

Network Working Group
Internet-Draft
Intended status: Experimental
Expires: January 5, 2015

J. Yi
T. Clausen
LIX, Ecole Polytechnique
July 4, 2014

**Smart Route Request for Lightweight On-demand Ad hoc Distance-vector
Routing - Next Generation
draft-yi-loadngsmartrreq-02**

Abstract

This document describes the Smart Route Request extension for Lightweight Ad hoc On-Demand - Next Generation (LOADng) distance vector routing protocol. It allows making use of existed routing information to forward Route Request message, and helps reducing routing overhead in LOADng.

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 January 5, 2015.

Copyright Notice

Copyright (c) 2014 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](#) and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as

described in the Simplified BSD License.

Table of Contents

1.	Introduction	3
2.	Terminology	3
3.	Applicability Statement	3
4.	Problem Statement	4
5.	Protocol Signaling and Information Bases	5
6.	Protocol Functioning	5
7.	Smart Route Request Message	6
7.1.	RREQ_SMART Generation	6
7.2.	RREQ_SMART Processing	6
7.3.	RREQ_SMART Forwarding	6
7.4.	RREQ_SMART Transmission	7
8.	Implementation Status	7
8.1.	Implementation of Ecole Polytechnique	8
9.	Security Considerations	8
10.	IANA Considerations	8
11.	References	8
11.1.	Normative References	8
11.2.	Informative References	9
Appendix A.	LOADng Smart Route Request Control Messages using RFC5444	9
A.1.	RREQ_SMART Messages Encoding Considerations	9
Appendix B.	RFC5444 -Specific IANA Considerations	10
Authors' Addresses	11

1. Introduction

Smart Route Request is an extension of LOADng protocol [[I-D.clausen-lln-loadng](#)] for use in forwarding Route Request message (RREQ), based on the routing information already known by the LOADng router.

In LOADng [[I-D.clausen-lln-loadng](#)], on receiving an RREQ message destined to other routers, an intermediate router has to multicast the RREQ message to all its neighbor routers. The Smart RREQ specified in this document makes use of available routing information in the local router, if possible, to reduce message multicasting. It does not require extract message exchange, and does not introduce computation overload.

Compared to RREQ dissemination by classical flooding, Smart RREQ can reduce up to 90% of route discovery overhead, depending on the scenarios applied.

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](#)].

This document uses the terminology and notation defined in [[I-D.clausen-lln-loadng](#)].

3. Applicability Statement

This protocol:

- o Is an extension of LOADng for RREQ message forwarding.
- o Makes use of routes available in the local router, and forward the RREQ message in unicast to the desired destination, if possible.
- o Can reduce the overhead used for route discovery in LOADng, especially in the scenarios where the data packets are sent to a few concentrators in the network.
- o Can work seamlessly with LOADng protocol, even with the LOADng Routers without Smart RREQ extension.

4. Problem Statement

In route discovery of LOADng [[I-D.clausen-11n-loadng](#)], the protocol explicitly prohibits intermediate routers from replying RREQ messages. Only the destination is permitted to respond to an RREQ by unicasting a Route Reply (RREP) message. For example, as shown in Figure 1, in a LOADng network, Router A initiates a route discovery to Router D. Even the intermediate Routers (Router B and Router C in this case) have already available routes to Router D, they have to broadcast the RREQ message until the message reaches the final destination Router D. The Router D can then send an RREP to build the router from Router A to Router D.

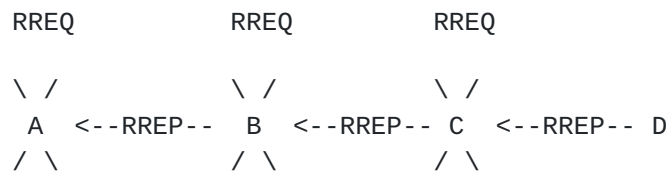


Figure 1: LOADng route discovery from Router A to Router D

Eliminating intermediate RREP can reduce the control message size and protocol complexity, but can also cause unnecessary multicast in the network. For example, in Figure 2, Router A initiates a route discovery to Router D. Router B, C and E have already routes to D. On receiving the RREQ, Router B, C and E still need to multicast the RREQ message to the whole network. The retransmission of the RREQ would be N (N is the number of routers in the network).

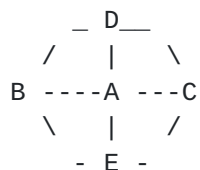


Figure 2: LOADng route discovery from Router A to Router D. Router B, C and E have already routes to Router D

In AODV [[RFC3561](#)], the intermediate routers can send reply to the originator of the router discovery. In the example of Figure 2, Router B, C and E will send an RREP to A directly, instead of flooding the RREQ message to the whole network. In this way, the RRE message can be kept "local" - but with the cost of complex sequence number check to avoid loops, and extra Gratuitous RREP message exchange.

The Smart Route Request described in this document reduces redundant multicast RREQ in LOADng if possible, while keeps the lightweight nature of LOADng.

5. Protocol Signaling and Information Bases

LOADng Smart Route Request is an extension of LOADng, and thus inherits all the information bases and signaling defined in [\[I-D.clausen-lln-loadng\]](#). Only one additional flag for RREQ message is introduced:

RREQ.smart-rreq is a boolean flag which, when set ('1'), indicates the message is an RREQ_SMART message, and MUST be processed according to this specification.

6. Protocol Functioning

This document concerns only RREQ dissemination of LOADng. When a LOADng Router initiates a route discovery, it floods an RREQ message with RREQ.smart-rreq flag set 1 (denoted RREQ_SMART message).

On receiving the RREQ_SMART message, the intermediate LOADng Router MUST process the message according to Section 12.2 of [\[I-D.clausen-lln-loadng\]](#). Prior to the message being transmitted, the LOADng Router will check if there is an available Routing Tuple to the destination of RREQ_SMART. If such Routing Tuple is found, the LOADng Router will unicast the RREQ_SMART message to the R_next_addr of the Routing Tuple. Or else, the RREQ_SMART message is transmitted according to the flooding operation specified for the network.

Figure 3 illustrates an example of operation of Smart Route Request. Router A requires a route discovery to Router D, and thus floods a RREQ_SMART message. Router B and C has already available routes to Router D. They will unicast the RREQ_SMART message to Router D.

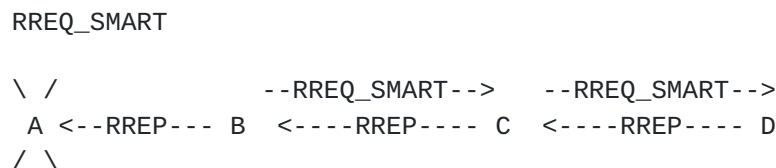


Figure 3: LOADng route discovery from Router A to Router D with Smart Route Request. Router B and Router C have already available routes to Router D

In the example in Figure 2, Router B, C and E will unicast the RREQ_SMART to Router D. Although those RREQ_SMART messages will not be processed by Router D because the messages carries longer routes, the RREQ_SMART message is kept local, instead of being flooded to the whole network. This is not a rare case in real application, like all the Routers send data to a concentrator in the network.

If the Smart RREQ defined in this specification is used, the RREQ_RETRIES parameter (defined in [[I-D.clausen-lln-loadng](#)]) MUST be great than 1. For a route discovery process, on the first RREQ message can be a RREQ_SMART message. For the following retries, the RREQ.smart-rreq flag MUST be cleared ('0').

7. Smart Route Request Message

The Smart Route Request Message (RREQ_SMART) are generated by a LOADng Router for the first try, when it has a data packet to deliver to a destination, but the LOADng Router has no matching tuple in the Routing Set. The basic operation follows Section 12 of [[I-D.clausen-lln-loadng](#)], except the RREQ.smart-rreq flag is treated differently, as specified in this section.

7.1. RREQ_SMART Generation

A LOADng Router with Smart Route Request extension generate an RREQ_SMART message for the first try of a route discovery process, with the following content:

- o RREQ.smart-rreq := TRUE;
- o All the other fields are set according to Section 12.1 of [[I-D.clausen-lln-loadng](#)].

If there was no RREP message received after $2 \times \text{NET_TRAVERSAL_TIME}$, another RREQ message MUST be initiated. The following RREQ messages MUST set RREQ.smart-rreq to FALSE, and be treated as normal RREQ message as specified in Section 12 of [[I-D.clausen-lln-loadng](#)].

7.2. RREQ_SMART Processing

RREQ_SMART messages are processed according to Section 12.2 of [[I-D.clausen-lln-loadng](#)].

7.3. RREQ_SMART Forwarding

The fields of an RREQ_SMART message considered for forwarding MUST be updated following Section 12.3 of [[I-D.clausen-lln-loadng](#)], prior to

it being transmitted.

The RREQ_SMART message is then transmitted, according to [Section 7.4](#).

7.4. RREQ_SMART Transmission

RREQ_SMART transmission is accomplished by the following procedure:

1. Find the Routing Tuple (henceforth, the "Matching Routing Tuple") in the Route Set, where:

- * R_dest_addr = RREQ.destination; and
- * R_next_addr does not equal to the interface address from which RREQ_SMART was received.

2. If a Matching Routing Tuple is found, find the Local Interface Tuple (henceforth, matching Local Interface Tuple) in the Local Interface Set where:

- * I_local_iface_addr_list contains R_local_iface_addr from the Matching Routing Tuple

The RREQ_SMART is transmitted over the LOADng Interface, identified by the Matching Interface Tuple to the neighbor LOADng Router, identified by R_next_addr from the Matching Routing Tuple.

3. Otherwise, the RREQ_SMART is transmitted according to [Section 12.4](#) of [[I-D.clausen-lln-loadng](#)].

8. Implementation Status

This section records the status of known implementations of the protocol defined by this specification at the time of posting of this Internet-Draft, and based on a proposal described in [[RFC6982](#)]. The description of implementations in this section is intended to assist the IETF in its decision processes in progressing drafts to RFCs. Please note that the listing of any individual implementation here does not imply endorsement by the IETF. Furthermore, no effort has been spent to verify the information presented here that was supplied by IETF contributors. This is not intended as, and must not be construed to be, a catalog of available implementations or their features. Readers are advised to note that other implementations may exist.

According to [[RFC6982](#)], "this will allow reviewers and working groups

to assign due consideration to documents that have the benefit of running code, which may serve as evidence of valuable experimentation and feedback that have made the implemented protocols more mature. It is up to the individual working groups to use this information as they see fit".

There are currently one publicly-known implementation of smart route request extension of LOADng specified in this document.

8.1. Implementation of Ecole Polytechnique

This implementation is developed by the Networking Group at Ecole Polytechnique and applied to LOADng [[I-D.clausen-lln-loadng](#)] for RREQ message forwarding. It can run over real network interfaces, and can also be integrated with the network simulator NS2. It is a Java implementation, and can be used on any platform that includes a Java virtual machine.

The implementation is based on -00 revision of this document, and includes about 20 line of additional code to the LOADng implementation. Simulation results based on NS2 have been published in [[IEEE ICWITS2012](#)]. The results show that in point-to-point scenarios, Smart RREQ extension can save 30% RREQ flooding overhead compared to LOADng; in multipoint-to-point scenarios, Smart RREQ extension can save up to 90% RREQ flooding overhead compared to LOADng.

9. Security Considerations

This Smart RREQ extension inherits the vulnerabilities of LOADng, as discussed in section 18 of [[I-D.clausen-lln-loadng](#)]. No additional vulnerability is introduced in this extension.

10. IANA Considerations

IANA is requested to

11. References

11.1. Normative References

[I-D.clausen-lln-loadng]
Clausen, T., Verdiere, A., Yi, J., Niktash, A., Igarashi, Y., Satoh, H., Herberg, U., Lavenue, C., Lys, T., and J. Dean, "The Lightweight On-demand Ad hoc Distance-vector

Routing Protocol - Next Generation (LOADng)",
[draft-clausen-lln-loadng-10](#) (work in progress),
October 2013.

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), March 1997.
- [RFC5444] Clausen, T., Dearlove, C., Dean, J., and C. Adjih, "Generalized Mobile Ad Hoc Network (MANET) Packet/Message Format", [RFC 5444](#), February 2009.

11.2. Informative References

- [IEEE_ICWITS2012] Yi, J., Clausen, T., and A. Bas, "Smart Route Request for on-demand route discovery in constrained environments", Proceedings of IEEE ICWITS2012, IEEE International Conference on Wireless Information Technology and Systems, 2012.
- [RFC3561] Perkins, C., Belding-Royer, E., and S. Das, "Ad hoc On-Demand Distance Vector (AODV) Routing", [RFC 3561](#), July 2003.
- [RFC6982] Sheffer, Y. and A. Farrel, "Improving Awareness of Running Code: The Implementation Status Section", [RFC 6982](#), July 2013.

Appendix A. LOADng Smart Route Request Control Messages using [RFC5444](#)

This section presents how the abstract LOADng Smart Route Request messages, used throughout this specification, are mapped into [\[RFC5444\]](#) messages. It only concerns the flag RREQ.smart-rreq.

A.1. RREQ_SMART Messages Encoding Considerations

This protocol makes use of RREQ message defined in [\[I-D.clausen-lln-loadng\]](#). Therefore, it reuses the RREQ Message Type defined in [\[I-D.clausen-lln-loadng\]](#), and defines one additional flags: RREQ.smart-rreq. Table 1 describes how the flag is mapped into [\[RFC5444\]](#).

RREQ Element	RFC5444 -Element	Considerations
RREQ.smart-rreq	FLAGS Message TLV	Encoded by way of a Message-Type-specific Message TLV of type FLAGS, defined in Table 3

Table 1: RREQ Message Elements for Smart Route Request

Appendix B. [RFC5444](#)-Specific IANA Considerations

This document only specifies one addition flag of RREQ, which has been allocated in "Message Types" namespace of [[RFC5444](#)], in [[I-D.clausen-lln-loadng](#)].

IANA is requested to add a RREQ Message-Type-specific Message TLV Type, in accordance with [Section 6.2.1 of \[RFC5444\]](#), with allocation policies as specified in Table 2.

Type	Description	Allocation Policy
129	FLAGS	
130-223	Unassigned	Expert Review

Table 2: RREQ Message-Type-specific TLV Type for LOADng-CT

Allocation of the FLAGS TLV from the RREQ Message-Type-specific Message TLV Types in Table 2 will create a new Type Extension registry, with type extension 0, as illustrated in Table 3.

Name	Type	Type Extension	Bit	Description
FLAGS	129	0	0	RREQ.smart-rreq flag (i.e., the RREQ message is a RREQ_SMART when it is set to 1)
FLAGS	129	1-255		Unassigned

Table 3: Message TLV Type assignment: FLAGS

Authors' Addresses

Jiazi Yi
LIX, Ecole Polytechnique

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

Thomas Clausen
LIX, Ecole Polytechnique
91128 Palaiseau Cedex,
France

Phone: +33-6-6058-9349
Email: T.Clausen@computer.org
URI: <http://www.thomasclausen.org>

