective of this protocol is for each LOADng router to, independently:

- Discover a bi-directional route to any destination in the network.

- Establish routes only when there is data traffic to be sent along that route.

- Maintain a route only for as long as it is an active route, i.e., there is traffic using the route.

- Generate control traffic based on network events only: when a new route is required, or when an active route is detected broken. Specifically, this protocol does not require periodic signaling.

4.1. Overview

These objectives are achieved, for each LOADng router, by:

- When having a data packet to deliver to a destination, for which no entry in the routing table exists, generating a Route Request (RREQ) encoding the destination address, and transmitting this to all of its neighbors.

- Upon receiving a RREQ, install or refresh an entry in the routing table towards the originator address from the RREQ, as well as to the neighbor LOADng router from which the RREQ was received. This will install the Reverse Route (towards the originator address from the RREQ).

- Upon receiving a RREQ, inspect the indicated destination address:

  * If that address is an address in the Destination Address Set of the LOADng router, generate a Route Reply (RREP), which is unicast in a hop-by-hop fashion along the installed Reverse Route.

  * If that address is not an address in the Destination Address Set of the LOADng router, consider the RREQ as a candidate for forwarding.

- When a RREQ is considered a candidate for forwarding, retransmit it according to the flooding operation specified for the network.

- Upon receiving a RREP, install a route towards the originator

address from the RREP, as well as to the neighbor LOADng router, from which that RREP was received. This will install the Forward Route (towards the originator address from the RREP).

- Upon receiving a RREP, forward it, as unicast, to the recorded next hop along the corresponding Reverse Route until the RREP gets to the LOADng router which has the destination address from the RREP in its Destination Address Set.

4.2. Routers and Interfaces

In order for a LOADng router to participate in a LLN, it MUST have at least one, and possibly more, LOADng interfaces. Each LOADng interface:

- Is configured with one or more interface addresses.

In addition to a set of LOADng interfaces as described above, each LOADng router:

- Has a number of router parameters.
- Has an Information Base.
- Generates and processes RREQ, RREP, RREP-ACK and RERR messages.

4.3. Information Base Overview

Necessary protocol state is recorded by way of four information sets: the "Routing Set", the "Blacklisted Neighbor Set", the "Destination Address Set", and the "Pending Acknowledgement Set".

The Routing Set contains tuples, representing next-hop, distance towards a destination address and the sequence number, all extracted from the message (RREQ or RREP) that generated the tuple so as to enable routing. The routing table is to be updated using this Routing Set. (A router MAY choose to use any or all destination addresses in the Routing Set to update the routing table, this selection is outside the scope of this specification.)

The Blacklisted Neighbor Set contains tuples representing neighbor

LOADng routers with which unidirectional connectivity has been detected.

The Destination Address Set contains tuples representing addresses, for which the LOADng router is responsible; i.e. addresses of this LOADng router, or of hosts and networks directly attached to this router and which use it to connect to the LLN. These addresses may in particular belong to devices which do not implement LOADng, and thus cannot process LOADng messages. This router SHOULD provide connectivity to these addresses by generating RREPs in response to RREQs directed towards them.

The Pending Acknowledgement Set contains tuples, representing transmitted RREPs for which an RREP-ACK is expected, but where this RREP-ACK has not yet been received.

The Routing Set and the Blacklisted Neighbor Set is updated by this protocol. The Destination Address Set is used, but not updated, by this protocol.

4.4. Signaling Overview

This protocol generates and processes the following routing messages:

Route Request (RREQ) - Generated by a LOADng router when it has a data packet to deliver to a given destination, but when it does not have an entry in its Routing Set indicating a route to that destination. A RREQ contains:

* The address (destination) to which a Forward Route is to be discovered by way of soliciting the LOADng router with that destination address in its Destination Address Set to generate a RREP.

* The address for which a Reverse Route is to be installed (originator) by RREQ forwarding and processing.

* The sequence number of the LOADng router, generating the RREQ.

A RREQ is flooded through the network, possibly employing an optimized flooding mechanism.

Route Reply (RREP) - Generated as a response to a RREQ by the LOADng router which has the address (destination) from the RREQ in its Destination Address Set. A RREP is sent in unicast towards the originator of that RREQ. A RREP contains:

* The address (originator) to which a Forward Route is to be installed when forwarding the RREP.

* The address (destination) towards which the RREP is to be sent. More precisely, the destination address indicates the unicast route which the RREP follows.

* The sequence number of the LOADng router, generating the RREP.

Route Reply Acknowledgement (RREP-ACK) - Generated by a LOADng router as a response to a RREP, in order to signal to the neighbor which transmitted the RREP that the RREP was successfully received. Receipt of an RREP-ACK indicates that the link between these two neighboring LOADng routers is bidirectional. A RREP-ACK is unicast to the neighbor from which the RREP has arrived, and is not forwarded. RREP-ACKs are generated only in response to an RREP which, by way of a flag, has explicitly indicated that an RREP-ACK is desired.

Route Error (RERR) - Generated by a LOADng router when a link on an active route to a destination is detected as broken by way of inability to forward a data packet towards that destination. A RERR is unicast to the source of the undeliverable data packet.

5. Protocol Parameters and Constants

The parameters and constants used in this specification are:

LL-LLN-Routers is a link-local-scoped multicast address of a group, which all LOADng routers MUST join.

NET_TRAVERSAL_TIME is the maximum expected time that a packet takes to transverse from one end of the network to the other.

RREQ_RETRIES is the maximum number of subsequent RREQs that a particular router may generate in order to discover a route to a destination, before declaring that destination unreachable.

RREQ_RATELIMIT is the maximum number of RREQ that a particular router is allowed to send per second.

R_HOLD_TIME is the minimum time a Routing Entry SHOULD be kept in the Routing Set after it was last refreshed. This MAY be a network-wide constant, but MAY also be a variable whose value is defined by an auxiliary mechanism, e.g., in an extension to this protocol.

MAX_DIST is the value (tuple) representing the maximum possible distance (R_dist field).

RREP_ACK_TIMEOUT is the minimum time after transmission of a RREP_ACK, that a LOADng router SHOULD wait for a RREP_ACK from a neighbor LOADng router, before considering that the link to this neighbor as unidirectional.

BLACKLIST_TIME is the time during which the link between the neighbor LOADng router and this LOADng router MUST be considered as non-bidirectional, and that therefore RREQs received from that neighbor LOADng router MUST be ignored) after being added. BLACKLIST_TIME should be greater than 2 x NET_TRANSVERSAL_TIME, to ensure that subsequent RREQs will reach the destination via a route excluding this link.

6. Information Base

Each LOADng router maintains an Information Base, containing the information sets necessary for protocol operation, as described in the following sections. The organization of information into these information sets is non-normative, given so as to facilitate description of message generation, forwarding and processing rules in this specification. An implementation may choose any representation or structure for when maintaining this information.

6.1. Routing Set

The Routing Set records the next hop on the route to each known destination, when such a route is known. It consists of Routing Tuples:

(R_dest_addr, R_next_addr, R_dist, R_metric, R_seq_num, R_valid_time)

where:

R_dest_addr - is the address of the destination, either the address of an interface of a destination LOADng router, or the address of an interface reachable via the destination LOADng router, but

which is outside the LLN;

R_next_addr - is the address of the "next hop" on the selected route to the destination;

R_dist - is the distance associated with the selected route to the destination with address R_dest_addr. R_dist is a tuple containing Route Cost, Weak Links and (depending on the metric used) additional fields; see Section 16.

R_metric - specifies how R_dist is defined and calculated, as well as the comparison operator '<=' for determining which of two route costs is lower. This is specified in Section 16;

R_seq_num - is the value of the <seq-num> field of the RREQ or RREP which installed or last updated this tuple. For the routing tuples installed by previous hop information of RREQ or RREP, R_seq_num MUST be set to -1.

R_valid_time - specifies the time until which the information recorded in this tuple is considered valid.

6.2. Blacklisted Neighbor Set

The Blacklisted Neighbor Set records the neighbors of a LOADng router, with which connectivity has been detected to be unidirectional. Specifically, the Blacklisted Neighbor Set records neighbors from which a RREQ has been received (i.e., through which a Forward Route would possible) but to which it has been determined that it is not possible to communicate (i.e., forwarding Route Replies via this neighbor fails, rendering installing the Forward Route impossible). It consists of blacklist tuples:

(B_neighbor_address, B_valid_time)

where:

B_neighbor_address - is the address of the blacklisted neighbor;

B_valid_time - specifies the time until which the information recorded in this tuple is considered valid.

6.3. Destination Address Set

The Destination Address Set records addresses, for which a LOADng router will generate RREPs in response to received RREQs. The Destination Address Set thus represents those destinations (hosts or external networks, or addresses of the LOADng router), for which this LOADng router is providing connectivity. It consists of destination address tuples:

(D_address)

where:

D_address - is the address of a destination (a host or a network), attached to this LOADng router and for which this LOADng router provides connectivity through the LLN.

The Destination Address Set is used for generating signaling, but is not itself updated by signaling specified in this document. Updates to the Destination Address Set are due to changes of the environment of a LOADng router - hosts or external networks being connected to or disconnected from a LOADng router. The Destination Address Set may be administrationally provisioned, or provisioned by external protocols.

6.4. Pending Acknowledgment Set

The Pending Acknowledgements Set contains RREPs which have been transmitted with the ACK_REQUIRED flag set, and for which a RREP-ACK has not yet been received. It consists of Pending Acknowledgment Tuples:

(P_next_hop, P_originator, P_seq_num, P_ack_timeout)

where:

P_next_hop - is the address of the neighbor to which the RREP was sent;

P_originator - is the address of the originator of the RREP;

P_seq_num - corresponds to the <seq-num> field of the sent RREP;

P_ack_timeout - is the time after which the neighbor is considered not to have a bidirectional link to this router and SHOULD be added to the blacklist; the tuple SHOULD then be discarded.

7. LOADng Router Sequence Numbers

Each LOADng router maintains a single sequence number, which must be included in each RREQ or RREP message it generates. Each router MUST make sure that no two messages (both RREQ and RREP) are generated with the same sequence number, and MUST generate sequence number such that these are monotonically increasing. This sequence number is used as freshness information for when comparing routes to the router having generated the message.

However, with a limited number of bits for representing sequence numbers, wrap-around (that the sequence number is incremented from the maximum possible value to zero) will occur. To prevent this from interfering with the operation of the protocol, the following MUST be

observed. The term MAXVALUE designates in the following the largest possible value for a sequence number. The sequence number S1 is said to be "greater than" (denoted '>') the sequence number S2 if:

S2 < S1 AND S1 - S2 <= MAXVALUE/2 OR

S1 < S2 AND S2 - S1 > MAXVALUE/2

8. Packet Format

The packet format, used by this protocol, is described in this section using the notational conventions from [RFC5444]. Example packets are illustrated in Appendix A.

This format uses network byte order (most significant octet first) for all fields. The most significant bit in an octet is numbered bit 0, and the least significant bit of an octet is numbered bit 7 [Stevens].

The general format for all packets, generated, forwarded and processed by this specification, is as follows:

```
    <packet> := <type>
                <tlv-block>
                <message>
```

where:

<type> is a 4 bit unsigned integer field and specifies the type of the <message> field, specified in Section 8.2.

<tlv-block> is specified in Section 8.1.

<message> is specified in Section 8.2.

8.1. TLV Block

The TLV Block contains zero or more Type-Length-Value elements (TLVs). A TLV allows the association of an arbitrary attribute with a packet. The attribute (value) is made up from an integer number of consecutive octets. Different attributes have different types; attributes which are unknown when parsing can be skipped, as specified by flags associated with a given TLV.

```
   <tlv-block> := <tlv-count>
                  (<tlv-type><tlv-flags><tlv-length><tlv-value>)*
```


where:

<tlv-count> is a 4 bit unsigned integer field, specifying the number of TLVs included.

<tlv-type> is a 4 bit unsigned integer field, specifying the type of the TLV.

<tlv-flags> is a 4 bit field specifying processing and forwarding rules related to the TLV processing:

bit 0-3 (RESERVED): SHOULD be set to zero on transmission and SHOULD be ignored upon receipt.

<tlv-length> is an 8 bit unsigned integer field, specifying the length of the following <tlv-value> field.

<tlv-value> is a field of length <length> octets.

8.2. Message Format

This section specifies the format of the <message> field for message types RREQ, RREP,RREP-ACK and RERR.

8.2.1. RREQ and RREP Message Format

The format of Route Request (RREQ) and Route Reply (RREP) messages is identical, RREQ and RREP messages being distinguished by the <type> field in the packet. They are as follows:

```
<message> := <flags>
             <addr-length>
             <seq-num>
             <metric>
             <weak-links>
             <route-cost>
             <destination>
             <originator>
```

where:

<flags> is a 4 bit unsigned integer field and specifies the interpretation of the remainder of the message.

For RREQ messages:

bit 0-3 (RESERVED): SHOULD be set to zero on transmission and SHOULD be ignored upon receipt.

For RREP messages:

bit 0 (ackrequired): When set ('1'), a RREP-ACK MUST be generated by the recipient of an RREP if the RREP is successfully processed. When cleared ('0'), a RREP-ACK MUST NOT be generated in response to processing of the RREP.

bit 1-3 (RESERVED): SHOULD be set to zero on transmission and SHOULD be ignored upon receipt.

<addr-length> is a 4 bit unsigned integer field, encoding the length of the destination and originator addresses (<destination> and <originator>) as follows:

<addr-length> := the length of an address in octets - 1

<addr-length> is thus 1 for 16-bit short addresses [RFC4944], 3 for IPv4 addresses, 7 for 64-bit extended addresses [RFC4944] or 15 for IPv6 addresses.

<seq-num> is a 16 bit unsigned integer field, containing the sequence number (see Section 7) of the LOADng router, generating the RREQ or RREP messages.

<metric> is a 4 bit unsigned integer field and specifies how the value of the <route-cost> field is calculated, as well as the comparison operator '<=' used for when determining which from among two route costs is lower.

<weak-links> is a 4 bit unsigned integer field and specifies the total number of weak links on the route from the originator to the destination.

<route-cost> is an 8 bit unsigned integer field and specifies the cost of the route from the <originator> to the <destination>.

<destination> is an identifier of <address-length> + 1 octets, specifying the address to which the RREQ or RREP should be sent. For a RREQ, this address would be the address for which a route is sought. For a RREP, this address is an unicast address of the router, which originated the RREQ triggering the RREP.

<originator> is an identifier of <address-length> + 1 octets, specifying the address of the router which generated this message, and to which a route is supplied by this message. For a RREQ, the route supplied corresponds to the "reverse route", whereas for a RREP the route supplied corresponds to the "forward route".

8.2.2. RREP-ACK Message Format

The format of a Route Reply Acknowledgement (RREP-ACK) message is as follows:

```
<message> := <flags>
             <addr-length>
             <seq-num>
             <originator>
```

where:

<flags> is a 4 bit unsigned integer field and specifies the interpretation of the remainder of the message:

bit 0-3 (RESERVED): SHOULD be set to zero on transmission and SHOULD be ignored upon receipt.

<addr-length> is a 4 bit unsigned integer field, encoding the length of the destination and originator addresses (<destination> and <originator>) as follows:

<addr-length> := the length of an address in octets - 1

<addr-length> is thus 1 for 16-bit short addresses [RFC4944], 3 for IPv4 addresses, 7 for 64-bit extended addresses [RFC4944] or 15 for IPv6 addresses.

<seq-num> is an 16 bit unsigned integer field and contains the value of the <seq-num> field from the RREP for which this RREP-ACK is sent.

<originator> is an identifier of <address-length> + 1 octets, specifying the address of the originator which has originated the RREP identified by <seq-num>.

8.2.3. RERR Message Format

The format of a Route Error (RERR) message is as follows:

```
<message> := <error-code>
             <addr-length>
             <source>
             <destination>
```

where:

<error-code> is a 4 bit unsigned integer field and specifies the reason for the error message being generated, according to Table 4.

<addr-length> is a 4 bit unsigned integer field, encoding the length of the destination and source addresses (<destination> and <source>) as follows:

<addr-length> := the length of an address in octets - 1

<addr-length> is thus 1 for 16-bit short addresses [RFC4944], 3 for IPv4 addresses, 7 for 64-bit extended addresses [RFC4944] or 15 for IPv6 addresses.

<source> is an identifier of <address-length> + 1 octets, specifying the source address of a data packet, for which delivery to <destination> failed. The LOADng router with this address is the ultimate target for the RERR message.

<destination> is an identifier of <address-length> + 1 octets, specifying the address of the destination, which has become unreachable, and for which an error is reported.

9. Route Maintenance

Entries in the Routing Set are maintained by way of four different mechanisms:

- RREQ/RREP exchange, as described in Section 12 and Section 13.
- Data traffic delivery success.
- Data traffic delivery failure.

- External signals indicating that an entry in the Routing Set necessitates updating.

- Information expiration.

Routing Tuples in the Routing Set contain a validity time, which specifies the time until which the information recorded in this tuple

is considered valid. After this time, the information in such tuples is to be considered as invalid, for the processing specified in this document.

Routing Tuples for actively used routes (i.e., a route via which traffic is currently transiting) SHOULD NOT be removed, unless there is evidence that they no longer provide connectivity - i.e., unless a link on that route has broken.

To this end, one or more of the following mechanisms (non-exhaustive list) MAY be used:

- If a lower layer mechanism provides signals, such as when delivery to a presumed neighbor LOADng router fails, this signal MAY be used to indicate that a link has broken, trigger early expiration of a Routing Tuple from the Routing Set, and to initiate Route Repair (see Section 13) or Route Error Signaling (see Section 14). Conversely, absence of such a signal when attempting delivery MAY be interpreted as validation that the corresponding Routing Tuple(s) are valid, and their R_valid_time refreshed correspondingly.

- Conversely, for each successful delivery of a packet to a presumed neighbor or a destination, if signaled by a lower layer or a transport mechanism, or each positive confirmation of the presence of a neighbor by way of an external neighbor discovery protocol, MAY be interpreted as validation that the corresponding Routing Tuple(s) are valid, and their R_valid_time refreshed correspondingly.

Furthermore, a LOADng router may experience that a route currently used for forwarding data packets is no longer operational, and must act to either rectify this situation locally (Section 13) or signal

this situation to the source of the data packets for which delivery was unsuccessful (Section 14).

10. Unidirectional Links Handling

Each LOADng router MUST monitor the bidirectionality of the links to its neighbors when processing Route Replies (RREP). To this end, one or more of the following mechanisms MAY be used (non exhaustive list):

- If a lower layer mechanism provides signals, such as when delivery to a presumed neighbor LOADng router fails, this signal MAY be used to detect that a link to this neighbor is broken or unidirectional; the LOADng router SHOULD then blacklist the neighbor, see Section 10.1

- If a mechanism such as NDP [RFC4861] is available, the LOADng router MAY use it;

- RREP-ACK message exchange, as described in Section 15

- Upper-layer mechanisms, such as transport-layer acknowledgements, MAY be used to detect unidirectional or broken links.

When a LOADng router detects, via one of these mechanisms, that a link to a LOADng neighbor router is unidirectional or broken, the router SHOULD blacklist this neighbor, see Section 10.1. Conversely, if a LOADng router detects via one of these mechanisms that a previously blacklisted LOADng router has a bidirectional link to this router, it MAY remove it from the blacklist before the <B_valid__time> of the corresponding tuple.

10.1. Blacklist Usage

The Blacklist is maintained according to Section 6.2. When a LOADng router is detected to have a unidirectional link to the LOADng router by means of a chosen mechanism, it is blacklisted, i.e. a tuple (B_neighbor_address, B_valid_time) is added to the list, such as:

- B_neighbor_address is the address of the blacklisted neighbor;

- B_valid_time := current_time + BLACKLIST_TIME

When a LOADng neighbor router is blacklisted, i.e. when there is a corresponding (neighbor_address, B_valid_time) tuple in the Blacklist, it is temporarily not considered as a neighbor, and thus:

- Every RREQ received from this neighbor SHOULD be discarded;
- If the LOADng router needs to establish a route to this neighbor, it SHOULD initiate a new route discovery by generating a RREQ towards the blacklisted neighbor.

11. Common Rules for RREQ and RREP Messages

RREQ and RREP messages, both, supply routes between their recipients and the originator of the RREQ or RREP message. The two message types therefore share common processing rules, and differ only in the following:

- RREQ messages are multicast or broadcast, intended to be received by all LOADng routers in the network, whereas RREP messages are all unicast, intended to be received only by routers on a specific route towards a specific destination.

- Receipt of a RREQ message MAY trigger generation of a RREP message.
- Receipt of a RREP message MAY trigger generation of a RREP-ACK message.

For the purpose of the processing description in this section, the following additional notation is used:

<= is the comparison operator specified by the <metrics> indicated in the RREQ or RREP message and described in Section 16.

previous-hop is the address of the LOADng router, from which the RREQ or RREP message was received.

> is the comparison operator for <seq-num> specified in Section 8.

11.1. Identifying Invalid RREQ or RREP Messages

A received RREQ or RREP message is invalid, and MUST be discarded without further processing, if any of the following conditions are true:

- The address contained in the <originator> field is an address of this router;
- There is a tuple in the Routing Set where:
  - R_dest_addr = <originator>
  - R_seq_num > <seq-num>
- The metrics type contained in the <metrics> field of the RREQ or RREP message is different from the metrics type used by the interface of this router on which the RREQ or RREP message was received, or some TLVs required by the metric type are absent from the received RREQ or RREP message;
- Furthermore, for RREQ messages only, a RREQ SHOULD be considered invalid if the previous-hop is blacklisted (i.e. its address is in a tuple in the Blacklist, see Section 10.1)

A LOADng router MAY recognize additional reasons for identifying that a RREQ or RREP message is invalid for processing, e.g., to allow a security protocol to perform verification of signatures and prevent processing of unverifiable RREQ or RREP message by this protocol.

11.2. RREQ and RREP Message Processing

A received and not invalid RREQ or RREP message is processed as follows:

1. The TLV fields included in the TLV block are added/removed/ updated according to their specification.
2. If the RREQ or RREP message was received over a "weak link", increment the <weak-links> field in the received RREQ or RREP by one.

3. Find the Routing Tuple (henceforth, matching Routing Tuple) where:

   * R_dest_addr = <originator>

4. If no matching Routing Tuple is found, then create a new matching Routing Tuple (the "reverse route" for RREQ messages or "forward route" for RREP messages) with:

   * R_dest_addr := <originator>

   * R_next_addr := previous-hop

   * R_dist := MAX_DIST

   * R_seq_num := -1

   * R_valid_time := current time + R_HOLD_TIME

5. The matching Routing Tuple, existing or new, is compared to the received RREQ or RREP message:

   1. If

      + (<route-cost>, <weak-links>, (<tlv>)*) <= R_dist; AND

      + R_seq_num = <seq-num>

      OR

      + <seq-num> > R_seq_num

      Then:

      + The message is used for updating the Routing Set according to Section 11.3.

      + If there is no matching Routing Tuple in the Routing Set with R_dest_addr = previous-hop, create a new matching Routing Tuple with:

        - R_dest_addr := previous-hop

- R_next_addr := previous-hop

- R_dist := MAX_DIST

- R_seq_num := -1

- R_valid_time = current time + R_HOLD_TIME

2. Otherwise, the RREQ or RREP message is not processed further, and is not considered for forwarding.

11.3. Updating Routing Tuples In Response to RREQ and RREP

A Routing Tuple in the routing set is updated when a received RREQ or RREP message provides a better route to the <originator> than the route current recorded in that Routing Tuple. The Routing set is updated thus:

- R_next_addr := previous-hop

- R_dist := (<route-cost>, <weak-links>, (<tlv>)*); the TLVs used are defined by the <metrics> included in the RREQ or RREP message.

- R_seq_num := <seq-num>

- R_valid_time := current time + R_HOLD_TIME

12. Route Requests (RREQs)

Route Requests (RREQs) are generated by a LOADng router, when it has data packets to deliver to a destination for which it has no matching entry in the Routing Set. The RREQ is transmitted to all directly reachable neighbor LOADng routers.

After originating a RREQ, a LOADng router waits for a corresponding RREP. If no such RREP is received within 2*NET_TRAVERSAL_TIME milliseconds, the LOADng router MAY issue a new RREQ for the sought destination (with an incremented seq_num) up to a maximum of RREQ_RETRIES times. A LOADng router SHOULD NOT originate more than RREQ_RATELIMIT RREQs per second. A LOADng router MAY use mechanisms such as exponential backoff to determine the rate at which it originates RREQs.

12.1. RREQ Generation

A RREQ packet is generated according to Section 8.2 with the following content:

- <type> := RREQ;
- <tlv-block> the TLV block, containing the TLVs needed;
- <addr-length> set to the length of the address, as specified in Section 8;
- <metric> set to indicate how route costs are to be calculated and compared, according to Table 3;
- <weak-links> := 0;
- <seq-num> set to the next unused sequence number, maintained by this router;
- <route-cost> := the cost associated with the interface over which the RREQ is transmitted, and according to the specification of the <metric> included in the RREQ, see Section 16;
- <destination> := the address to which a route is sought;
- <originator> := the address of the LOADng router, generating the RREQ.

12.2. RREQ Processing

On receiving a RREQ message, a LOADng router must process the message according to this section:

1. If the message is invalid for processing, as defined in Section 11.1, the message MUST be discarded without further processing. The message is not considered for forwarding.
2. Otherwise, the message is processed according to Section 11.2.
3. If <destination> in the RREQ message does not correspond to an address of this LOADng router, then the message is considered for forwarding according to Section 12.3.
4. If <destination> in the RREQ message corresponds to an address of this LOADng router, then an RREP can be generated, see Section 13.1. The RREQ is not considered for forwarding.

12.3. RREQ Forwarding

A Route Request (RREQ), considered for forwarding, MUST be updated as follows, prior to it being transmitted:

1. The <route-cost> field must be updated according to the cost associated with the interface over which the RREQ is transmitted, and according to the specification of the <metrics> included in the RREQ, see Section 16.

RREQ forwarding MAY be undertaken using classic flooding, may employ a reduced relay set mechanism such as [SMF] or any other information diffusion mechanism such as [RFC6206]. Care must be taken that NET_TRAVERSAL_TIME is chosen so as to accommodate for the maximum time that may take for a RREQ to transverse the network, accounting for in-router delays incurring due to or imposed by such algorithms.

12.4. RREQ Transmission

RREQs, initially generated or forwarded, are sent to all neighbor LOADng routers. If LOADng is operating as an IP routing protocol, the destination address for this RREQ MUST be the link local multicast address LL-LLN-Routers, and the source address MUST be the address of the interface over which the RREQ is sent.

When a RREQ is transmitted, all receiving LOADng routers will process the RREQ message and MAY consider the RREQ message for forwarding at the same, or at almost the same, time. If using data link and physical layers that are subject to packet loss due to collisions, such RREQ messages SHOULD be jittered as described in [RFC5148].

13. Route Replies (RREPs)

Route Replies (RREPs) are generated by a LOADng router in response to a RREQ where the <destination> corresponds to one of that LOADng router's addresses. RREPs are sent, hop by hop, in unicast towards the originator of the corresponding RREQ, along the Reverse Route installed by that RREQ. A router, upon forwarding a RREP, installs the Forward Route towards the <destination>.

Thus, with forwarding of RREQs installing the Reverse Route and forwarding of RREPs installing the Forward Route, bi-directional

routes are provided between the <originator> and <destination> indicated in the RREQ.

13.1. RREP Generation

At least one RREP MUST be generated in response to a (set of) received RREQ messages with identical (<originator>,<seq-num>). A RREP can be generated immediately as a response to each RREQ processed, or can be generated after a certain delay after the arrival of the first RREQ, in order to use the "best" received RREQ (received over lowest-cost path, over path with least Weak Links etc). A LOADng router MAY generate further RREPs for subsequent RREQs received with the same (<originator>,<seq-num>) pairs, if these indicate a better route. The content of RREP is as follows:

- <type> := RREP;
- <tlv-block> the TLV block;
- <flag> bit-0 set to ('1') if RREP-ACK need to be generated. Otherwise, bit-0 is set to ('0');
- <addr-length> set to the length of the address, as specified in Section 8;
- <seq-num> set to the next unused sequence number, maintained by this LOADng router;
- <metric> set to indicate how route costs are to be calculated and compared, according to Table 3;
- <weak-links> := 0;
- <route-cost> := the cost associated with the interface over which the RREP is transmitted, and according to the specification of the <metric> included in the RREP message, see Section 16;
- <destination> := the address to which this RREP message is to be

sent; this corresponds to the <originator> address from the RREQ message, in response to which this RREP message is generated;

- <originator> := the address of the LOADng router, generating the RREP.

The specification of the TLVs included in the <tlv-block> of the RREQ responsible to generate the RREP MUST stipulate if, and under which conditions, these are to be included in the <tlv-block> of the RREP.

13.2. RREP Processing

On receiving a RREP message, a LOADng router must process the message according to this section:

1. If the message is invalid for processing, as defined in Section 11.1, the message MUST be discarded without further processing. The message is not considered for forwarding.

2. Otherwise, the message is processed according to Section 11.2.

3. If the RREP message has the ACK-REQUIRED flag set, an RREP_ACK message MUST be sent to the previous-hop, according to Section 15.1.

4. If the <destination> in the RREP message does not correspond to an address of this LOADng router, the RREP message is considered for forwarding according to Section 13.3.

13.3. RREP Forwarding

A Route Reply (RREP) message, considered for forwarding, MUST be updated as follows, prior to it being transmitted:

1. The <route-cost> and must be updated according to the cost associated with the interface over which the RREP is transmitted, and according to the specification of the <metric> included in the RREP, see Section 16.

2. If this interface of the LOADng router uses RREP-ACKs to check the bidirectionality of the links, the ACK_REQUIRED flag MUST be set to 1, see Section 15.

The RREP message is then unicast to the next hop towards the <destination> indicated in the RREP.

13.4. RREP Transmission

A Route Reply (RREP) is, ultimately, destined for the LOADng router listed in the <destination> field, and is forwarded in unicast towards this LOADng router. The RREP MUST, however, be transmitted so as to allow it to be processed in each intermediate LOADng router to:

- install proper forward routes
- permit that <route-cost> and <weak-links> be updated to reflect the route, and

- permit that TLVs included may be processed/added/removed according to their specification.

14. Route Errors (RERRs)

If Route Repair is not successful, or if Route Repair is not attempted, a LOADng router MUST generate a Route Error (RERR), and send this RERR along the Reverse Route to the source of the data packet for which delivery was unsuccessful.

14.1. RERR Generation

A RERR packet is generated by the LOADng router, detecting the link breakage, with the following content:

- <type> := RERR;
- <tlv-block> the TLV block, containing the TLVs needed;
- <error-code> := the most appropriate error code from among those recorded in Table 4;

- <addr-length> := the length of the address, as specified in Section 8;

- <source> := the source address from the unsuccessfully delivered data packet.

- <destination> := the destination address from the unsuccessfully delivered data packet.

14.2. RERR Processing

For the purpose of the processing description below, the following additional notation is used:

previous-hop is the address of the LOADng router, from which the RERR was received.

Upon receiving an RERR, a LOADng router MUST update its Routing Set as follows:

1. The TLV fields are added/removed/updated according to their specification.

2. Find the Routing Tuple (henceforth "matching Routing Tuple") in the Routing Set where:

   * R_dest_addr = <destination>

   * R_next_addr = previous-hop

3. If no matching Routing Tuple is found, the RERR is not processed further, and is not considered for forwarding.

4. Otherwise, if one matching Routing Tuple is found, this matching Routing Tuple is updated as follows:

   * R_valid_time := expired

   The RERR message is, then, considered for forwarding.

14.3. RERR Forwarding

A RERR is, ultimately, destined for the LOADng router which has the address from the <source> field, in its Destination Address Set.

A RERR, considered for forwarding is therefore processed as follows:

1. Find the Destination Address Tuple (henceforth, matching Destination Address Tuple) in the Destination Address Set where:

   * D_address = the address from the <source> field of the RERR

2. If one or more matching Destination Address Tuples is found, the RERR message is discarded and not retransmitted.

3. Otherwise, if no matching Destination Address Tuples is found, the RERR message is transmitted according to Section 14.4.

14.4. RERR Transmission

An RERR is transmitted, as unicast, to LOADng router, recorded the next-hop for the <source> indicated in the RERR message. The RERR MUST be transmitted hop-by-hop such that it can be processed in each intermediate LOADng router. This serves to:

- Allow intermediate routers to update their Routing Sets, i.e., remove entries for this destination.

- Permit that TLVs included may be processed/added/removed according to their specification.

15. Route Reply Acknowledgements (RREP-ACKs)

A LOADng router SHOULD use RREP-ACK exchange to monitor bidirectionality of links with neighbor routers, except if another mechanism, as described in Section 10, provides for such bidirectionality information.

A LOADng router MUST signal in a transmitted RREP that it is expecting a RREP-ACK, by setting the ackrequired flag in the RREP. When doing so, the LOADng router MUST also add a tuple (P_next_hop, P_originator, P_seq_num, P_ack_timeout) to the Pending Acknowledgement Set, and set P_ack_timeout to RREP_ACK_TIMEOUT.

15.1. RREP-ACK Generation

Upon reception of a RREP message with the <ack-required> flag set, a LOADng router MUST generate a RREP-ACK and send this RREP-ACK in unicast to the neighbor which originated the RREP.

A RREP-ACK packet is generated by a LOADng router with the following content:

- <type> := RREP-ACK;
- <tlv-block> the TLV block, containing the TLVs needed;
- <addr-length> := the length of the address, as specified in Section 8;
- <seq-num> := the <seq-num> field of the received RREP;
- <originator> := the <originator> field of the received RREP.

15.2. RREP-ACK Processing

On receiving a RREP-ACK from a LOADng neighbor router, a LOADng router SHOULD do the following:

1. The TLV fields are added/removed/updated according to their specification.
2. Check whether a corresponding RREP is pending, i.e. if the Pending List contains a tuple (next_hop, originator, seq_num, ack_timeout) such as:
    * next_hop is the address of the LOADng neighbor router from which the RREP-ACK was received;

* originator corresponds to the <originator> field of the RREP-ACK;

* seq_num corresponds to the <seq-num> field of the RREP-ACK.

3. If such a tuple exists, then the RREP has been correctly acknowledged and the tuple MUST be discarded.

4. Otherwise, i.e. if no such tuple exists, then no further processing is required.

15.3. RREP-ACK Forwarding

A RREP-ACK is intended only for a specific direct neighbor, and MUST NOT be forwarded.

15.4. RREP-ACK Transmission

A RREP-ACK is transmitted, in unicast, to the neighbor LOADng router from which the RREP was received.

16. Metrics

This specification permits the use of different metrics for when calculating route costs, and specifies one particularly simplified such metric in Section 16.3 purely intended as an example - while encouraging that more appropriate metrics be developed for different deployment environments.

16.1. The Default <= Comparison Operator

The objective of the <= comparison operator is to be able to determine which of two routes is "best", i.e., which route has the lowest cost. A link between a pair of interfaces may have a nominal and administratively assigned cost associated (such as, for example, representing a nominal bandwidth), however may also have a dynamic component making an link with an otherwise low cost a less attractive choice - a "Weak Link" - for when establishing a new route (such as, for example, if a high loss-rate is experienced across that link). This may make a longer (in term of cost) route preferable over a shorter route involving such "Weak Links".

To accommodate this situation, this specification includes in RREQs and RREPs, both a <route-cost> element, representing the cost of the route traveled, and a <weak-links> element, counting the number of weak links encountered. When a destination receives multiple copies of the same RREQ, via different routes, the default <= comparison operator is defined so as to prefer routes with fewer weak links,

even if such a route has an absolute higher route cost.

Let (WL, RC) be the pair (weak-links, route-cost) received in one RREQ, and let (WL', RC') be the pair (weak-links, route-cost) received in another RREQ. The comparison operator <= is then defined as:

(WL,RC) <= (WL',RC') if and only if:

WL < WL'; OR
WL == WL' AND RC <= RC'

16.2. Specifying New Metrics

When defining a metric, the following considerations SHOULD be taken into consideration, and MUST be taken into consideration when requesting a code-point from IANA for the 1-64 range of the Cost Types registry defined in Table 3:

- The definition of the R_dist field, as well as the value of MAX_DIST.
- The mechanism for determining when a link qualifies as a "Weak Link". Examples include when an SNR or SIR is above/below a given threshold, etc. This MAY be by way of lower-layer information, message statistics or any other means.
- The mechanism for determining how to update the <route-cost> field when a RREP or RREQ is transmitted over an interface.
- The required TLV fields for R_dist, as well as the mechanism for determining how to update those fields.
- The <= comparison operator. This MAY be by way of indicating that the definition in Section 16.1 is used, or an operator MAY be specified using also, e.g., information contained in TLVs; in either case, the comparison operator to use MUST be specified. Furthermore, the comparison operator MUST specify a strict ordering of the R_dist space, i.e. R_dist1 can always be compared to R_dist2 and (R_dist1 <= R_dist2 && R_dist1 > R_dist2) if and only if R_dist1 = R_dist2.

16.3. Example Metric: Hop Count

This section is intended to exemplify of how to specify Metrics. It represents a simple "hop count" based cost. It is RECOMMENDED to define a more appropriate metric for the environment in which the protocol is to operate.

16.3.1. Simple Hop Count

R_dist: R_dist := (RC, WL) where RC is the Route Cost and WL the number of weak links. MAX_DIST := (255, 15).

Weak Link: No link is ever considered as a Weak Link. Consequently, when generating a RREQ or RREP, the <weak-link> element is set to zero, the <weak-link> is never incremented when forwarding.

Route Cost: When generating a RREQ or a RREP, the <route-cost> is initialized to one, the <route-cost> is incremented by one when forwarding.

TLVs: No additional TLVs are used.

<= : (WL,RC) <= (WL',RC') if and only if: RC < RC' OR (RC = RC' AND RC <= RC')

17. Security Considerations

Currently, this document does not specify any specific security measures. By way of enabling inclusion of TLVs, development of security measures, appropriate for a given deployment, is however supported.

18. IANA Considerations

18.1. Multicast Addresses

IANA is requested to allocate LL-LLN-ROUTERS well-known, link-scoped multicast addresses for both IPv4 and IPv6.

18.2. Packet Types

IANA is requested to create a new registry for packet types, with initial assignments and allocation policies as specified in Table 1.

Type	Description	Allocation Policy
0	Route Request (RREQ)	
1	Route Reply (RREP)	
2	Route Error (RERR)	
3	Route Reply Acknowledgement (RREP-ACK)	
4-64	Unassigned	Expert Review
65-127	Unassigned	Experimental Use

Table 1: Packet Types

18.3. TLV Types

IANA is requested to create a new registry for TLV types, with initial assignments and allocation policies as specified in Table 2.

Type	Description	Allocation Policy
0-64	Unassigned	Expert Review
65-127	Unassigned	Experimental Use

Table 2: TLV Types

18.4. Metrics

IANA is requested to create a new registry for Metrics, with initial assignments and allocation policies as specified in Table 3.

Code	Description	Allocation Policy
0	Hop Count While Avoiding Weak Links (Section 16)	
1-64	Unassigned	Expert Review
65-127	Unassigned	Experimental

		Use

Table 3: Metrics

When assigning a new Metric, the specification requesting that assignment MUST specify the way in which each LOADng router calculates the <route-cost> field in RREQs and RREPs, as well as the criteria for incrementing the <weak-links> field in RREQs and RREPs. The specification MUST also specify the comparison operations '<=' for determining from among two RREQs (or RREPs) for the same destination represents the shortest route; note that this comparison operation SHOULD involve the <route-cost> field and MAY use other information such as <weak-links> or content of specific TLV types included in the RREQ or RREP.

18.5. Error Codes

IANA is requested to create a new registry for Error Codes, with initial assignments and allocation policies as specified in Table 4.

Code	Description	Allocation Policy
0	No available route	
1-64	Unassigned	Expert Review
65-127	Unassigned	Experimental Use

Table 4: Error Codes

19. Contributors

This specification is the result of the joint efforts of the following contributors -- listed alphabetically.

o Alberto Camacho, LIX, France, <alberto@albertocamacho.com>

- Thomas Heide Clausen, LIX, France, <T.Clausen@computer.org>

- Axel Colin de Verdiere, LIX, France, <axel@axelcdv.com>

- Kenneth Garey, Maxim Integrated Products, USA, <kenneth.garey@maxim-ic.com>

- Ulrich Herberg, Fujitsu Laboratories of America, USA <ulrich.herberg@us.fujitsu.com>

- Yuichi Igarashi, Hitachi Ltd, Yokohama Research Laboratory, Japan, <yuichi.igarashi.hb@hitachi.com>

- Afshin Niktash, Maxim Integrated Products, USA, <afshin.niktash@maxim-ic.com>

- Hiroki Satoh, Hitachi Ltd, Yokohama Research Laboratory, Japan, <hiroki.satoh.yj@hitachi.com>

- Jiazi Yi, LIX, France, <jiazi@jiaziyi.com>

20. Acknowledgments

The authors would like to acknowledge the team behind AODV, specified in RFC3561 for their contributions. The authors would also like to acknowledge the efforts of K. Kim (picosNet Corp/Ajou University), S.

Daniel Park (Samsung Electronics), G. Montenegro (Microsoft Corporation), S. Yoo (Ajou University) and N. Kushalnagar (Intel Corp.) for their work on an initial version of a specification, from which this protocol is derived.

21. References

21.1. Normative References

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

[RFC5444] Clausen, T., Dean, J., Dearlove, C., and C. Adjih, "Generalized Mobile Ad Hoc Network (MANET) Packet/Message Format", RFC 5444, February 2009.

21.2. Informative References

[RFC3561] Perkins, C., Belding-Royer, E., and S. Das, "Ad hoc On-Demand Distance Vector (AODV) Routing", RFC 3561, July 2003.

[RFC4944] Montenegro, G., Kushalnagar, N., Hui, J., and D. Culler, "Transmission of IPv6 Packets over IEEE 802.15.4 Networks", RFC 4944, September 2007.

[RFC6206] Levis, P., Clausen, T., Gnawali, O., and J. Ko, "The Trickle Algorithm", RFC 6206, March 2011.

[SMF] Macker, J., "Simplified Multicast Forwarding", draft-ietf-manet-smf-12 (work in progress), July 2011.

[RFC5148] Clausen, T., Dearlove, C., and B. Adamson, "Jitter Considerations in Mobile Ad Hoc Networks (MANETs)", RFC 5148, February 2008.

[RFC4861] Narten, T., Nordmark, E., Simpson, W., and H. Soliman, "Neighbor Discovery for IP version 6 (IPv6)", RFC 4861, September 2007.

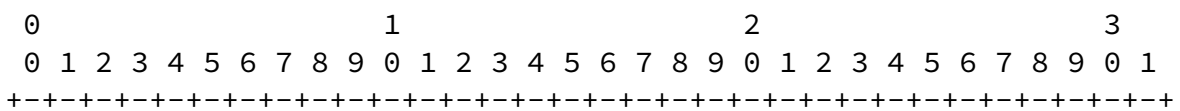
[Stevens] Stevens, W., "TCP/IP Illustrated Volume 1 - The Protocols", 1994.

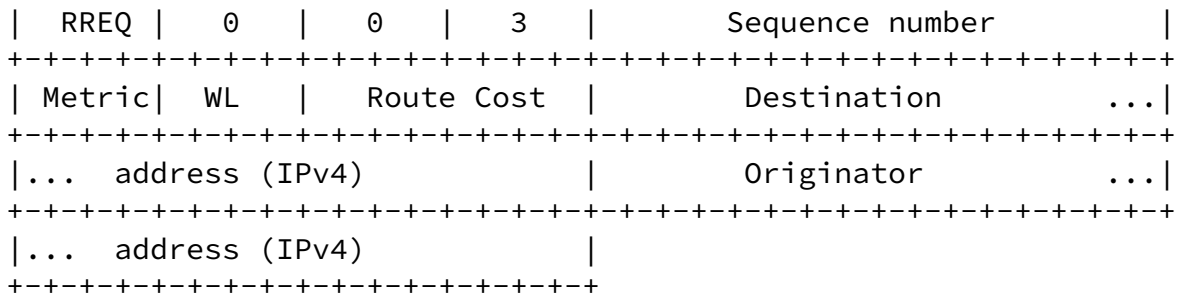
Appendix A. LOADng Control Packet Illustrations

This section presents example packets following these specifications.

A.1. RREQ

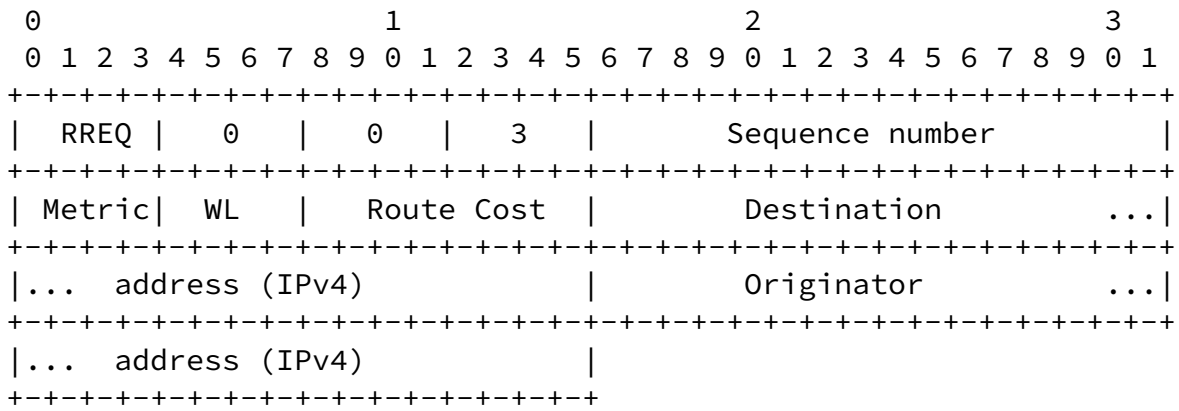
This figure describes the format of a sample RREQ packet using 32 bits IPv4 addresses. The packet is as follows:





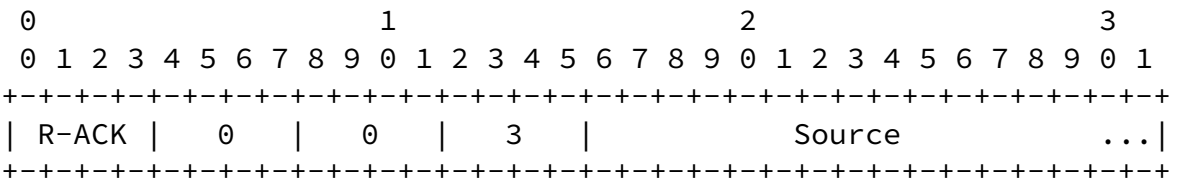
A.2. RREP

This figure describes the format of a sample RREP packet using 32 bits IPv4 addresses. The packet is as follows:



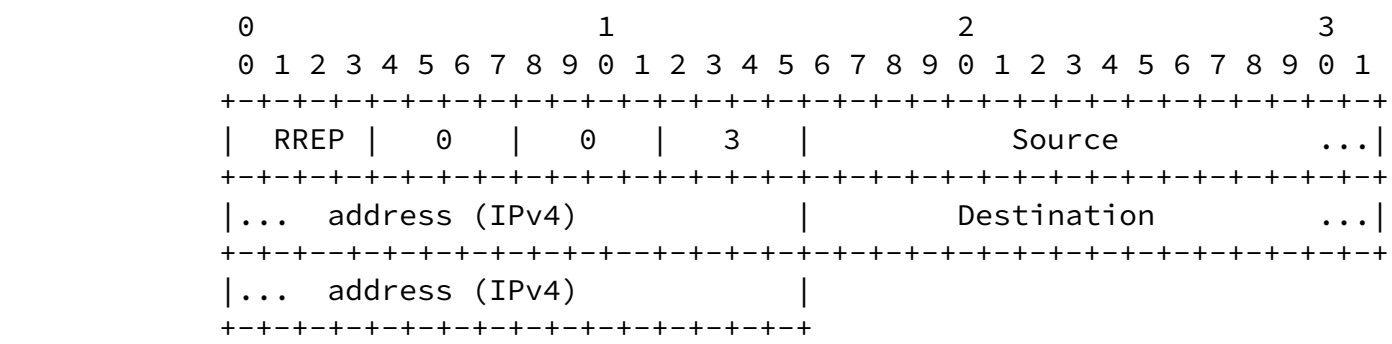
A.3. RREP-ACK

This figure describes the format of a sample RREP-ACK packet using 32 bits IPv4 addresses, as follows:



A.4. RERR

This figure describes the format of a sample RERR packet using 32 bits IPv4 addresses, as follows:



Authors' Addresses

Thomas Heide Clausen
LIX, Ecole Polytechnique

Phone: +33 6 6058 9349
EMail: T.Clausen@computer.org
URI: http://www.ThomasClausen.org/

Axel Colin de Verdiere
LIX, Ecole Polytechnique

Phone: +33 6 1264 7119
EMail: axel@axelcdv.com
URI: http://www.axelcdv.com/

Jiazi Yi
LIX, Ecole Polytechnique

Phone: +33 1 6933 4031
EMail: jiazi@jiaziyi.com
URI: http://www.jiaziyi.com/

Afshin Niktash
Maxim Integrated Products

Phone: +1 94 9450 1692
EMail: afshin.niktash@maxim-ic.com
URI: http://www.Maxim-ic.com/

Yuichi Igarashi
Hitachi, Ltd., Yokohama Research Laboratory

Phone: +81 45 860 3083
EMail: yuichi.igarashi.hb@hitachi.com
URI: http://www.hitachi.com/

Hiroki Satoh
Hitachi, Ltd., Yokohama Research Laboratory

Phone: +81 44 959 0205
EMail: hiroki.satoh.yj@hitachi.com
URI: http://www.hitachi.com/

Ulrich Herberg
Fujitsu Laboratories of America

Phone: +1 408 530 4528
EMail: ulrich@herberg.name
URI: http://www.herberg.name/