The LLN On-demand Ad hoc Distance-vector Routing Protocol - Next Generation (LOADng)

Thomas Heide Clausen, Axel Colin de Verdière

▶ **To cite this version:**

Thomas Heide Clausen, Axel Colin de Verdière. The LLN On-demand Ad hoc Distance-vector Routing Protocol - Next Generation (LOADng). [Research Report] RR-7692, INRIA. 2011. inria-00611181

HAL Id: inria-00611181
https://hal.inria.fr/inria-00611181

Submitted on 25 Jul 2011

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

The LLN On-demand Ad hoc Distance-vector Routing Protocol - Next Generation (LOADng)

Thomas Heide Clausen, Axel Colin de Verdiere

N° 7692

July 2011

ISSN 0249-6399 ISRN INRIA/RR--7692--FR+ENG

INSTITUT NATIONAL
DE RECHERCHE
EN INFORMATIQUE
ET EN AUTOMATIQUE

INRIA

centre de recherche
SACLAY - ÎLE-DE-FRANCE

The LLN On-demand Ad hoc Distance-vector Routing Protocol - Next Generation (LOADng)

Thomas Heide Clausen, Axel Colin de Verdiere

Thème : COM – Systèmes communicants
Équipe-Projet Hipercom

Rapport de recherche n° 7692 — July 2011 — 31 pages

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

Key-words: Routing, Sensor Networks, Low Power, Lossy, LOAD

Centre de recherche INRIA Saclay – Île-de-France
Parc Orsay Université
4, rue Jacques Monod, 91893 ORSAY Cedex
Téléphone : +33 1 72 92 59 00

The LLN On-demand Ad hoc Distance-vector Routing Protocol - Next Generation (LOADng)

Résumé : Ce document décrit le protocole de routage "distance vector" LOAD - Next Generation (LLN Ad hoc On-Demand). Il s'agit d'un protocole réactif qui s'adresse aux réseaux à faible puissance et fort taux de perte (Low power and Lossy Networks, LLNs). Il est dérivé d'AODV et a été modifié de manière répondre aux besoin de tels réseaux.

Mots-clés : Routing, Sensor Networks, Low Power, Lossy, LOAD

1 Introduction

The LLN On-demand Ad hoc Distance-vector Routing Protocol - Next Generation (LOADng) is a routing protocol, derived from AODV [7] and extended for use in Low power Lossy Networks (LLNs). A reactive protocol, the basic operations of LOAD 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 destination, and 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 path 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 [7], LOADng is simplified as follows:

- Only the destination is permitted to respond to a RREQ; intermediate routers are explicitly prohibited to respond to RREQs, even if they may have active routes to the destination. This eliminates the need for Destination Sequence Numbers, while retaining loop freedom. This also eliminates Gratuitous RREPs. The rationale for this simplification is, that it simplifies protocol operation and reduces message size. Also, except for in cases where the network graph has articulation points, this does not reduce the control traffic overhead incurred.
- A LOADng router does not maintain a precursor list, thus when forwarding of a data packet to the recorded next hop on the path 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 paths are in use concurrently.

Compared to [7], 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 in a given LLN each device has a unique address, and that all addresses within an LLN are of the same address length.
- Control messages can include a set of TLV (Type-Length-Value) elements, permitting flexible 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 [1].

Additionally, this memorandum uses the following terminology:

- A router which implements this routing protocol.
- The address of a router or host, to which a route is sought discovered and maintained.
- The address of a router, which seeks to discover and maintain a route to a Destination.
- A route set up so as to send data packets from the Originator to the Destination. The Forward Route is set up when a LOADng Router forwards Route Reply (RREP) messages.
- A route set up so as to send data packets from the Destination to the Originator. The Reverse Route is set up when a LOADng Router forwards Route Request (RREQ) messages. It is used for forwarding RREP messages, as well as for forwarding data packets.
- The cost (weight) between a pair of LOADng Routers, determined by a LOADng router upon reciept of a packet.
- The sum of the Link Costs for the links that a RREQ or RREP has crossed.
- A link which is marginally usable, *i.e.,*MAY be used if no other links are available, but SHOULD be avoided if at all possible - even if it entails an ultimately longer path. As an example, a Weak Link might be defined as a link with a significant loss-rate.

This memorandum employs the same notational conventions as in [3].

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 paths 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-path maintenance; if a destination becomes unreachable, rediscovery of that single (bi-directional) path 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:

- Be able to discover a bi-directional path to any destination in the network.
- Establish paths only when there is data traffic to be sent along that path.
- Maintain paths only for as long as there is traffic using the path, *i.e.,*maintain router state only for paths in active use.
- Generate control traffic based on network events only: when a new path is required, or when an active path is detected broken. Specifically, this protocol does not require periodic signaling.

4.1 Overview

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

- A LOADng router with a data packet to deliver to a destination, for which no entry in the routing table exists, generates a Route Request (RREQ) encoding the destination address, and transmits this to all of its neighbors.
- Upon receiving an RREQ, a LOADng router will install or refresh an entry in the routing table towards the originator of the RREQ, as well as to the neighbor LOADng router from which the RREQ was received. This will install the Reverse Route.
- Upon receiving a RREQ, a LOADng router will also inspect the indicated destination address:
  - If that address is an address of that LOADng router, it will 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 of the LOADng router, it will consider the RREQ as a candidate for forwarding.
- A RREQ considered for forwarding is retransmitted according to the forwarding operation specified for the network. This may be simply classic flooding, or may be an optimized flooding mechanism.
- Upon receiving a RREP, a LOADng router will install a forward route towards the originator of that RREP, as well as to the neighbor LOADng router, from which that RREP was received. This will install the Forward Route.
- Upon receiving a RREP, a LOADng router will also forward it, as unicast, to the recorded next hop along the recorded Reverse Route.

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 network 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 and RERR messages.

4.3 Information Base Overview

Necessary protocol state is recorded by way of two information sets: the "Routing Set", and the "Route Request Set". The Routing Set contains tuples, representing next-hop and distance towards a destination so as to enable routing. The Route Request Set tracks route discovery processes in progress, so as to enable a LOADng router to eliminate multiple forwards of RREQs.

4.4 Signaling Overview

This protocol generates and processes the following protocol packets:

- 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 path to that destination. A RREQ is flooded through the network, possibly employing an optimized flooding mechanism, and contains the address of the sought destination as well as the address of the LOADng router, originating the RREQ. When forwarding a RREQ, a LOADng router's Routing Set is updated so as to reflect the path towards the originator of the RREQ, and the Route Request Set is updated so as to allow suppression of subsequent copies of the same RREQ.
- Generated as a response to a RREQ by the destination sought, *i.e.,*by the LOADng router using or responsible for the address contained in the RREQ. A RREP is unicast to the originator of the RREQ in a hop-by-hop fashion. When forwarding a RREP, a LOADng router's Routing Set is updated so as to reflect the path towards the originator of the RREP.

5 Protocol Parameters and Constants

The parameters and constants used in this specification are:

LL-LLN-Routers
NET_TRAVERSAL_TIME
RREQ_RETRIES
RREQ_RATELIMIT
R_hold_time
D_hold_time

6 Information Base

Each LOADng router maintains an Information Base, containing several information sets, as described in the following sections. These information sets are given so as to facilitate description of message generation, forwarding and processing rules. In particular, an implementation may chose any representation or structure for when maintaining this information.

6.1 Routing Set

A LOADng router's Routing Set records the next hop along a path to each destination, for which such a path is known. It consists of Routing Tuples:

```
(R_dest_addr, R_next_addr, R_dist, R_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 path to the destination;
- R_dist is the cost of the selected path to the destination with address R_dest_addr;
- R_time specifies when this Tuple expires and MUST be removed.

6.2 Route Request Set

A LOADng router's Route Request Set records information about issued route discovery processes. It consists of Route Request Tuples:

```
(D_src_addr, D_rreq_id, D_time)
```

where:

- D_src_addr is the address of the LOADng router, generating a RREQ and thus initiating the route discovery process;
- D_rreq_id is a sequence number, uniquely identifying each RREQ issued by a given LOADng router. The pair (D_src_addr, D_rreq_id) uniquely identifies a RREQ in the network;
- D_time specifies when this Tuple expires and MUST be removed.

7 Packet Format

The packet format, used by this protocol, is described using the notational conventions from [3]. Example packets are illustrated in A.

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 7.2.
- <tlv-block> is specified in 7.1.
- <message> is specified in 7.2.

7.1 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 the interpretation of the remainder of the TLV:
  - bits 0-3: RESERVED
- <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.

7.2 Message Format

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

7.2.1 RREQ and RREP Message Format

The format of Route Request (RREQ) and Route Reply (RREP) messages is identical, as follows:

```
<message> := <flags>
             <addr-length>
             <cost-type>
             <weak-links>
             <rreq-ID>
             <route-cost>
             <destination>
             <originator>
```

where:

- <flags>is a 4 bit unsigned integer field and specifies the interpretation of the remainder of the message:
  - RESERVED
- <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 [6], 3 for IPv4 addresses, 7 for 64-bit extended addresses [6] or 15 for IPv6 addresses.
- address-length is a variable whose value is the length of an address in octets, and is calculated as follows:
  - address-length = <addr-length> + 1
- <cost-type> 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 '<=' for determining which of two route costs is lower.
- <weak-links> is a 4 bit unsigned integer field and specifies the total number of weak links on the routing path from the originator to the destination.
- <rreq-ID> is an 8 bit unsigned integer field and specifies a sequence number uniquely identifying the particular RREQ when taken in conjunction with the originator.
- <route-cost> is an 8 bit unsigned integer field and specifies the cost of the routing path from the originator to the destination.
- <destination> is an identifier with length equal to address-length, specifying the address of the destination, to which a route is sought.
- <originator> is an identifier with length equal to address-length, specifying the address of the originator, which has initiated route discovery for the destination.

7.2.2 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 4.
- <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 [6], 3 for IPv4 addresses, 7 for 64-bit extended addresses [6] or 15 for IPv6 addresses.
- is a variable whose value is the length of an address in octets, and is calculated as follows:
  - address-length = <addr-length> + 1
- <source> is an identifier with length equal to address-length, 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 with length equal to address-length, specifying the address of the destination, which has become unreachable and for which an error is reported.

8 Route Requests

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

Each RREQ, generated by a LOADng router, includes a Route Request ID (RREQID). The RREQID MUST be unique for each RREQ generated by a router, and RREQIDs for a LOADng router MUST be generated so as to be monotonically increasing.

After originating a RREQ, a LOADng router waits for a corresponding RREP. If no such RREP is received within NET_TRAVERSAL_TIME milliseconds, the LOADng router MAY issue a new RREQ for the sought destination (with an incremented RREQID) up to a maximum of RREQ_RETRIES times. A LOADng router SHOULD NOT originate more than RREQ_RATELIMIT RREQs per second.

8.1 RREQ Generation

A RREQ message is generated with:

- <addr-length> set to the length of the address, as specified in 7.
- <cost-type> set to indicate how path costs are to be calculated and compared, according to 3.
- <weak-links> set to 0.
- <rreq-ID> set to the next unused RREQID.
- <route-cost> set to the cost associated with the interface over which the RREQ is transmitted, and according to the specification of the <cost-type> included in the RREQ, see 11.
- <destination> set to the address to which a path is sought.
- <originator> set to the address of the LOADng router, generating the RREQ.

8.2 RREQ Processing

On receiving a RREQ message, a LOADng router MUST first check if the message is invalid for processing by this LOADng router, as defined in 8.2.1 and, if so, discard the message without further processing. Otherwise, for each received and valid RREQ, the LOADng router MUST process it according to this section.

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

- is the comparison operator, specified for the <cost-type> indicated in the RREQ and described in 11.
- is the address of the LOADng router, from which the RREQ was received.

The RREQ message is processed as follows:

1. If the RREQ was received over a "weak link", increment the <weak-links> field in the received RREQ by one.
2. Find the Route Request Tuple (henceforth "matching Route Request Tuple") in the Route Request Set where:
   - D_src_addr = originator; AND
   - D_rreq_id = rreq-id
3. If there is no matching Route Request Tuple, then:
   (a) Create new matching Route Request Tuple with:
      - D_src_addr := originator
      - D_rreq_id := rreq-id
      - D_time = current time + D_hold_time
4. Find the Routing Tuple (henceforth "matching Routing Tuple") in the Routing Set where:
   - R_dest_addr = <originator>
5. If there is no matching Routing Tuple, then:
   (a) Create a new matching Routing Tuple (the "reverse route") with:
      - R_dest_addr = <originator>
      - R_next_addr = previous-hop
      - R_dist = infinity
      - R_time = current time + R_hold_time
6. The matching Routing Tuple, existing or new, is compared to the received RREQ:
   (a) If R_dist <= <route-cost> (using the comparison operator, defined for the <cost-type> specified in the RREQ) then the reverse route currently recorded in the Routing Set is better than the route possible via the RREQ. The RREQ is not processed further, and is not considered for forwarding.
   (b) Otherwise, the reverse route possible via the RREQ is better than the reverse route current recorded in the Routing Set:
      i. If for the matching Route Request Tuple:
         - D_rreq_id is greater than <rreq-id>

      then this RREQ represents a path, older than the one recorded. The RREQ is not processed further, and is not considered for forwarding.
      ii. Otherwise:

A. Update the matching Routing Tuple thus:

   R_next_addr = previous-hop
   R_dist = <route-cost>
   R_time = current time + R_hold_time

B. TLVs, included in the TLV block, are processed according to their specification.
C. If <destination> corresponds to an address of this LOADng router, then an RREP is generated, see 9.1.
D. Otherwise, if <destination> does not correspond to an address of this LOADng router, then the RREQ is considered for forwarding, see 8.3.

8.2.1 Invalid RREQ Messages

A received RREQ 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;
- The RREQID, contained in the <rreq-id> field is less than a RREQID previously recorded for the address contained in the <originator> field.

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

8.3 RREQ Forwarding

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

- 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 <cost-type> included in the RREQ, see 11.

RREQ forwarding MAY be undertaken using classic flooding, may employ a reduced relay set mechanism such as [5] or any other information diffusion mechanism such as [4]. Care must be taken that NET_TRAVERSAL_TIME is chosen so as to accommodate for the worst-case time that it takes a RREQ to transverse the net, accounting for in-router delays incurring due to or imposed by such algorithms.

8.4 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 [2].

9 Route Replies

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

9.1 RREP Generation

A RREP message is generated by making a copy of the RREQ message, in response to which the RREP is generated, then modifying that copy as follows:

- <type> := RREP
- <weak-links> := 0
- <route-cost> := the cost associated with the interface over which the RREP is transmitted, and according to the specification of the <cost-type> included in the RREP, see 11.

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

9.2 RREP Processing

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

- is the comparison operator, specified for the <cost-type> indicated in the RREP and described in 11.
- is the address of the LOADng router, from which the RREP was received.

On receiving a RREP, a LOADng router MUST:

- Find the Route Request Tuple (henceforth "matching Route Request Tuple") in the Route Request Set where:
  - D_src_addr = originator; AND
  - D_rreq_id = rreq-id
- If there is no matching Route Request Tuple, then the RREP MUST be discarded, *i.e.*,it MUST NOT be processed further and it MUST NOT be considered for forwarding.
- Otherwise, if there is a matching Route Request Tuple:

- Find the Routing Tuple (henceforth "matching Routing Tuple") in the Routing Set where:
  - R_dest_addr = <destination>
- If there is no matching Routing Tuple, then:
  - Create a new matching Routing Tuple (the "forward route") with:
    - R_dest_addr = <destination>
    - R_next_addr = previous-hop
    - R_dist = infinity
    - R_time = current time + R_hold_time
- The matching Routing Tuple, existing or new, is compared to the received RREP:
  - If R_dist <= <route-cost> (using the comparison operator, defined for the <cost-type> specified in the RREP) then the forward route currently recorded in the Routing Set is better than the forward route possible via the RREP. The RREP is not processed further, and is not considered for forwarding.
  - Otherwise, the forward route possible via the RREP is better than the forward route currently recorded in the Routing Set:
    - If for the matching Route Request Tuple:
      D_rreq_id is greater than <rreq-id>
      then this RREP represents a path, older than the one recorded. The RREP is not processed further, and is not considered for forwarding.
    - Otherwise, update the matching Routing Tuple thus:
      R_next_addr = previous-hop
      R_dist = <route-cost>
      R_time = current time + R_hold_time

      TLVs, included in the TLV block, are processed according to their specification.
      The RREP is considered for forwarding, see 9.3.

9.3 RREP Forwarding

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

- The <route-cost> field must be updated according to the cost associated with the interface over which the RREP is transmitted, and according to the specification of the <cost-type> included in the RREP, see 11.

9.4 RREP Transmission

A Route Reply (RREP) is, ultimately, destined for the LOADng router listed in the <originator> 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 path, and
- permit that TLVs included may be processed/added/removed according to their specification.

10 Route Maintenance

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

- RREQ/RREP exchange, as described in 8 and 9;
- Data traffic delivery success;
- Data traffic delivery failure;
- External signals indicating that an entry in the Routing Set necessitates updating.

The latter three are detailed in this section.

Routing Tuples in the Routing Set contain an expiration time, after which such tuples are to be removed (or, considered as removed) so as to both be conservative in routing state required, and so as to ensure that stale tuples (*i.e.,*tuples which do no longer reflect a topology, recently verified to be current) disappear.

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:

- For very short, predictable, flows, the initial R_hold_time MAY be set to large enough so as to not require refresh of the route for the duration of the flow. An extension MAY, for example, convey this duration in a TLV, as well as specify appropriate processing of this TLV when processing a RREQ/RREP.
- 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 10.1) or Route Error Signaling (see 10.2). 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_hold_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_hold_time refreshed correspondingly.

Regardless, a LOADng router may experience that a path, currently used for forwarding data packets, is no longer operational, and must act to either rectify this situation locally (10.1) or signal this situation to the source of the data packets for which delivery was unsuccessful (10.2).

10.1 Route Repair

If a link breakage is detected, an intermediate LOADng router MAY attempt to repair the route locally, by issuing a RREQ and awaiting an RREP. This RREQ is generated with the intermediate LOADng router's address as <originator> and is generated, processed and forwarded exactly as specified in 8 and 9, and with the same constraints on RREQ generation.

10.2 Route Errors (RERR)

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 path to the source of the data packet for which delivery was unsuccessful.

10.2.1 RERR Generation

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

- <error-code> = the most appropriate error code from among those recorded in 4;
- <addr-length> = the length of the address, as specified in 7;
- <source> = the source address from the unsuccessfully delivered data packet.
- <destination> = the destination address from the unsuccessfully delivered data packet.

10.2.2 RERR Processing

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

- 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:

- Find the Routing Tuple (henceforth "matching Routing Tuple") in the Routing Set where:
  - R_dest_addr = <destination>
  - R_next_addr = previous-hop
- If no matching Routing Tuple is found, the RERR is not processed further, and is not considered for forwarding.
- Otherwise, if one matching Routing Tuple is found, this matching Routing Tuple is updated as follows:
  - R_time = expired

  The RERR message is, then, transmitted to the recorded R_next_addr towards <destination>.

10.2.3 RERR Transmission

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

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

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

11 Cost Types

This specification permits the use of different cost-types for when calculating route costs, and specifies one particularly simplified such cost-type in 11.3 purely intended as an example - while encouraging that more appropriate cost-types be developed for different deployment environments.

11.1 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 paths, the default <= comparison operator is defined so as to prefer paths with fewer weak links, even if such a path 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:

11.2 Specifying New Cost Types

When defining a cost-type, 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 3:

- 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 <= comparison operator. This MAY be by way of indicating that the definition in 11.1 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.

11.3 Example Cost Type: Hop Count

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

11.3.1 Simple Hop Count

- 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.
- When generating a RREQ or a RREP, the <route-cost> is initialized to one, the <route-cost> is incremented by one when forwarding.
- (WL,RC) <= (WL',RC') if and only if: RC < RC' OR if RC == RC'

12 Security Considerations

Currently, this memorandum 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.

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

Table 1: Packet Types

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

Table 2: TLV Types

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

Table 3: Cost Types

13 IANA Considerations

13.1 Multicast Addresses

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

13.2 Packet Types

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

13.3 TLV Types

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

13.4 Cost Types

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

When assigning a new Cost Type, 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

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

Table 4: Error Codes

RREQs (or RREPs) for the same destination represents the shortest path; 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.

13.5 Error Codes

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

14 Contributors

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

- Thomas Heide Clausen, LIX, France, <T.Clausen@computer.org>
- Axel Colin de Verdiere, LIX, France, <axel@axelcdv.com>
- Ulrich Herberg, Fujitsu Laboratories of America, USA <ulrich.herberg@us.fujitsu.com>

15 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.

References

[1] S. Bradner. Key words for use in rfcs to indicate requirement levels. Best Common Practice (BCP) 14, RFC 2119, March 1997.

[2] T. Clausen, C. Dearlove, and B. Adamson. Jitter Considerations in Mobile Ad Hoc Networks (MANETs). Standards Track RFC 5148, Feb. 2008.

[3] T. Clausen, C. Dearlove, J. Dean, and C. Adjih. Generalized MANET Packet/Message Format. Standards Track RFC 5444, Feb. 2009.

[4] P. Levis, T. Clausen, J. Hui, O. Gnawali, and J. Ko. The Trickle Algorithm. Standards Track RFC 6206, Mar. 2011.

[5] J. Macker. Simplified Multicast Forwarding. Internet Draft, work in progress, draft-ietf-manet-smf-10, Mar. 2010.

[6] G. Montenegro, N. Kushalnagar, J. Hui, and D. Culler. Transmission of IPv6 Packets over IEEE 802.15.4 Networks. Standards Track RFC 4944, Sep. 2007.

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

A LOADng Control Packet Illustrations

Centre de recherche INRIA Saclay – Île-de-France
Parc Orsay Université - ZAC des Vignes
4, rue Jacques Monod - 91893 Orsay Cedex (France)

Centre de recherche INRIA Bordeaux – Sud Ouest : Domaine Universitaire - 351, cours de la Libération - 33405 Talence Cedex
Centre de recherche INRIA Grenoble – Rhône-Alpes : 655, avenue de l'Europe - 38334 Montbonnot Saint-Ismier
Centre de recherche INRIA Lille – Nord Europe : Parc Scientifique de la Haute Borne - 40, avenue Halley - 59650 Villeneuve d'Ascq
Centre de recherche INRIA Nancy – Grand Est : LORIA, Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex
Centre de recherche INRIA Paris – Rocquencourt : Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex
Centre de recherche INRIA Rennes – Bretagne Atlantique : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex
Centre de recherche INRIA Sophia Antipolis – Méditerranée : 2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)
http://www.inria.fr
ISSN 0249-6399