

Network Working Group  
Internet-Draft  
Intended status: Informational  
Expires: April 25, 2013

T. Clausen  
A. Camacho  
J. Yi  
A. Colin de Verdiere  
LIX, Ecole Polytechnique  
Y. Igarashi  
H. Satoh  
Y. Morii  
Hitachi, Ltd., Yokohama Research  
Laboratory  
U. Herberg  
Fujitsu Laboratories of America  
C. Lavenu  
EDF R&D  
October 22, 2012

Interoperability Report of the Lightweight On-demand Ad hoc Distance-  
vector Routing Protocol - Next Generation (LOADng)  
draft-lavenu-lln-loadng-interoperability-report-03

## Abstract

This document reports experience with the LOADng routing protocol, as obtained by way of a number of interoperability tests during the protocol development.

## 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 April 25, 2013.

## Copyright Notice

Copyright (c) 2012 IETF Trust and the persons identified as the document authors. All rights reserved.

Internet-Draft

LOADng Interop Report

October 2012

This document is subject to [BCP 78](http://trustee.ietf.org/license-info) 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.

This document may contain material from IETF Documents or IETF Contributions published or made publicly available before November 10, 2008. The person(s) controlling the copyright in some of this material may not have granted the IETF Trust the right to allow modifications of such material outside the IETF Standards Process. Without obtaining an adequate license from the person(s) controlling the copyright in such materials, this document may not be modified outside the IETF Standards Process, and derivative works of it may not be created outside the IETF Standards Process, except to format it for publication as an RFC or to translate it into languages other than English.

## Table of Contents

<a href="#">1.</a>	Introduction . . . . .	<a href="#">5</a>
<a href="#">2.</a>	Terminology . . . . .	<a href="#">5</a>
<a href="#">3.</a>	Interoperability Scenarios . . . . .	<a href="#">6</a>
3.1.	Scenario 01: 1-hop Bidirectional Route Establishment - Forward Route and Reverse Route initial installation . . .	<a href="#">6</a>
<a href="#">3.1.1.</a>	Scenario Topology . . . . .	<a href="#">6</a>
<a href="#">3.1.2.</a>	Expected Message Sequencing . . . . .	<a href="#">6</a>
3.2.	Scenario 02: 1-hop Bidirectional Route Establishment -Forward Route and Reverse Route updating . . . . .	<a href="#">7</a>
<a href="#">3.2.1.</a>	Scenario Topology . . . . .	<a href="#">7</a>
<a href="#">3.2.2.</a>	Expected Message Sequencing . . . . .	<a href="#">7</a>
3.3.	Scenario 03: 2-hop bidirectional route establishment - Forward Route and Reverse Route initial installation . . .	<a href="#">8</a>
<a href="#">3.3.1.</a>	Scenario Topology . . . . .	<a href="#">8</a>
<a href="#">3.3.2.</a>	Expected Message Sequencing . . . . .	<a href="#">9</a>
3.4.	Scenario 04: 2-hop bidirectional route establishment - Forward Route and Reverse Route updating . . . . .	<a href="#">10</a>
<a href="#">3.4.1.</a>	Scenario Topology . . . . .	<a href="#">10</a>
<a href="#">3.4.2.</a>	Expected Message Sequencing . . . . .	<a href="#">10</a>

3.5.	Scenario 05: 2-hop bidirectional route establishment -	
	Link breakage handling . . . . .	<a href="#">11</a>
<a href="#">3.5.1.</a>	Scenario Topology . . . . .	<a href="#">11</a>
<a href="#">3.5.2.</a>	Expected Message Sequencing . . . . .	<a href="#">11</a>
3.6.	Scenario 06: 3-hop bidirectional route establishment -	

	Forward Route and Reverse Route initial installation . . .	<a href="#">12</a>
<a href="#">3.6.1.</a>	Scenario Topology . . . . .	<a href="#">12</a>
<a href="#">3.6.2.</a>	Expected Message Sequencing . . . . .	<a href="#">13</a>
3.7.	Scenario 07: 3-hop bidirectional route establishment -	
	Forward Route and Reverse Route updating . . . . .	<a href="#">14</a>
<a href="#">3.7.1.</a>	Scenario Topology . . . . .	<a href="#">14</a>
<a href="#">3.7.2.</a>	Expected Message Sequencing . . . . .	<a href="#">15</a>
3.8.	Scenario 08: 3-hop bidirectional route establishment -	
	Link breakage handling . . . . .	<a href="#">16</a>
<a href="#">3.8.1.</a>	Scenario Topology . . . . .	<a href="#">16</a>
<a href="#">3.8.2.</a>	Expected Message Sequencing . . . . .	<a href="#">16</a>
3.9.	Scenario 09: 4-hop bidirectional route establishment -	
	Forward Route and Reverse Route initial installation . . .	<a href="#">17</a>
<a href="#">3.9.1.</a>	Scenario Topology . . . . .	<a href="#">18</a>
<a href="#">3.9.2.</a>	Expected Message Sequencing . . . . .	<a href="#">18</a>
3.10.	Scenario 10: 4-hop bidirectional route establishment -	
	Link breakage handling . . . . .	<a href="#">20</a>
<a href="#">3.10.1.</a>	Scenario Topology . . . . .	<a href="#">20</a>
<a href="#">3.10.2.</a>	Expected Message Sequencing . . . . .	<a href="#">21</a>
3.11.	Scenario 11: Establishment of the best bidirectional	
	route . . . . .	<a href="#">22</a>
<a href="#">3.11.1.</a>	Scenario Topology . . . . .	<a href="#">22</a>
<a href="#">3.11.2.</a>	Expected Message Sequencing . . . . .	<a href="#">22</a>
<a href="#">3.12.</a>	Scenario 12: Blacklisting . . . . .	<a href="#">23</a>
<a href="#">3.12.1.</a>	Scenario Topology . . . . .	<a href="#">24</a>
<a href="#">3.12.2.</a>	Expected Message Sequencing . . . . .	<a href="#">24</a>
4.	Interop 01: Yokohama, Japan, October 2011 . . . . .	<a href="#">27</a>
<a href="#">4.1.</a>	Version of LOADng Specification Tested . . . . .	<a href="#">27</a>
<a href="#">4.2.</a>	Place and Date of Interoperability Test . . . . .	<a href="#">28</a>
<a href="#">4.3.</a>	Participating Implementations . . . . .	<a href="#">28</a>
<a href="#">4.4.</a>	Scenarios Tested . . . . .	<a href="#">28</a>
<a href="#">4.5.</a>	Additional Interoperability Test Considerations . . . . .	<a href="#">28</a>
<a href="#">4.6.</a>	Results For Scenario 01 . . . . .	<a href="#">29</a>
<a href="#">4.7.</a>	Results For Scenario 02 . . . . .	<a href="#">29</a>
<a href="#">4.8.</a>	Results For Scenario 03 . . . . .	<a href="#">29</a>
<a href="#">4.9.</a>	Results For Scenario 04 . . . . .	<a href="#">30</a>

<a href="#">4.10.</a>	Results For Scenario 05 . . . . .	<a href="#">30</a>
<a href="#">4.11.</a>	Results For Scenario 06 . . . . .	<a href="#">31</a>
<a href="#">4.12.</a>	Results For Scenario 07 . . . . .	<a href="#">31</a>
<a href="#">4.13.</a>	Results For Scenario 08 . . . . .	<a href="#">31</a>
<a href="#">4.14.</a>	Results For Scenario 09 . . . . .	<a href="#">32</a>
<a href="#">4.15.</a>	Results For Scenario 10 . . . . .	<a href="#">32</a>
<a href="#">4.16.</a>	Results For Scenario 11 . . . . .	<a href="#">32</a>
<a href="#">4.17.</a>	Results For Scenario 12 . . . . .	<a href="#">33</a>
<a href="#">4.18.</a>	Conclusions . . . . .	<a href="#">33</a>
<a href="#">5.</a>	Interop 02: San Jose, USA March 2012 . . . . .	<a href="#">35</a>
<a href="#">5.1.</a>	LOADng version tested . . . . .	<a href="#">35</a>
<a href="#">5.2.</a>	Place and Date of Interoperability Test . . . . .	<a href="#">35</a>

<a href="#">5.3.</a>	Participating Implementations . . . . .	<a href="#">35</a>
<a href="#">5.4.</a>	Interoperability Test Considerations . . . . .	<a href="#">36</a>
<a href="#">5.5.</a>	Results For Scenario 01 . . . . .	<a href="#">36</a>
<a href="#">5.6.</a>	Results For Scenario 03 . . . . .	<a href="#">36</a>
<a href="#">5.7.</a>	Results For Scenario 05 . . . . .	<a href="#">36</a>
<a href="#">6.</a>	Interop 03: Los Angeles, USA, June 2012 . . . . .	<a href="#">37</a>
<a href="#">6.1.</a>	Version of LOADng Specification Tested . . . . .	<a href="#">37</a>
<a href="#">6.2.</a>	Place and Date of Interoperability Test . . . . .	<a href="#">37</a>
<a href="#">6.3.</a>	Participating Implementations . . . . .	<a href="#">37</a>
<a href="#">6.4.</a>	Scenarios Tested . . . . .	<a href="#">37</a>
<a href="#">6.5.</a>	Additional Interoperability Test Considerations . . . . .	<a href="#">37</a>
<a href="#">6.6.</a>	Results For Scenario 01-02 . . . . .	<a href="#">38</a>
<a href="#">6.7.</a>	Results For Scenario 03-04-05 . . . . .	<a href="#">38</a>
<a href="#">6.8.</a>	Results For Scenario 06-07-08 . . . . .	<a href="#">39</a>
<a href="#">6.9.</a>	Results For Scenario 09-10 . . . . .	<a href="#">40</a>
<a href="#">6.10.</a>	Results For Scenario 11 . . . . .	<a href="#">40</a>
<a href="#">6.11.</a>	Conclusions . . . . .	<a href="#">40</a>
<a href="#">7.</a>	Security Considerations . . . . .	<a href="#">41</a>
<a href="#">8.</a>	IANA Considerations . . . . .	<a href="#">41</a>
<a href="#">9.</a>	Contributors . . . . .	<a href="#">41</a>
<a href="#">10.</a>	Acknowledgments . . . . .	<a href="#">41</a>
<a href="#">11.</a>	References . . . . .	<a href="#">41</a>
<a href="#">11.1.</a>	Normative References . . . . .	<a href="#">41</a>
<a href="#">11.2.</a>	Informative References . . . . .	<a href="#">42</a>

## 1. Introduction

This document reports experience with the LOADng [[LOADng](#)] routing protocol, as obtained by way of a number of interoperability tests during the protocol development.

Interoperability tests between LOADng Routers implemented on the basis of the different versions of the protocol have been undertaken mainly to:

- o Show evidence that interoperable LOADng implementations do exist.
- o Clarify and improve the overall quality of the LOADng specification.
- o Demonstrate that the final LOADng internet draft can be considered as a standalone specification allowing the development of interoperable implementations of LOADng.

## 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.

Destination - The address of a router or host, to which a route is sought discovered and maintained.

Originator - The address of a router, which seeks to discover and maintain a route to a Destination.

Forward Route - 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.

Reverse Route - 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.

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., 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.

### [3.](#) Interoperability Scenarios

This section describes the various tests and scenarios carried out between the implementations involved in the various interoperability tests.

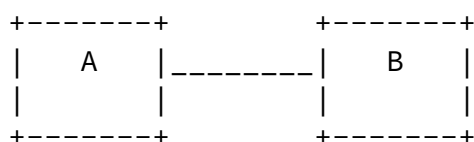
The testbed required is composed of up to five LOADng Routers, connected according to the specific topology described for each test scenario below. The LOADng routing protocol was run over UDP and IPv4. Either Ethernet or 802.11 wireless network was used in the test.

### [3.1.](#) Scenario 01: 1-hop Bidirectional Route Establishment - Forward Route and Reverse Route initial installation

For each implementation, this test aims to verify the initial installation of a bidirectional route (Forward Route and Reverse Route from A to B) within the LOADng Router routing tables (Routing Sets) through the effective generation and processing of LOADng control messages (RREQ, RREP, RREP-ACK).

#### [3.1.1.](#) Scenario Topology

The testbed is composed of two LOADng Routers:



This test suite consists in establishing a bidirectional route between LOADng Router A and LOADng Router B.

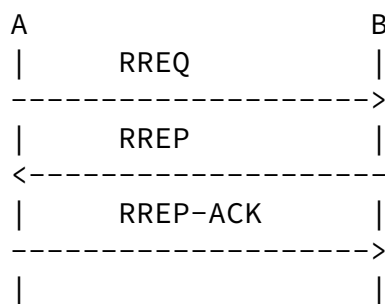
#### [3.1.2.](#) Expected Message Sequencing

The expected message sequencing is as follows:

- o LOADng Router A generates an RREQ message intended for LOADng Router B.

- o Upon receiving the RREQ, LOADng Router B installs a new tuple in its Routing Set towards LOADng Router A (Reverse Route from LOADng Router B to LOADng Router A) and sends an unicast RREP message intended for LOADng Router A, soliciting an RREP-ACK message.
- o Upon receiving the RREP, LOADng Router A installs a new tuple in its Routing Set towards LOADng Router B (Forward Route from LOADng Router A to LOADng Router B) and sends an unicast RREP-ACK message

to LOADng Router B.

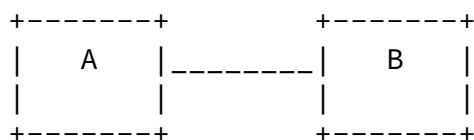


### [3.2.](#) Scenario 02: 1-hop Bidirectional Route Establishment -Forward Route and Reverse Route updating

For each implementation, this test aims to verify the refreshment of a bidirectional route (Forward Route and Reverse Route from A to B) already installed within the LOADng Router routing tables (Routing Sets) through the effective generation and processing of LOADng control messages (RREQ, RREP, RREP-ACK).

#### [3.2.1.](#) Scenario Topology

The testbed is composed of two LOADng Routers:



This test suite consists in updating a bidirectional route between LOADng Router A and LOADng Router B.

#### [3.2.2.](#) Expected Message Sequencing

The expected message sequencing is as follows:

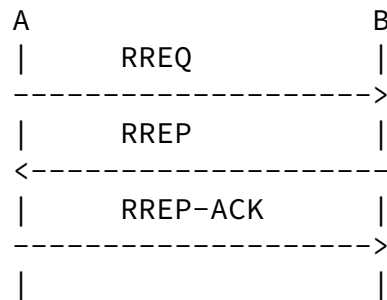
- o LOADng Router A generates an RREQ message intended for LOADng Router B.

- o Upon receiving the RREQ, LOADng Router B updates the corresponding



route (Reverse Route from LOADng Router B to LOADng Router A) already installed within its Routing Set and sends an unicast RREP message intended for LOADng Router A, soliciting an RREP-ACK message.

- o Upon receiving the RREP, LOADng Router A updates the corresponding route (Forward Route from LOADng Router A to LOADng Router B) already installed within its Routing Set and sends an unicast RREP-ACK message to LOADng Router B.

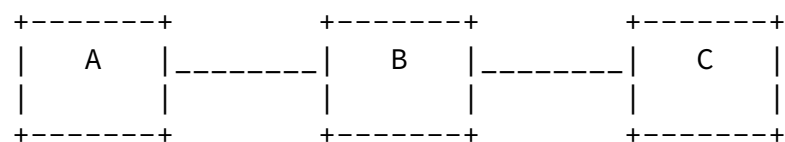


### [3.3.](#) Scenario 03: 2-hop bidirectional route establishment - Forward Route and Reverse Route initial installation

This test aims to verify the initial installation of a bidirectional route (Forward Route and Reverse Route from A to C) within the LOADng Router routing tables (Routing Sets) through the effective forwarding of LOADng control traffic by LOADng Router B which is located between LOADng Router A and LOADng Router C. It is also verified that RREP-ACK messages are not forwarded by the LOADng Routers these messages are intended for.

#### [3.3.1.](#) Scenario Topology

The testbed is composed of three LOADng Routers. Control traffic generated by either LOADng Router A towards LOADng Router C or LOADng Router C towards LOADng Router A has to be forwarded by LOADng Router B:

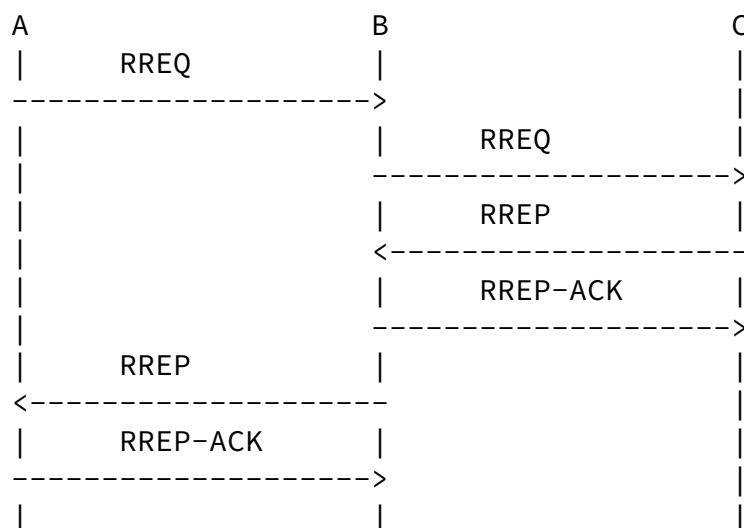


This test suite consists in establishing a bidirectional route between LOADng Router A and LOADng Router C.

### 3.3.2. Expected Message Sequencing

The expected message sequencing is as follows:

- o LOADng Router A generates an RREQ message intended for LOADng Router C.
- o Upon receiving the RREQ, LOADng Router B installs a new tuple in its Routing Set towards LOADng Router A (Reverse Route from LOADng Router B to LOADng Router A) and forwards the received RREQ.
- o Upon receiving the RREQ, LOADng Router C installs a new tuple in its Routing Set towards LOADng Router A (Reverse Route from LOADng Router C to LOADng Router A) and a new tuple towards LOADng Router B (Reverse route from LOADng Router C to LOADng Router B). The reception of the RREQ also triggers the generation of an unicast RREP message intended for LOADng Router A, soliciting an RREP-ACK message.
- o Upon receiving the RREP, LOADng Router B installs a new tuple in its Routing Set towards LOADng Router C (Forward Route from LOADng Router B to LOADng Router C), sends an unicast RREP-ACK message to LOADng Router C and forwards the RREP received previously.
- o Upon receiving the RREP, LOADng Router A installs a new tuple in its Routing Set towards LOADng Router B (Forward Route from LOADng Router A to LOADng Router B) and a new tuple towards LOADng Router C (Forward Route from LOADng Router A to LOADng Router C). The reception of the RREP also triggers an unicast RREP-ACK message intended for LOADng Router B.

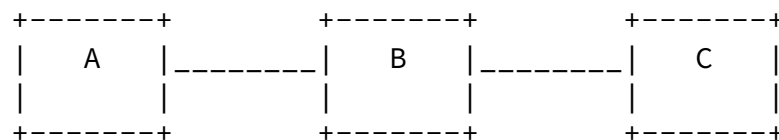


### [3.4.](#) Scenario 04: 2-hop bidirectional route establishment - Forward Route and Reverse Route updating

This test aims to verify the refreshment of a bidirectional route (Forward Route and Reverse Route from A to C) already installed within the LOADng Router routing tables (Routing Sets) through the effective forwarding of LOADng control traffic by LOADng Router B which is located between LOADng Router A and LOADng Router C.

#### [3.4.1.](#) Scenario Topology

The testbed is composed of three LOADng Routers. Control traffic generated by either LOADng Router A towards LOADng Router C or LOADng Router C towards LOADng Router A has to be forwarded by LOADng Router B:



This test suite consists in updating a bidirectional route between LOADng Router A and LOADng Router C.

#### [3.4.2.](#) Expected Message Sequencing

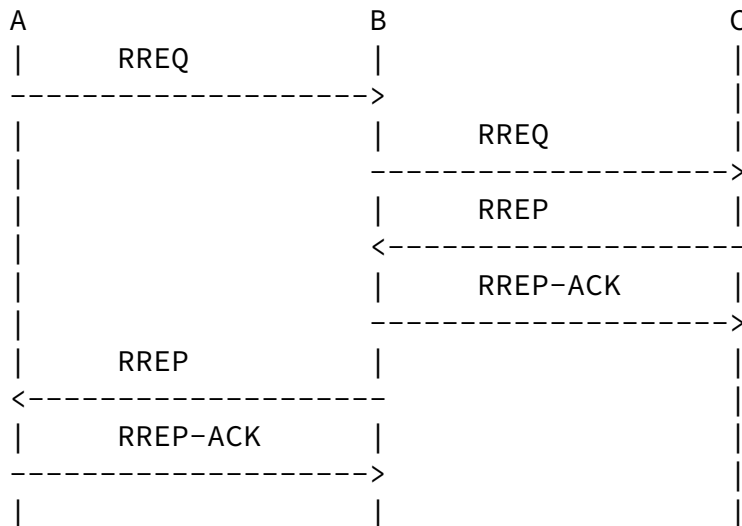
The expected message sequencing is as follows:

- o LOADng Router A generates an RREQ message intended for LOADng Router C.
- o Upon receiving the RREQ, LOADng Router B updates the corresponding route (Reverse Route from LOADng Router B to LOADng Router A) already installed within its Routing Set and forwards the received RREQ.
- o Upon receiving the RREQ, LOADng Router C updates the corresponding routes (Reverse Routes from LOADng Router C to LOADng Router A and from LOADng Router C to LOADng Router B). The reception of the

RREQ also triggers the generation of an unicast RREP message intended for LOADng Router A, soliciting an RREP-ACK message.

- o Upon receiving the RREP, LOADng Router B updates the corresponding route (Forward route from LOADng Router B to LOADng Router C), sends an unicast RREP-ACK message to LOADng Router C and forwards the RREP received previously.

- o Upon receiving the RREP, LOADng Router A updates the corresponding routes (Forward routes from LOADng Router A to LOADng Router B and from LOADng Router A to LOADng Router C). The reception of the RREP also triggers an unicast RREP-ACK message intended for LOADng Router B.

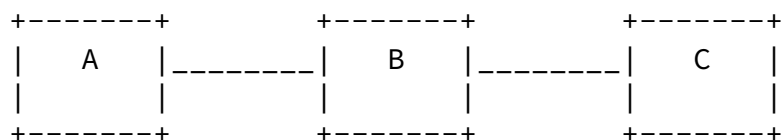


### [3.5.](#) Scenario 05: 2-hop bidirectional route establishment - Link breakage handling

This test aims to verify the proper generation and processing of an RERR message after an artificially created link breakage on an previously established bidirectional route.

#### [3.5.1.](#) Scenario Topology

The testbed is composed of three LOADng Routers. Control traffic generated by either LOADng Router A towards LOADng Router C or LOADng Router C towards LOADng Router A has to be forwarded by LOADng Router B:



This test suite consists in handling link breakages between routers.

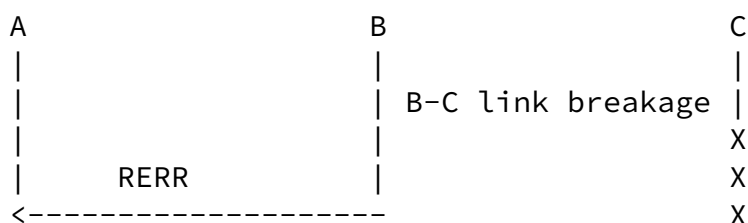
### 3.5.2. Expected Message Sequencing

The expected message sequencing is as follows:

- o A bidirectional route is already established between LOADng Routers A and C.
- o At some time, link breakage is detected by LOADng Router B. Consequently, an unicast RERR message intended for LOADng Router A (here the assumption is made that the unsuccessful delivered data traffic would have been generated by LOADng Router A) is transmitted.

Note: link breakage is provoked artificially and its detection by LOADng Router B is triggered manually (normally, this would be triggered by failure in sending data traffic intended for LOADng Router C).

- o Upon receiving the RERR, LOADng Router A updates its Routing Set by invalidating the existing Forward Route from LOADng Router A to LOADng Router C.

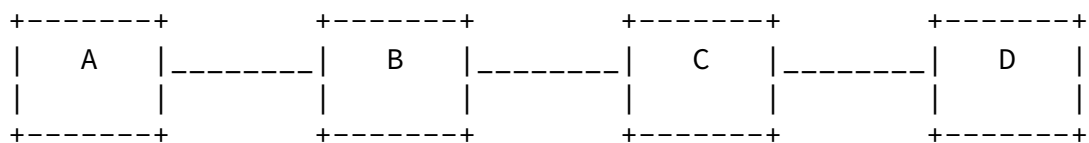


### [3.6.](#) Scenario 06: 3-hop bidirectional route establishment - Forward Route and Reverse Route initial installation

This test aims to verify the initial installation of a bidirectional route (Forward Route and Reverse Route from A to D) within the LOADng Router routing tables (Routing Sets) through the effective forwarding of LOADng control traffic by LOADng Routers B and C, which are located between LOADng Router A and LOADng Router D. It is also verified that RREP-ACK messages are not forwarded by the LOADng Routers these messages are intended for.

#### [3.6.1.](#) Scenario Topology

The testbed is composed of four LOADng Routers. Control traffic generated by either LOADng Router A towards LOADng Router D or LOADng Router D towards LOADng Router A has to be forwarded by LOADng Routers B and C:



This test suite consists in establishing a bidirectional route between LOADng Router A and LOADng Router D.

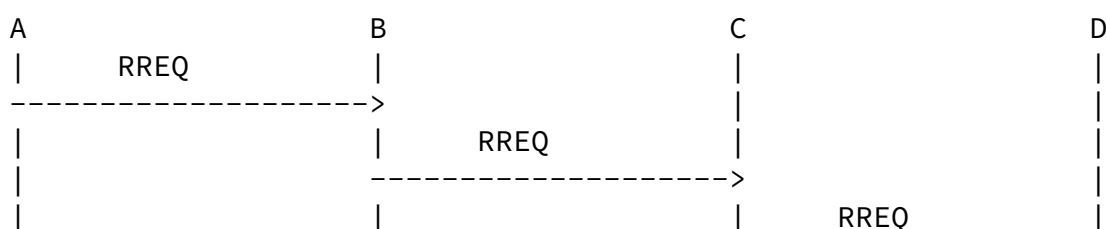
#### [3.6.2.](#) Expected Message Sequencing

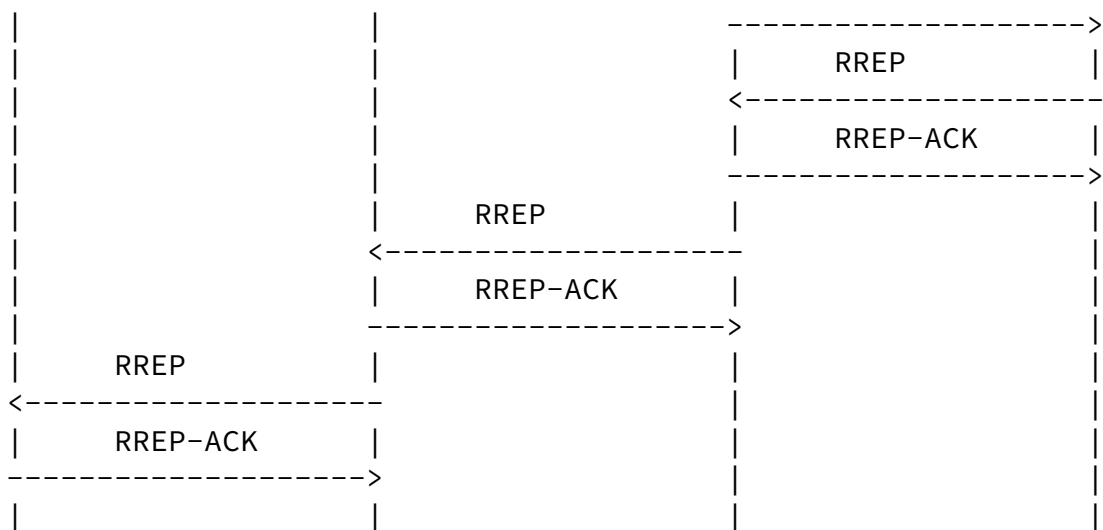
The expected message sequencing is as follows:

- o LOADng Router A generates an RREQ message intended for LOADng Router D.
- o Upon receiving the RREQ, LOADng Router B installs a new tuple in its Routing Set towards LOADng Router A (Reverse Route from LOADng Router B to LOADng Router A) and forwards the received RREQ.

- o Upon receiving the RREQ, LOADng Router C installs a new tuple in its Routing Set towards LOADng Router A (Reverse Route from LOADng Router C to LOADng Router A) and a new tuple towards LOADng Router B (Reverse route from LOADng Router C to LOADng Router B) and forwards the received RREQ.
- o Upon receiving the RREQ, LOADng Router D installs a new tuple in its Routing Set towards LOADng Router A (Reverse Route from LOADng Router D to LOADng Router A) and a new tuple towards LOADng Router C (Reverse route from LOADng Router D to LOADng Router C). The reception of the RREQ also triggers the generation of an unicast RREP message intended for LOADng Router A, soliciting an RREP-ACK message.
- o Upon receiving the RREP, LOADng Router C installs a new tuple in its Routing Set towards LOADng Router D (Forward Route from LOADng Router C to LOADng Router D), sends an unicast RREP-ACK message to LOADng Router D and forwards the RREP received previously.
- o Upon receiving the RREP, LOADng Router B installs a new tuple in its Routing Set towards LOADng Router D (Forward Route from LOADng Router B to LOADng Router D) and a new tuple towards LOADng Router C (Forward Route from LOADng Router B to LOADng Router C). An unicast RREP-ACK message is also sent to LOADng Router C and the RREP received previously is forwarded.
- o Upon receiving the RREP, LOADng Router A installs a new tuple in its Routing Set towards LOADng Router B (Forward Route from LOADng Router A to LOADng Router B) and a new tuple towards LOADng Router

D (Forward Route from LOADng Router A to LOADng Router D). The reception of the RREP also triggers an unicast RREP-ACK message intended for LOADng Router B.



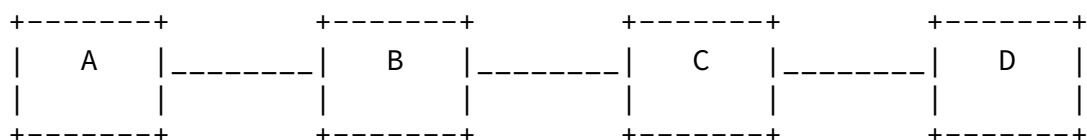


### 3.7. Scenario 07: 3-hop bidirectional route establishment - Forward Route and Reverse Route updating

This test aims to verify the refreshment of a bidirectional route (Forward Route and Reverse Route from A to D) already installed within the LOADng Router routing tables (Routing Sets) through the effective forwarding of LOADng control traffic by LOADng Routers B and C which are located between LOADng Router A and LOADng Router D.

#### 3.7.1. Scenario Topology

The testbed is composed of four LOADng Routers. Control traffic generated by either LOADng Router A towards LOADng Router D or LOADng Router D towards LOADng Router A has to be forwarded by LOADng Routers B and C:



This test suite consists in updating a bidirectional route between

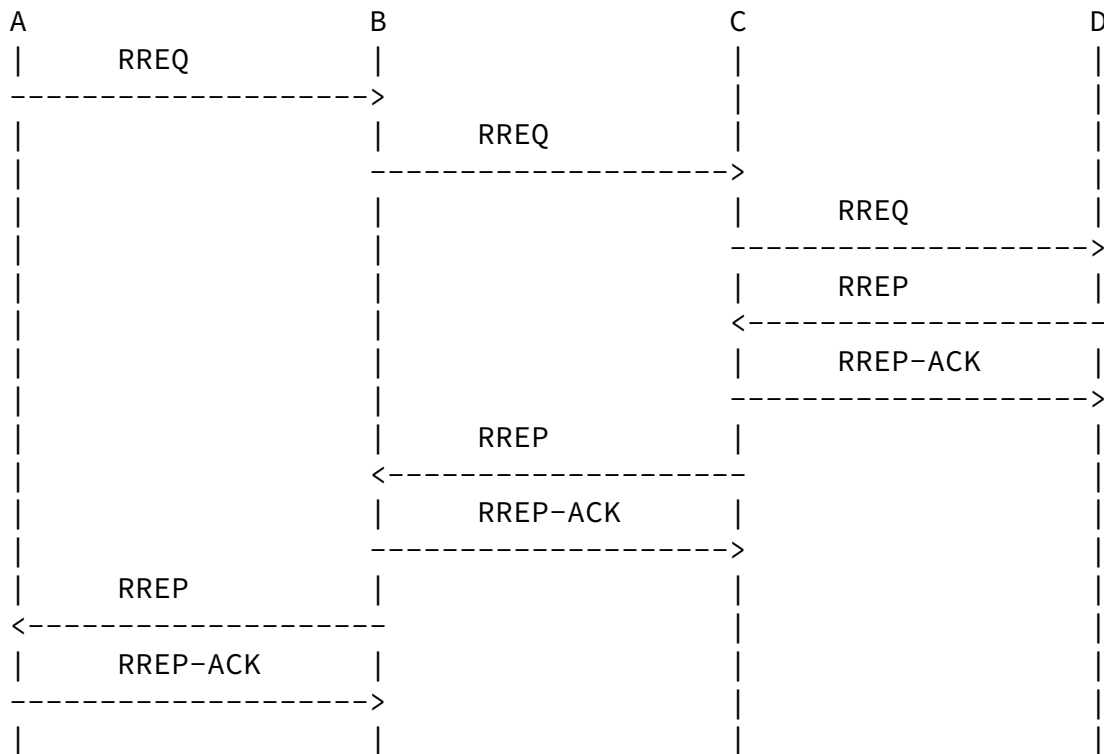
LOADng Router A and LOADng Router D.

#### 3.7.2. Expected Message Sequencing



The expected message sequencing is as follows:

- o LOADng Router A generates an RREQ message intended for LOADng Router D.
- o Upon receiving the RREQ, LOADng Router B updates the corresponding route (Reverse Route from LOADng Router B to LOADng Router A) already installed within its Routing Set and forwards the received RREQ.
- o Upon receiving the RREQ, LOADng Router C updates the corresponding routes (Reverse Routes from LOADng Router C to LOADng Router A and from LOADng Router C to LOADng Router B) already installed within its Routing Set and forwards the received RREQ.
- o Upon receiving the RREQ, LOADng Router D updates the corresponding routes (Reverse Routes from LOADng Router D to LOADng Router A and from LOADng Router D to LOADng Router C) already installed within its Routing Set. The reception of the RREQ also triggers the generation of an unicast RREP message intended for LOADng Router A, soliciting an RREP-ACK message.
- o Upon receiving the RREP, LOADng Router C updates the corresponding route (Forward Route from LOADng Router C to LOADng Router D), sends an unicast RREP-ACK message to LOADng Router D and forwards the RREP received previously.
- o Upon receiving the RREP, LOADng Router B updates the corresponding routes (Forward Route from LOADng Router B to LOADng Router D and from LOADng Router B to LOADng Router C). An unicast RREP-ACK message is also sent to LOADng Router C and the RREP received previously is forwarded.
- o Upon receiving the RREP, LOADng Router A updates the corresponding routes (Forward Route from LOADng Router A to LOADng Router B and from LOADng Router A to LOADng Router D). The reception of the RREP also triggers an unicast RREP-ACK message intended for LOADng Router B.

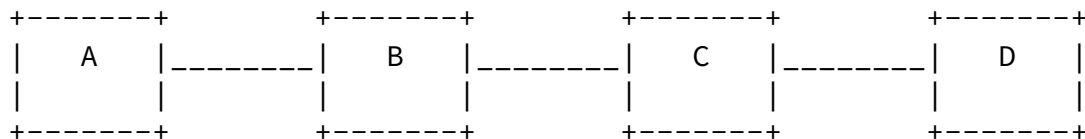


### [3.8.](#) Scenario 08: 3-hop bidirectional route establishment - Link breakage handling

This test aims to verify the proper generation, processing and forwarding of a RERR message after an artificially created link breakage on an previously established bidirectional route.

#### [3.8.1.](#) Scenario Topology

The testbed is composed of four LOADng Routers. Control traffic generated by either LOADng Router A towards LOADng Router D or LOADng Router D towards LOADng Router A has to be forwarded by LOADng Routers B and C:



This test suite consists in handling link breakages between LOADng Routers.

#### [3.8.2.](#) Expected Message Sequencing

The expected message sequencing is as follows:

Internet-Draft

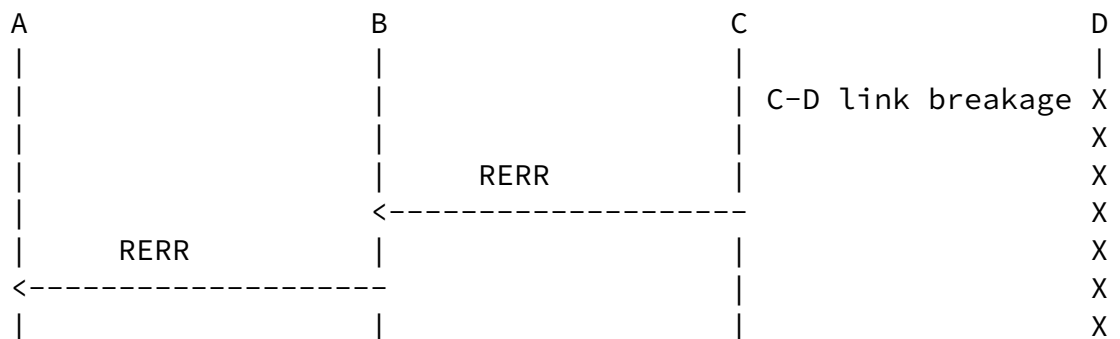
LOADng Interop Report

October 2012

- o A bidirectional route is already established between LOADng Routers A and D.
- o At some time, link breakage is detected by LOADng Router C. Consequently, an unicast RERR message intended for LOADng Router A (here the assumption is made that the unsuccessful delivered data traffic would have been generated by LOADng Router A) is transmitted to LOADng Router B according to the Reverse Route from LOADng Router C to LOADng Router A computed previously.

Note: link breakage is provoked artificially and its detection by LOADng Router C is triggered manually (normally, this would be triggered by failure in sending data traffic intended for LOADng Router D).

- o Upon receiving the RERR, LOADng Router B updates its Routing Set by invalidating the existing Forward Route from LOADng Router B to LOADng Router D. Afterwards, the RERR message is forwarded according to the existing Reverse Route from LOADng Router B to LOADng Router A.
- o Upon receiving the RERR, LOADng Router A updates its Routing Set by invalidating the existing Forward Route from LOADng Router A to LOADng Router D.



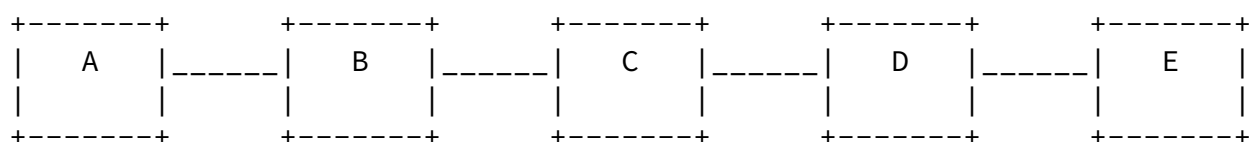
### [3.9.](#) Scenario 09: 4-hop bidirectional route establishment - Forward Route and Reverse Route initial installation

This test aims to verify the initial installation of a bidirectional

route (Forward Route and Reverse Route from A to E) within the LOADng Router routing tables (Routing Sets) through the effective forwarding of LOADng control traffic by LOADng Routers B, C and D, which are located between LOADng Router A and LOADng Router E. It is also verified that RREP-ACK messages are not forwarded by the LOADng Routers these messages are intended for.

### [3.9.1.](#) Scenario Topology

The testbed is composed of five LOADng Routers. Control traffic generated by either LOADng Router A towards LOADng Router E or LOADng Router E towards LOADng Router A has to be forwarded by LOADng Routers B, C and D:



This test suite consists in establishing a bidirectional route between LOADng Router A and LOADng Router E.

### [3.9.2.](#) Expected Message Sequencing

The expected message sequencing is as follows:

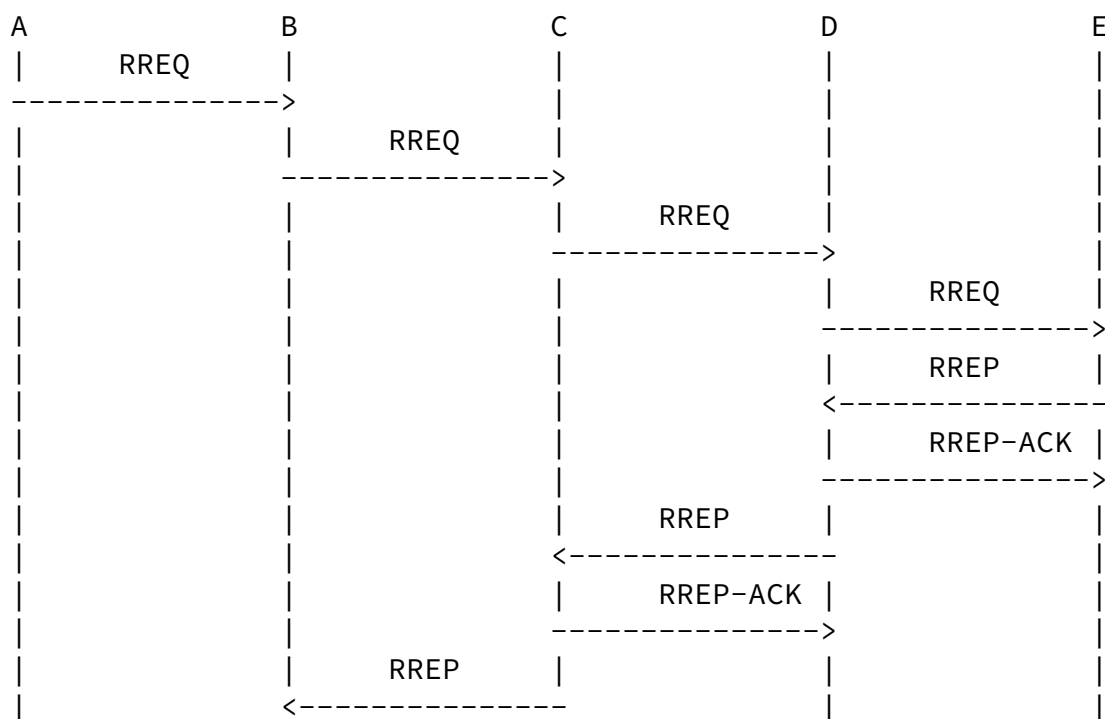
- o LOADng Router A generates an RREQ message intended for LOADng Router E.
- o Upon receiving the RREQ, LOADng Router B installs a new tuple in its Routing Set towards LOADng Router A (Reverse Route from LOADng Router B to LOADng Router A) and forwards the received RREQ.
- o Upon receiving the RREQ, LOADng Router C installs a new tuple in its Routing Set towards LOADng Router A (Reverse Route from LOADng Router C to LOADng Router A) and a new tuple towards LOADng Router B (Reverse route from LOADng Router C to LOADng Router B) and forwards the received RREQ.
- o Upon receiving the RREQ, LOADng Router D installs a new tuple in

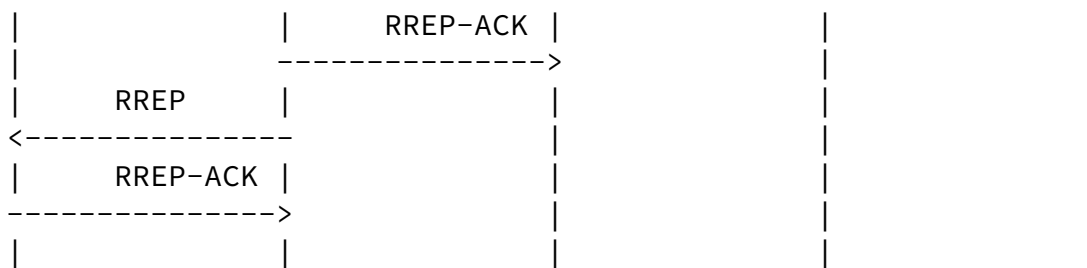
its Routing Set towards LOADng Router A (Reverse Route from LOADng Router D to LOADng Router A) and a new tuple towards LOADng Router C (Reverse route from LOADng Router D to LOADng Router C) and forwards the received RREQ.

- o Upon receiving the RREQ, LOADng Router E installs a new tuple in its Routing Set towards LOADng Router A (Reverse Route from LOADng Router E to LOADng Router A) and a new tuple towards LOADng Router D (Reverse route from LOADng Router E to LOADng Router D). The reception of the RREQ also triggers the generation of an unicast RREP message intended for LOADng Router A, soliciting an RREP-ACK message.
- o Upon receiving the RREP, LOADng Router D installs a new tuple in its Routing Set towards LOADng Router E (Forward Route from LOADng

Router D to LOADng Router E), sends an unicast RREP-ACK message to LOADng Router E and forwards the RREP received previously.

- o Upon receiving the RREP, LOADng Router C installs a new tuple in its Routing Set towards LOADng Router E (Forward Route from LOADng Router C to LOADng Router E) and a new tuple towards LOADng Router D (Forward Route from LOADng Router C to LOADng Router D). An unicast RREP-ACK message is also sent to LOADng Router D and the RREP received previously is forwarded.
- o Upon receiving the RREP, LOADng Router B installs a new tuple in its Routing Set towards LOADng Router E (Forward Route from LOADng Router B to LOADng Router E) and a new tuple towards LOADng Router C (Forward Route from LOADng Router B to LOADng Router C). An unicast RREP-ACK message is also sent to LOADng Router C and the RREP received previously is forwarded.
- o Upon receiving the RREP, LOADng Router A installs a new tuple in its Routing Set towards LOADng Router B (Forward Route from LOADng Router A to LOADng Router B) and a new tuple towards LOADng Router E (Forward Route from LOADng Router A to LOADng Router E). The reception of the RREP also triggers an unicast RREP-ACK message intended for LOADng Router B.



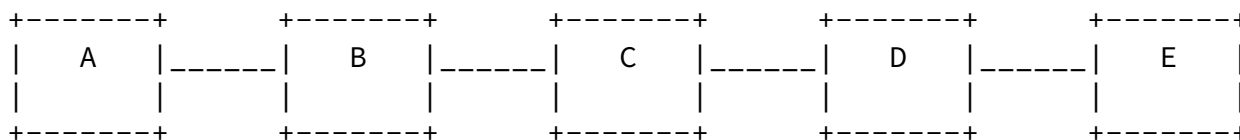


### 3.10. Scenario 10: 4-hop bidirectional route establishment - Link breakage handling

This test aims to verify the proper generation, processing and forwarding of a RERR message after an artificially created link breakage on an previously established bidirectional route.

#### 3.10.1. Scenario Topology

The testbed is composed of five LOADng Routers. Control traffic generated by either LOADng Router A towards LOADng Router E or LOADng Router E towards LOADng Router A has to be forwarded by LOADng Routers B, C and D:



This test suite consists in handling link breakages between routers.

#### 3.10.2. Expected Message Sequencing

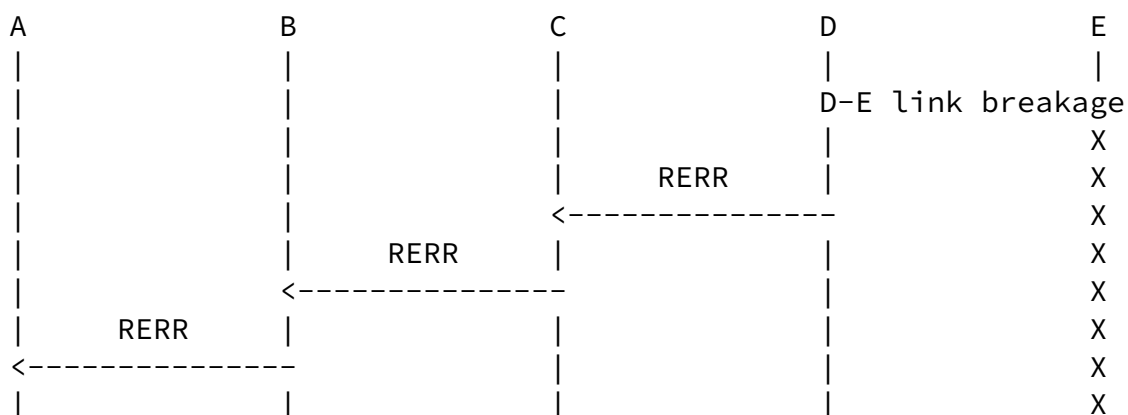
The expected message sequencing is as follows:

- o A bidirectional route is already established between LOADng Routers A and E.
- o At some time, a link breakage to E is detected by LOADng Router D. Consequently, an unicast RERR message intended for LOADng Router A (here the assumption is made that the unsuccessful delivered data traffic would have been generated by LOADng Router A) is transmitted to LOADng Router C according to the Reverse Route from

LOADng Router C to LOADng Router A computed previously.

Note: link breakage is provoked artificially and its detection by LOADng Router D is triggered manually (normally, this would be triggered by failure in sending data traffic intended for LOADng Router E).

- o Upon receiving the RERR, LOADng Router C updates its Routing Set by invalidating the existing Forward Route from LOADng Router C to LOADng Router E. Afterwards, the RERR message is forwarded according to the existing Reverse Route from LOADng Router C to LOADng Router A.
- o Upon receiving the RERR, LOADng Router B updates its Routing Set by invalidating the existing Forward Route from LOADng Router B to LOADng Router E. Afterwards, the RERR message is forwarded according to the existing Reverse Route from LOADng Router B to LOADng Router A.
- o Upon receiving the RERR, LOADng Router A updates its Routing Set by invalidating the existing Forward Route from LOADng Router A to LOADng Router E.



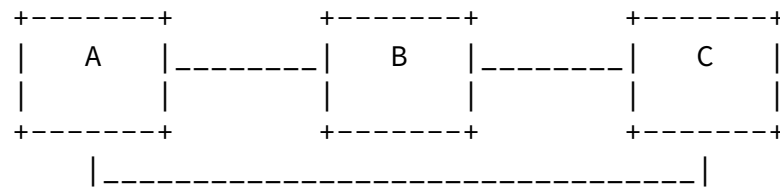
### [3.11.](#) Scenario 11: Establishment of the best bidirectional route

This test aims to verify the processing of multiple RREQs when installing a bidirectional route (Forward Route and Reverse Route from A to C) within the LOADng Router routing tables (Routing Sets).



### [3.11.1.](#) Scenario Topology

The testbed is composed of three LOADng Routers. Control traffic generated by either LOADng Router A towards LOADng Router C or LOADng Router C towards LOADng Router A can be forwarded by LOADng Router B or transmitted via the direct link between LOADng Routers A and C:



This test consists in establishing a bidirectional route between LOADng Router A and LOADng Router C. Hop count metric is used for measuring different routes.

### [3.11.2.](#) Expected Message Sequencing

The expected message sequencing is as follows:

- o LOADng Router A generates an RREQ message intended for LOADng Router C. According to RREQ transmission rules, the generated RREQ message is transmitted to all neighbor LOADng Routers.
- o Upon receiving the RREQ, LOADng Router B installs a new tuple in its Routing Set towards LOADng Router A (Reverse Route from LOADng Router B to LOADng Router A) and forwards the received RREQ.

At the same time, upon receiving the same RREQ via its direct link with LOADng Router A, LOADng Router C installs a new tuple in its Routing Set (Reverse Route from LOADng Router C to LOADng Router A). The reception of the RREQ also triggers the generation of an unicast RREP message intended for LOADng Router A, requiring RREP-ACK message.

- o Upon receiving the same RREQ via LOADng Router B, LOADng Router C compares the RREQ.route-metric information carried by the RREQ with the already existing tuple within its Routing Set (Reverse Route from LOADng Router C to LOADng Router A) according to the comparison operator specified by the metric used (the "hop count" metric was used). Thus, the best route is chosen considering only

the hop count:

Already existing tuple:

`<R_hop_count> = 1`

Tuple corresponding to the newly received RREQ:

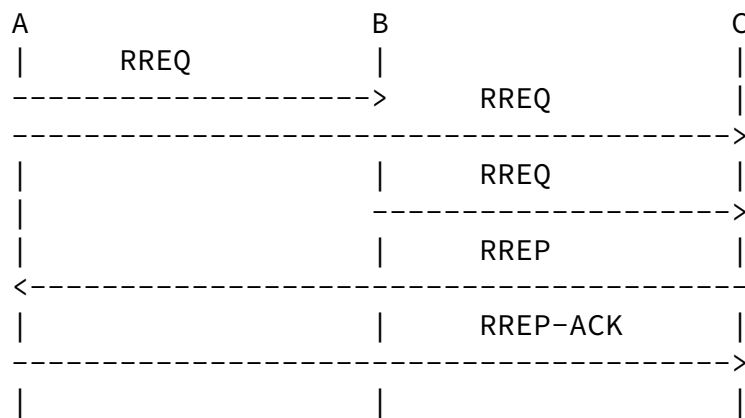
`<R_hop_count> = 2`

According to the comparison operator specified by the metric used:

`1 < 2`

Consequently, the newly received RREQ message is discarded without affecting the Routing Set or triggering the generation of any RREP message.

- o Upon receiving the RREP via its direct link with LOADng Router C, LOADng Router A installs a new tuple in its Routing Set (Forward Route from LOADng Router A to LOADng Router C). The reception of the RREP also triggers an unicast RREP-ACK message intended for LOADng Router C.



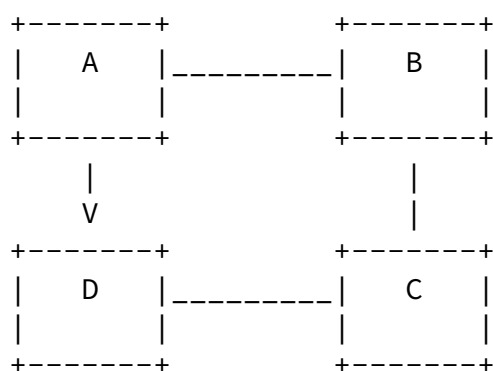
Note: the RREQ forwarded by LOADng Router B towards C is not necessarily received before LOADng Router C generates the RREP message intended for LOADng Router A. Indeed, the order in which those messages are transmitted is dependent on the transmission delays of each single link between LOADng Routers A, B and C.

### [3.12.](#) Scenario 12: Blacklisting

This test aims to verify the effectiveness of avoiding unidirectional links using blacklisting.

### [3.12.1.](#) Scenario Topology

The testbed is composed of four LOADng Routers with a unidirectional link between LOADng Routers A and D (direct communication from D towards A is impossible).



This test consists in establishing a bidirectional route between LOADng Router A and LOADng Router D.

### [3.12.2.](#) Expected Message Sequencing

First attempt to establish a bidirectional route between LOADng Routers A and D:

- o LOADng Router A generates an RREQ message (RREQ.seq-num = 0, RREQ.originator = A) intended for LOADng Router D.
- o Upon receiving the RREQ, LOADng Router B installs a new tuple in its Routing Set towards LOADng Router A (Reverse Route from LOADng Router B to LOADng Router A) and forwards the received RREQ.

At the same time, upon receiving the same RREQ via its direct (unidirectional) link with LOADng Router A, LOADng Router D installs a new tuple in its Routing Set towards LOADng Router A (Reverse Route from LOADng Router D to LOADng Router A). The reception of the RREQ also triggers the generation of an unicast RREP message intended for LOADng Router A. The RREP.ackrequired the sent RREP message is set ('1').

- o Upon receiving the RREQ, LOADng Router C installs a new tuple in its Routing Set towards LOADng Router A (Reverse Route from LOADng Router C to LOADng Router A).

Router C to LOADng Router A) and a new tuple towards LOADng Router B (Reverse route from LOADng Router C to LOADng Router B) and forwards the received RREQ.

- o Upon receiving the same RREQ (RREQ.seq-num = 0, RREQ.originator = A) again via LOADng Router C, LOADng Router D compares the

RREQ.route-metric information carried by the RREQ with the already existing tuple within its Routing Set (Reverse Route from LOADng Router D to LOADng Router A) according to the comparison operator specified by the metric used (hop count):

Already existing tuple:

$\langle R\_hop\_count \rangle = 1$

Tuple corresponding to the newly received RREQ:

$\langle R\_hop\_count \rangle = 2$

According to the comparison operator specified by the metric used:

$1 < 2$

Consequently, the newly received RREQ message is discarded without affecting the Routing Set or triggering the generation of any RREP message.

- o Due to the unidirectional nature of the existing link between LOADng Routers A and D, the RREP message previously sent by LOADng Router D intended for LOADng Router A did not reach its destination. After an elapsed time equaling RREP\_ACK\_TIMEOUT, LOADng Router D is not expecting an RREP-ACK message anymore. This results in recording LOADng Router A neighbor in LOADng Router D's Blacklist.

Second attempt to establish a bidirectional route between LOADng Routers A and D:

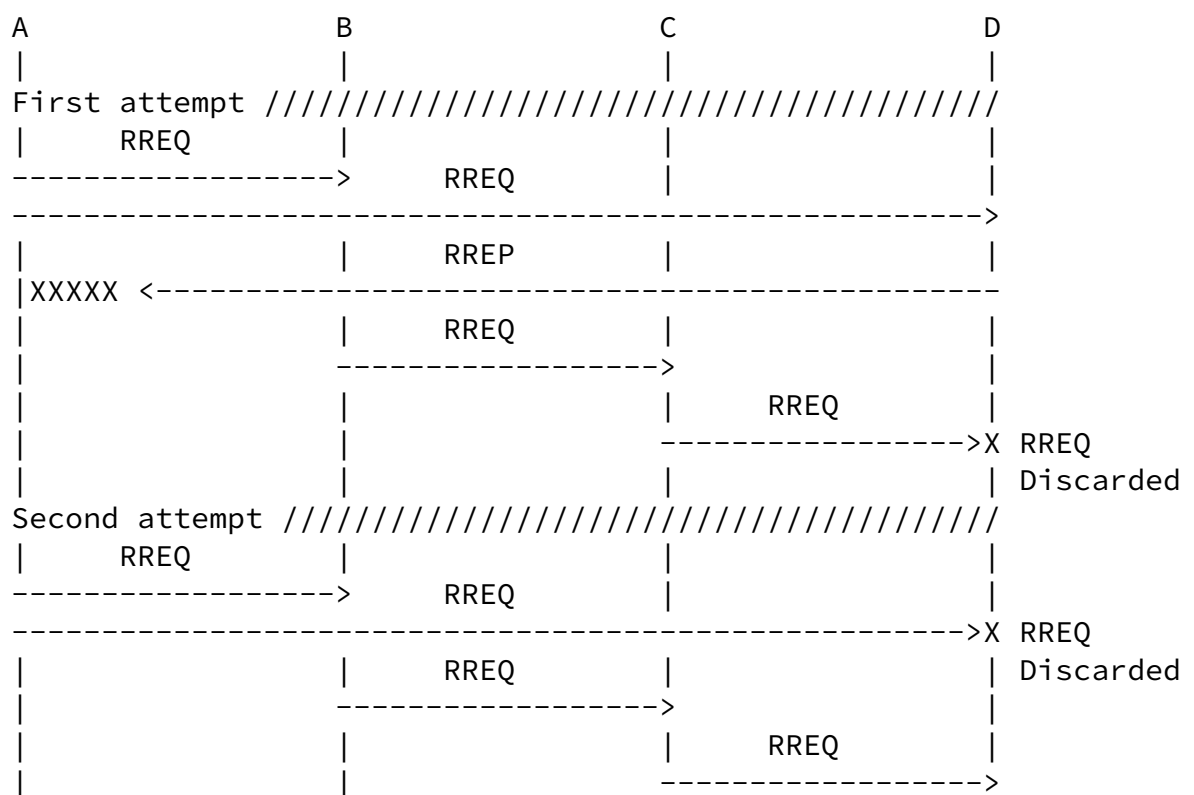
- o LOADng Router A generates an RREQ message (RREQ.seq-num = 1, RREQ.originator = A) intended for LOADng Router D.

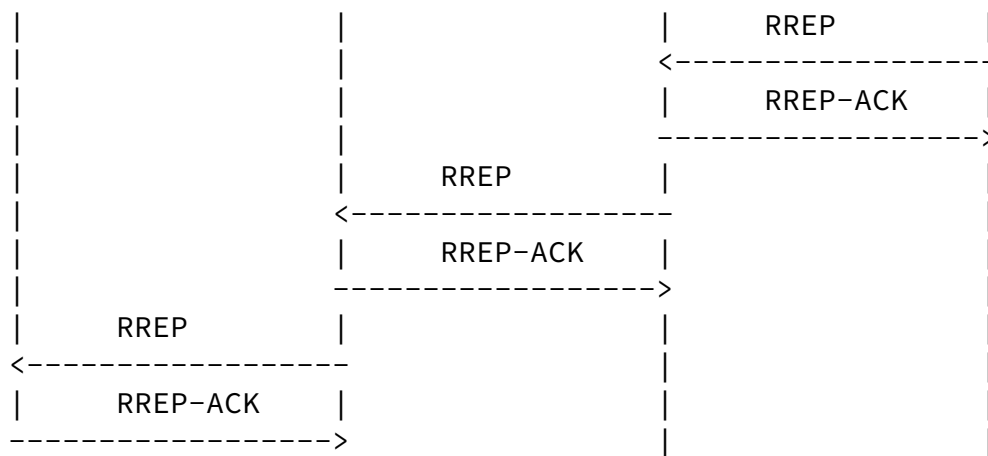
- o Upon receiving the RREQ, LOADng Router B installs a new tuple in its Routing Set towards LOADng Router A (Reverse Route from LOADng Router B to LOADng Router A) and forwards the received RREQ.

At the same time, upon receiving the same RREQ via its blacklisted neighbor LOADng Router A, LOADng Router D discards the message.

- o Upon receiving the RREQ, LOADng Router C updates the corresponding routes (Reverse Routes from LOADng Router C to LOADng Router A and from LOADng Router C to LOADng Router B) and forwards the received RREQ.

- o Upon receiving the RREQ, LOADng Router D updates the already installed route (Reverse Route from LOADng Router C to LOADng Router A) and installs a new tuple towards LOADng Router C (Reverse route from LOADng Router D to LOADng Router C). The reception of the RREQ also triggers the generation of an unicast RREP message intended for LOADng Router A. The RREP.ackrequired of the sent RREP message is set ('1').
- o Upon receiving the RREP, LOADng Router C installs a new tuple in its Routing Set towards LOADng Router D (Forward Route from LOADng Router C to LOADng Router D), sends an unicast RREP-ACK message to LOADng Router D and forwards the RREP received previously.
- o Upon receiving the RREP, LOADng Router B installs a new tuple in its Routing Set towards LOADng Router D (Forward Route from LOADng Router B to LOADng Router D) and a new tuple towards LOADng Router C (Forward Route from LOADng Router B to LOADng Router C). An unicast RREP-ACK message is also sent to LOADng Router C and the RREP received previously is forwarded.
- o Upon receiving the RREP, LOADng Router A installs a new tuple in its Routing Set towards LOADng Router D (Forward Route from LOADng Router A to LOADng Router D) and a new tuple towards LOADng Router B (Forward Route from LOADng Router A to LOADng Router B). The reception of the RREP also triggers an unicast RREP-ACK message intended for LOADng Router B.





#### [4.](#) Interop 01: Yokohama, Japan, October 2011

##### [4.1.](#) Version of LOADng Specification Tested

The interoperability tests were conducted according to the specification in [[LOADng-00](#)].

NOTE: Due to the evolution of [[LOADng](#)] and this document, some of the conventions used in [Section 3](#), such as routing metric and some fields of messages, may be different from the description in [[LOADng-00](#)].

##### [4.2.](#) Place and Date of Interoperability Test

This section reports experience with the LOADng routing protocol, resulting from interoperability testing performed at Hitachi YRL in Yokohama, Japan, from October 17th to October 19th 2011.

##### [4.3.](#) Participating Implementations

The following implementations were used to perform the interoperability tests in this section, listed alphabetically:

Ecole Polytechnique: "LIX" - This implementation was jointly developed by Axel Colin de Verdiere, Jiazi Yi, Ulrich Herberg and Thomas Clausen of Ecole Polytechnique's networking team. It

consists of approximately 6000 lines of JAVA code running in a Mac OS environment. It supports RREQ, RREP, RREP-ACK and RERR generation, processing, forwarding and transmission.

Hitachi YRL 1: "Hitachi 1" - This implementation was fully developed by Yuichi Igarashi of Hitachi YRL. It consists of 1589 lines of C code running in the Hitachi proprietary micro OS environment embedded in a 16MHz H8 micro processor. It supports RREQ, RREP, RREP-ACK and RERR generation, processing, forwarding and transmission.

Hitachi YRL 2: "Hitachi 2" - This implementation was jointly developed by Nobukatsu Inomata of Hitachi ULSI Systems and Yoko Morii of Hitachi YRL. It consists of 1987 lines of C++ code running in a Mac OS environment. It supports RREQ, RREP, RREP-ACK generation, processing, forwarding and transmission, and RERR processing.

#### [4.4.](#) Scenarios Tested

This interoperability test includes all scenarios 01-12 (inclusive).

#### [4.5.](#) Additional Interoperability Test Considerations

Wireshark packet sniffers, modified to interpret LOADng control traffic, were used to monitor each link, so as to verify proper message sequencing.

For each test, the initiation of the communication resulting in the generation of the first LOADng control traffic message is always triggered manually. In addition, RREP-ACK LOADng control messages were systematically expected from each LOADng Router upon reception of a RREP LOADng control message in order to allow the detection of unidirectional links.

#### [4.6.](#) Results For Scenario 01

The following table is summarizing the results obtained for the different combinations for which a 1-hop Forward Route and Reverse Route initial installation test was performed:

+-----+-----+-----+-----+



	LIX	Hitachi 1	Hitachi 2
LIX	N/R	Pass	Pass
Hitachi 1	Pass	N/R	Pass
Hitachi 2	Pass	Pass	N/R

Table 1

#### [4.7.](#) Results For Scenario 02

The following table is summarizing the results obtained for the different combinations for which a 1-hop Forward Route and Reverse Route updating test was performed:

	LIX	Hitachi 1	Hitachi 2
LIX	N/R	Pass	Pass
Hitachi 1	Pass	N/R	Pass
Hitachi 2	Pass	Pass	N/R

Table 2

#### [4.8.](#) Results For Scenario 03

The following table is summarizing the results obtained for the different combinations for which a 2-hop Forward Route and Reverse Route initial installation test was performed:

A	B	C	Result
Hitachi 1	LIX	Hitachi 2	Pass
Hitachi 2	LIX	Hitachi 1	Pass
LIX	Hitachi 1	Hitachi 2	Pass
Hitachi 2	Hitachi 1	LIX	Pass
LIX	Hitachi 2	Hitachi 1	Pass
Hitachi 1	Hitachi 2	LIX	Pass

Table 3

#### 4.9. Results For Scenario 04

The following table is summarizing the results obtained for the different combinations for which a 2-hop Forward Route and Reverse Route updating test was performed:

A	B	C	Result
Hitachi 1	LIX	Hitachi 2	Pass
Hitachi 2	LIX	Hitachi 1	Pass
LIX	Hitachi 1	Hitachi 2	Pass
Hitachi 2	Hitachi 1	LIX	Pass
LIX	Hitachi 2	Hitachi 1	Pass
Hitachi 1	Hitachi 2	LIX	Pass

Table 4

#### 4.10. Results For Scenario 05

The following table is summarizing the results obtained for the different combinations for which a Link breakage handling test was performed:

A	B	C	Result
Hitachi 1	LIX	LIX	Pass
LIX	Hitachi 1	LIX	Pass

Table 5

[4.11.](#) Results For Scenario 06

The following table is summarizing the results obtained for the different combinations for which a 3-hop Forward Route and Reverse Route initial installation test was performed:

A	B	C	D	Result
Hitachi 1	LIX	LIX	Hitachi 2	Pass
Hitachi 1	LIX	Hitachi 2	LIX	Pass
LIX	Hitachi 2	Hitachi 1	LIX	Pass

Table 6

[4.12.](#) Results For Scenario 07

The following table is summarizing the results obtained for the different combinations for which a 3-hop Forward Route and Reverse Route updating test was performed:

A	B	C	D	Result
Hitachi 1	LIX	LIX	Hitachi 2	Pass
Hitachi 1	LIX	Hitachi 2	LIX	Pass
LIX	Hitachi 2	Hitachi 1	LIX	Pass

Table 7

[4.13.](#) Results For Scenario 08

The following table is summarizing the results obtained for the different combinations for which a Link breakage handling test was performed:

A	B	C	D	Result
Hitachi 1	LIX	LIX	Hitachi 2	Pass
LIX	Hitachi 1	LIX	Hitachi 2	Pass

+-----+-----+-----+-----+-----+

Table 8

Clausen, et al.

Expires April 25, 2013

[Page 31]

Internet-Draft

LOADng Interop Report

October 2012

#### [4.14.](#) Results For Scenario 09

The following table is summarizing the results obtained for the different combinations for which a 4-hop Forward Route and Reverse Route initial installation test was performed:

A	B	C	D	E	Result
Hitachi 2	Hitachi 1	LIX	Hitachi 1	LIX	Pass

Table 9

#### [4.15.](#) Results For Scenario 10

The following table is summarizing the results obtained for the different combinations for which a Link breakage handling test was performed:

A	B	C	D	E	Result
Hitachi 2	Hitachi 1	LIX	Hitachi 1	LIX	Pass

Table 10

#### [4.16.](#) Results For Scenario 11

The following table is summarizing the results obtained for the different combinations for which a test consisting in the establishment of the best bidirectional route was performed:

A	B	C	Result
---	---	---	--------

LIX	Hitachi 1	Hitachi 2	Pass
LIX	Hitachi 2	Hitachi 1	Pass
Hitachi 2	Hitachi 1	LIX	Pass
Hitachi 1	LIX	Hitachi 2	Pass

Table 11

#### 4.17. Results For Scenario 12

The following table is summarizing the results obtained for the different combinations for which a Blacklisting test was performed:

A	B	C	D	Result
Hitachi 2	LIX	Hitachi 1	LIX	Pass
LIX	LIX	Hitachi 1	Hitachi 2	Pass
Hitachi 2	LIX	LIX	Hitachi 1	Pass

Table 12

#### 4.18. Conclusions

The different test scenarios carried that were carried out for different interoperable and independent implementations allowed to completely cover the [LOADng-00] specification by checking each technical feature one by one. In addition, the completion of this process permitted the improvement of the overall quality and accuracy of the [LOADng-00] specification.

Chap.	Item	Technical feature	Test suites					
			A	B	C	D	E	F

6.1			Originator	X X X	X X
+-----+	Information	Routing Set	+-----+	+-----+	+-----+
6.1	Base		Previous	X X X	X
+-----+					
6.2		Blacklist Neighbor set			X
+-----+					
8.1		TLV		X X X X X X	
+-----+					
8.2.1	Packet	Route Request Message	X X X X X X		
+-----+	Format				
8.2.1		Route Reply Message	X X X X X X		
+-----+					
8.2.2		Route Reply Ack Message	X X X X X X		
+-----+					
8.2.3		Route Error Message	X X X		
+-----+					
10.1	Unidirectional	Blacklist			X
	link handling				
+-----+					

11.1		Invalid RREQ, RREP	X X X X X X
+-----+	Common rules	+-----+	+-----+
11.2	for RREQ, RREP	RREQ, RREP Processing	X X X X X X
+-----+	Message	+-----+	+-----+
11.3		Updating RREQ, RREP	X X X X X X
+-----+		+-----+	+-----+
12.1		RREQ Generation	X X X X X X
+-----+		+-----+	+-----+
12.2	Route	RREQ Processing	X X X X X X
+-----+	Requests	+-----+	+-----+
12.3	(RREQs)	RREQ Forwarding	X X X X X X
+-----+		+-----+	+-----+
12.4		RREQ Transmission	X X X X X X
+-----+		+-----+	+-----+
13.1		RREP Generation	X X X X X X
+-----+		+-----+	+-----+
13.2	Route	RREP Processing	X X X X X X
+-----+	Replies	+-----+	+-----+
13.3	(RREPs)	RREP Forwarding	X X X X X X
+-----+		+-----+	+-----+
13.4		RREP Transmission	X X X X X X
+-----+		+-----+	+-----+

14.1		RERR Generation	X X X
+-----+		+-----+	+-----+
14.2	Route	RERR Processing	X X X
+-----+	Errors	+-----+	+-----+
14.3	(RERRs)	RERR Forwarding	X X
+-----+		+-----+	+-----+
14.4		RERR Transmission	X X X
+-----+		+-----+	+-----+
15.1		RREP-ACK Generation	X X X X X X
+-----+		+-----+	+-----+
15.2	Route	RREQ-ACK Processing	X X X X X X
+-----+	Reply	+-----+	+-----+
15.3	Acknowledgement	RREQ-ACK Forwarding	X X X X X X
+-----+	(RREP-ACKs)	+-----+	+-----+
15.4		RREQ-ACK Transmission	X X X X X X
+-----+		+-----+	+-----+
16	Metrics	Hop Count While	X X X X X X
		Avoiding Weak Links	
+-----+		+-----+	+-----+

Test suite A : 1-hop bidirectional route establishment (scenarios 01, 02)

Test suite B : 2-hop bidirectional route establishment (scenarios 03, 04, 05)

Test suite C : 3-hop bidirectional route establishment (scenarios 06, 07, 08)

Test suite D : 4-hop bidirectional route establishment (scenarios 09, 10)

Test suite E : Establishment of the best bidirectional route (scenario 11)

Test suite F : Blacklisting (scenario 12)

## [5.](#) Interop 02: San Jose, USA March 2012

### [5.1.](#) LOADng version tested

The interoperability tests were conducted according to the specification in [[LOADng-03](#)].

NOTE: Due to the evolution of [[LOADng](#)] and this document, one of the conventions used in [Section 3](#), such as routing metric and some fields of messages, may be different from the description in [[LOADng-03](#)].

## [5.2.](#) Place and Date of Interoperability Test

This section reports experience with the LOADng routing protocol, resulting from interoperability testing performed at Fujitsu Laboratories of America (FLA), San Jose, USA, on April 13, 2012.

## [5.3.](#) Participating Implementations

The following implementations were used to perform the interoperability tests this section, listed alphabetically:

Ecole Polytechnique: "LIX" - This implementation was jointly developed by Axel Colin de Verdiere, Jiazi Yi, Ulrich Herberg and Thomas Clausen of Ecole Polytechnique's networking team. It consists of approximately 6000 lines of JAVA code running in a Mac OS environment. It supports RREQ, RREP, RREP-ACK and RERR generation, processing, forwarding and transmission.

Fujitsu Laboratories of America: "FLA" - This implementation was developed by Ulrich Herberg from Fujitsu Laboratories of America. It is a Java implementation, supporting basic features (RREQ, RREP, RREP-ACK generation, processing, forwarding and transmission).

## [5.4.](#) Interoperability Test Considerations

As an intermediate test, only a subset of the scenarios described were tested (01, 03 and 05), for verifying interoperability bug-fixing the involved implementations.

## [5.5.](#) Results For Scenario 01



The following table is summarizing the results obtained for the different combinations for which a 1-hop Forward Route and Reverse Route initial installation test was performed:

	LIX	FLA
LIX	N/R	Pass
FLA	Pass	N/R

Table 13

#### 5.6. Results For Scenrio 03

The following table is summarizing the results obtained for the different combinations for which a 2-hop Forward Route and Reverse Route initial installation test was performed:

A	B	C	Result
LIX	FLA	LIX	Pass
LIX	LIX	FLA	Pass

Table 14

#### 5.7. Results For Scenario 05

The following table is summarizing the results obtained for the different combinations for which a Link breakage handling test was performed:

A	B	C	Result
LIX	FLA	LIX	Pass

Table 15

## [6.](#) Interop 03: Los Angeles, USA, June 2012

### [6.1.](#) Version of LOADng Specification Tested

The interoperability tests were conducted according to the specification in [[LOADng-04](#)].

NOTE: Due to the evolution of [[LOADng](#)] and this document, some of the conventions used in [Section 3](#), such as routing metric and some fields of messages, may be different from the description in [[LOADng-04](#)].

### [6.2.](#) Place and Date of Interoperability Test

This section reports experience with the LOADng routing protocol, resulting from interoperability testing performed at the Los Angeles Airport Hilton, USA, on June 6, 2012.

### [6.3.](#) Participating Implementations

The following implementations were used to perform the interoperability tests this section, listed alphabetically:

Ecole Polytechnique: "LIX" - This implementation was jointly developed by Axel Colin de Verdiere, Jiazi Yi, Ulrich Herberg and Thomas Clausen of Ecole Polytechnique's networking team. It consists of approximately 6000 lines of JAVA code running in a Mac OS environment. It supports RREQ, RREP, RREP-ACK and RERR generation, processing, forwarding and transmission.

Fujitsu Laboratories of America: "FLA" - This implementation was developed by Ulrich Herberg from Fujitsu Laboratories of America. It is a Java implementation, supporting basic features (RREQ, RREP, RREP-ACK generation, processing, forwarding and transmission).

### [6.4.](#) Scenarios Tested

This interoperability test includes scenarios 01-12 (inclusive).

### [6.5.](#) Additional Interoperability Test Considerations

Wireshark packet sniffers, that have been modified to interpret LOADng control traffic, were used to monitor each single underlying link.

For each test, the initiation of the communication resulting in the

generation of the first LOADng control traffic message is always triggered manually. In addition, RREP-ACK LOADng control messages were systematically expected from each LOADng Router upon reception of a RREP LOADng control message in order to allow the detection of unidirectional links.

#### [6.6.](#) Results For Scenario 01-02

The following table is summarizing the results obtained for the different combinations for which test 1 (Forward Route and Reverse Route initial installation) was performed:

+-----+-----+-----+		
	LIX	FLA
+-----+-----+-----+		
LIX	N/R	Pass
FLA	Pass	N/R
+-----+-----+-----+		

Table 16

The following table is summarizing the results obtained for the different combinations for which test 2 (Forward Route and Reverse Route updating) was performed:

+-----+-----+-----+		
	LIX	FLA
+-----+-----+-----+		
LIX	N/R	Pass
FLA	Pass	N/R
+-----+-----+-----+		

Table 17

#### [6.7.](#) Results For Scenario 03-04-05

The following table is summarizing the results obtained for the different combinations for which these test 1 (Forward Route and Reverse Route initial installation) and test 2 (Forward Route and Reverse Route updating) were performed:

A	B	C	Test 1	Test 2
LIX	FLA	LIX	Pass	Pass
LIX	LIX	FLA	Pass	Pass
FLA	LIX	LIX	Pass	Pass

Table 18

The following table is summarizing the results obtained for the different combinations for which these test 3 (Link breakage handling) was performed:

A	B	C	Test 3
FLA	LIX	LIX	Pass
LIX	FLA	LIX	Pass

Table 19

#### [6.8.](#) Results For Scenario 06-07-08

The following table is summarizing the results obtained for the different combinations for which these test 1 (Forward Route and Reverse Route initial installation) and test 2 (Forward Route and Reverse Route updating) were performed:

A	B	C	D	Test 1	Test 2
LIX	FLA	LIX	LIX	Pass	Pass
LIX	LIX	FLA	LIX	Pass	Pass
FLA	LIX	LIX	LIX	Pass	Pass
LIX	LIX	LIX	FLA	Pass	Pass

Table 20

The following table is summarizing the results obtained for the different combinations for which these test 3 (Link breakage handling) was performed:

Clausen, et al.

Expires April 25, 2013

[Page 39]

Internet-Draft

LOADng Interop Report

October 2012

A	B	C	D	Test 3
FLA	LIX	LIX	LIX	Pass
LIX	LIX	LIX	FLA	Pass

Table 21

#### [6.9.](#) Results For Scenario 09-10

The following table is summarizing the results obtained for the different combinations for which test 1 (Forward Route and Reverse Route initial installation) and test 2 (Link breakage handling) were performed:

A	B	C	D	E	Test 1	Test 2
FLA	FLA	LIX	LIX	LIX	Pass	Pass
LIX	LIX	LIX	FLA	FLA	Pass	Pass

Table 22

#### [6.10.](#) Results For Scenario 11

The following table is summarizing the results obtained for the different combinations for which this test was performed:

A	B	C	Result
---	---	---	--------

	LIX		FLA		LIX		Pass	
	LIX		LIX		FLA		Pass	
	FLA		LIX		LIX		Pass	

Table 23

## 6.11. Conclusions

The different test scenarios carried that were carried out for different interoperable and independent implementations allowed to cover all major features of the LOADng specification by checking each technical feature one by one. In addition, the completion of this process permitted the improvement of the overall quality and accuracy of the [LOADng] specification.

Clausen, et al.

Expires April 25, 2013

[Page 40]

Internet-Draft

LOADng Interop Report

October 2012

## 7. Security Considerations

This document does currently not specify any security considerations.

## 8. IANA Considerations

This document has no actions for IANA.

## 9. Contributors

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

- o Alberto Camacho, LIX, France, <alberto@albertocamacho.com>
- o Thomas Heide Clausen, LIX, France, <T.Clausen@computer.org>
- o Axel Colin de Verdiere, LIX, France, <axel@axelcdv.com>
- o Ulrich Herberg, Fujitsu Laboratories of America, USA, <ulrich.herberg@us.fujitsu.com>
- o Yuichi Igarashi, HITACHI YRL, Japan, <yuichi.igarashi.hb@hitachi.com>

- o Nobukatsu Inomata, HITACHI ULSI Systems, Japan, <nobukatsu.inomata.rf@hitachi.com>
- o Yoko Morii, HITACHI YRL, Japan, <yoko.morii.cs@hitachi.com>
- o Hiroki Satoh, HITACHI YRL, Japan, <hiroki.satoh.yj@hitachi.com>
- o Jiazi Yi, LIX, France, <jiazi@jiaziyi.com>

## [10.](#) Acknowledgments

TBD

## [11.](#) References

### [11.1.](#) Normative References

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

Clausen, et al.

Expires April 25, 2013

[Page 41]

---

Internet-Draft

LOADng Interop Report

October 2012

### [11.2.](#) Informative References

- [LOADng] Clausen, T., Colin de Verdiere, A., Yi, J., Niktash, A., Igarashi, Y., Satoh, H., Herberg, U., Lavenue, C., Lys, T., and C. Perkins, "The LLN On-demand Ad hoc Distance-vector Routing Protocol - Next Generation (LOADng)", [draft-clausen-lln-loadng](#) (work in progress), July 2012.
- [LOADng-00] Clausen, T., Colin de Verdiere, A., Yi, J., Lavenue, C., Lys, T., Niktash, A., Igarashi, Y., and H. Satoh, "The LLN On-demand Ad hoc Distance-vector Routing Protocol - Next Generation (LOADng)", [draft-clausen-lln-loadng-00](#) (work in progress), October 2011.
- [LOADng-03] Clausen, T., Colin de Verdiere, A., Yi, J., Niktash, A., Igarashi, Y., Satoh, H., Herberg, U., Lavenue, C., and T. Lys, "The LLN On-demand Ad hoc Distance-vector Routing Protocol - Next Generation (LOADng)",

[draft-clausen-lln-loadng-03](#) (work in progress),  
March 2012.

[LOADng-04] Clausen, T., Colin de Verdiere, A., Yi, J., Niktash, A., Igarashi, Y., Satoh, H., Herberg, U., Lavenu, C., and T. Lys, "The LLN On-demand Ad hoc Distance-vector Routing Protocol - Next Generation (LOADng)", [draft-clausen-lln-loadng-04](#) (work in progress), April 2012.

[LOADng-05] Clausen, T., Colin de Verdiere, A., Yi, J., Niktash, A., Igarashi, Y., Satoh, H., Herberg, U., Lavenu, C., Lys, T., and C. Perkins, "The LLN On-demand Ad hoc Distance-vector Routing Protocol - Next Generation (LOADng)", [draft-clausen-lln-loadng-05](#) (work in progress), July 2012.

#### Authors' Addresses

Thomas Heide Clausen  
LIX, Ecole Polytechnique

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

Clausen, et al.

Expires April 25, 2013

[Page 42]

---

Internet-Draft

LOADng Interop Report

October 2012

Alberto Camacho  
LIX, Ecole Polytechnique

Phone: +34 636 309 835  
EMail: alberto@albertocamacho.com  
URI: <http://www.albertocamacho.com/>

Jiazi Yi  
LIX, Ecole Polytechnique



Phone: +33 1 6933 4031  
EMail: [jiazi@jiaziyi.com](mailto:jiazi@jiaziyi.com)  
URI: <http://www.jiaziyi.com/>

Axel Colin de Verdier  
LIX, Ecole Polytechnique

Phone: +33 6 1264 7119  
EMail: [axel@axelcdv.com](mailto:axel@axelcdv.com)  
URI: <http://www.axelcdv.com/>

Yuichi Igarashi  
Hitachi, Ltd., Yokohama Research Laboratory

Phone: +81 45 860 3083  
EMail: [yuichi.igarashi.hb@hitachi.com](mailto:yuichi.igarashi.hb@hitachi.com)  
URI: <http://www.hitachi.com/rd/yrl/index.html>

Hiroki Satoh  
Hitachi, Ltd., Yokohama Research Laboratory

Phone: +81 44 959 0205  
EMail: [hiroki.satoh.yj@hitachi.com](mailto:hiroki.satoh.yj@hitachi.com)  
URI: <http://www.hitachi.com/rd/yrl/index.html>

Yoko Morii  
Hitachi, Ltd., Yokohama Research Laboratory

Phone: +81 45 860 3083  
EMail: [yoko.morii.cs@hitachi.com](mailto:yoko.morii.cs@hitachi.com)  
URI: <http://www.hitachi.com/rd/yrl/index.html>

E-Mail: [ulrich@herberg.name](mailto:ulrich@herberg.name)  
URI: <http://www.herberg.name/>

Cedric Lavenu  
EDF R&D

Phone: +33 1 4765 2729  
E-Mail: [cedric-2.lavenu@edf.fr](mailto:cedric-2.lavenu@edf.fr)  
URI: <http://www.edf.fr/>