T. Clausen
LIX, Ecole Polytechnique, France
C. Dearlove
BAE Systems Advanced Technology
Centre
J. Dean
Naval Research Laboratory
C. Adjih
INRIA Rocquencourt
July 28, 2006

Generalized MANET Packet/Message Format
draft-ietf-manet-packetbb-02

Status of this Memo

Copyright Notice

Abstract

This document specifies a multi-message packet format that may be used by mobile ad hoc network routing and other protocols.

Table of Contents

1. Introduction

Signaling in a MANET (mobile ad hoc network) routing protocol consists, mainly, of stating IP addresses and attributes associated with such IP addresses. Since this is a task common to many such protocols, this specification presents a generalized packet format, suitable for signaling. This format may be employed both by mobile ad hoc network routing protocols and by other protocols with similar requirements.

This document is a specification of a multi-message packet format. Messages encapsulate protocol information (including addresses and their attributes), and are themselves are encapsulated in packets. The structure of addresses, attributes, messages, and packets is specified via regular expressions. This specification gives the syntax, but not the interpretation, of the information carried within a packet, other than the header information which may be used to control the dissemination of the messages. Packets are intended to be encapsulated by a suitable transport protocol, typically UDP, and carried over IP (IPv4 or IPv6).

This document specifies:

- o A packet format, allowing zero or more messages to be contained within a single transmission, and optionally including a packet header.
- o A message format, where a message is composed of a message header and a message body.
- o A message header format containing *all* necessary information to allow a node to make forwarding decisions without inspecting and processing the message body. Message header information permits single- and multi-hop message diffusion.
- o A message body format, containing attributes associated with the message or the originator of the message, as well as blocks of addresses with associated attributes.
- o An address block format, where an address block represents sets of addresses in a compact (compressed) form.
- o A generalized type-length-value (TLV) format representing attributes. Multiple TLVs can be included and associated with a packet, a message, an address, or a set of addresses.

The specification has been explicitly designed with the following properties in mind:

Parsing logic - the regular expression specification facilitates generic, protocol independent, parsing logic.

Extensibility - packets and messages defined by a protocol using this specification are extensible through defining new message types and new TLVs. Full backward compatibility can be maintained.

Efficiency - when reported addresses share common bit sequences (e.g. prefixes or IPv6 interface identifiers) the address block representation allows for a compact representation.

Separation of forwarding and processing - duplicate detection and controlled scope message forwarding decisions can be made solely using information contained in the message header, without processing the message body.

2. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119. [1].

Additionally, this document uses the following terminology:

Packet - the top level entity in this specification. Packets are transmitted hop-by-hop and are not forwarded. A packet contains zero or more messages, and may contain a packet header.

Message - the fundamental entity carrying protocol information, in the form of addresses and TLVs. Messages are transmitted in packets, and may be forwarded based on their header information.

Address - an address of the same type and length as the source IP address in the IP datagram carrying the packet.

TLV - a Type-Length-Value structure. This is a generic way in which an attribute can be represented and correctly parsed, without the parser having to understand the attribute.

element - a syntactic entity defined in the regular expression specification, represented using the notation <foo>.

<foo> - if <foo> is an 8 or 16 bit field then <foo> is also used to represent the value of that field.

? - zero or one occurrences of the preceding element.

* - zero or more occurrences of the preceding element.

+ - one or more occurrences of the preceding element.

bar - a variable, usually obtained through calculations based on the value(s) of field(s). Variables are introduced in the specification solely as a means to clarify the description.

address-length - a variable whose value is the length of an address in octets, it is 4 if using IPv4, 16 if using IPv6.

3. Applicability Statement

This specification describes a generic multi-message packet format, for carrying MANET routing protocol signals. The specification has been developed from that used by OLSR (The Optimized Link State Routing Protocol) [4].

The specification is designed specifically with IP (IPv4/IPv6) in mind. All addresses within a control message are assumed to be of the same size, deduced from IP. In the case of mixed IPv6 and IPv4 addresses, IPv4 addresses are carried in IPv6 as specified in [2].

The packets defined by this specification may use any transport protocol appropriate to the protocol using this specification. When the diffusion mechanism enabled by this specification is employed, UDP may be most appropriate.

This specification is particularly appropriate for extensible protocols. It offers external extensibility in the form of new message types. It offers internal extensibility in the form of TLVs, which may be added to existing message types.

4. Protocol Overview and Functioning

This specification does not describe a protocol. It describes a packet format, which may be used by any mobile ad hoc network routing or other protocol.

5. Signaling Framework

This section provides syntactical specification of a packet, represented by the element <packet> and the elements from which it is composed. The specification is given in the form of regular expressions. Illustrations of specified elements are given in Appendix A.

The length of a <packet> is obtained as the size of the payload of the transport protocol employed.

5.1. Packets

<packet> is defined by:

```
<packet> = {<packet-header><pad-octet>*}?
           {<message><pad-octet>*}*
```

where <message> is defined in Section 5.2, and <pad-octet> is defined in Section 5.4. The packet is parsed until all octets are used.

<packet-header> is defined by:

```
<packet-header> = <zero>
                  <packet-semantics>
                  <packet-seq-number>?
                  <tlv-block>?
```

where:

<zero> is an 8 bit field with all bits cleared ('0'). This field serves to identify that the packet starts with a packet header.

<packet-semantics> is an 8 bit field, specifying the composition of the packet header:

bit 0 (pnoseqnum): if cleared ('0'), then the packet header contains a <packet-seq-number>. If set ('1'), then the packet header does not include a <packet-seq-number>.

bit 1 (ptlv): if cleared ('0'), then the packet header does not include a TLV block. If set ('1'), then the packet header includes a TLV block.

bits 2-7: are RESERVED, and MUST each be cleared ('0') to be in conformance with this version of the specification.

<packet-seq-number> is omitted if the pnoseqnum bit is set ('1'), otherwise is a 16 bit field, specifying a packet sequence number.

<tlv-block> is omitted if the ptlv bit is cleared ('0'), and is otherwise defined in Section 5.3.

Note that since the message type zero is reserved (see Section 7), the presence or absence of a packet header can be determined by inspecting the first octet of the packet.

5.2. Messages

Information is carried through messages. Messages contain:

- A message header.
- A message TLV block that contains zero or more TLVs, associated with the whole message.
- Zero or more address blocks, each containing one or more addresses.
- A TLV block, containing zero or more TLVs, following each address block.

<message> is defined by:

```
<message>         = <msg-header>
                    <tlv-block>
                    {<addr-block><tlv-block>}*

<msg-header>      = <msg-type>
                    <msg-semantics>
                    <msg-size>
                    <msg-header-info>

<msg-header-info> = <originator-address>?
                    <hop-limit>?
                    <hop-count>?
                    <msg-seq-number>?
```

where:

<tlv-block> is defined in Section 5.3.

<addr-block> is defined in Section 5.2.1.

<msg-type> is an 8 bit field, specifying the type of message. A type with all bits cleared ('0') MUST NOT be used. The two most significant bits have the following semantics:

bit 7 (msguser): message types with this bit cleared ('0') are defined in this specification or can be allocated via standards action. Message types with this bit set ('1') are reserved for private/local use.

bit 6 (msgprot): for message types with the msg-user bit cleared ('0'), this bit specifies, if cleared ('0'), that the message type is protocol independent, i.e. is not specific to any one protocol, or, if set ('1'), that the message type is specific to the protocol for which it is defined.

<msg-semantics> is an 8 bit field, specifying the interpretation of the remainder of the message header:

bit 0 (noorig): if cleared ('0'), then <originator-address> and <msg-seq-number> are included in <msg-header-info>. If set ('1'), then <originator-address> and <msg-seq-number> are not included in <msg-header-info>; this reduced message header does not provide for duplicate suppression.

bit 1 (nohops): if cleared ('0'), then <hop-limit> and <hop-count> are included in the <msg-header-info>. If set ('1'), then <hop-limit> and <hop-count> are not included in the <msg-header-info>; this reduced message header does not provide for scope-delimited forwarding.

bit 2 (typedep): if cleared ('0'), then the message sequence number in the message is type-independent. If set ('1'), then the message sequence number contained in the message is type dependent (the message originator maintains a sequence number specific to <msg-type>). This bit MUST be cleared ('0') if the noorig bit is set ('1').

bits 3-7: are RESERVED and MUST each be cleared ('0') to be in conformance with this version of the specification.

<msg-size> is a 16 bit field, specifying the size of the <message>, counted in octets.

<originator-address> is an identifier of length equal to address-length, which serves to uniquely identify the node that originated the message.

<hop-limit> is an 8 bit field, which contains the maximum number of hops a message should be further transmitted.

<hop-count> is an 8 bit field, which contains the number of hops a message has traveled.

<msg-seq-number> is a 16 bit field, which contains a unique number, generated by the originator node. The <originator-address>, <msg-seq-number>, and, if the typedep bit in the <msg-semantics> field is set, the <msg-type> of a message serves to uniquely identify the message in the network.

5.2.1. Address Blocks

An address is specified as a sequence of octets of the form head:mid:tail. An address block is an ordered set of addresses sharing the same head and tail, and having individual mids.

<address block> is defined by:

```
<address-block> = <num-addr>
                  <head-octet>
                  <head>?
                  <tail-octet>?
                  <tail>?
                  <mid>*
```

where:

<num-addr> is an 8 bit field containing the number of addresses represented in the address block, which MUST NOT be zero.

<head-octet> is an 8 bit field, where:

bits 0-6: contain the length of the head. The corresponding variable head-length is calculated by:

```
head-length = <head-octet> & 127
```

bit 7 (notail): if cleared ('0') then the address block contains a <tail-octet>. If set ('1') then no <tail-octet> is included.

<head> is omitted if head-length == 0, otherwise it is a field of the head-length leftmost octets of all the addresses.

<tail-octet> is omitted if the notail bit is set ('1'), otherwise it is an 8 bit field, where:

bits 0-6: contain the length of the tail. The corresponding variable tail-length is calculated by:

tail-length = <tail-octet> & 127

bit 7 (zerotail): if cleared ('0'), then a <tail> is included. If set ('1') then no <tail> is included, and the tail-length rightmost octets of each address in the block are zero-valued.

If the <tail-octet> is omitted then tail-length = 0.

<tail> is omitted if tail-length == 0 or the zerotail bit is set ('1'), otherwise it is a field of the head-length leftmost octets of all the addresses.

mid-length is a variable, which MUST be non-negative, calculated by:

mid-length = address-length - head-length - tail-length

<mid> is omitted if mid-length == 0, otherwise each <mid> is a field of length mid-length octets, representing the mid of the corresponding address in the address block.

5.3. TLVs and TLV Blocks

A TLV is defined by:

```
<tlv-block>       = <tlv-length>
                    <tlv>*
```

where:

<tlv-length> is a 16 bit field, which contains the total length (in octets) of the immediately following <tlv>s.

<tlv> is defined in Section 5.3.1.

5.3.1. TLVs

There are three kinds of TLV, each represented by an element <tlv>:

- o A packet TLV, included in a packet header.

- o A message TLV, included in a message before all address blocks.

- o An address block TLV, included in a TLV block following an address block. An address block TLV applies to:

  * all addresses in the address block; OR

  * any continuous sequence of addresses in the address block; OR

  * a single address in the address block.

<tlv> is defined by:

```
 <tlv> = <tlv-type>
         <tlv-semantics>
         <index-start>?
         <index-stop>?
         <length>?
         <value>?
```

where:

<tlv-type> is an 8 bit field, specifying the type of the TLV. The two most significant bits have the following semantics:

bit 7 (tlvuser): TLV types with this bit cleared ('0') are defined in this specification or can be allocated via standards action. TLV types with this bit set ('1') are reserved for private/ local use.

bit 6 (tlvprot): for TLV types with the tlv-user bit cleared ('0'), this bit specifies, if cleared ('0'), that the TLV type is protocol independent, i.e. is not specific to any one protocol, or, if set ('1'), that the TLV type is specific to the protocol for which it is defined.

<tlv-semantics> is an 8 bit field specifying the interpretation of the remainder of the TLV:

bit 0 (extended) and bit 1 (novalue): must not both be set ('1'). Otherwise, they are interpreted according to Table 1.

extended	novalue	length	value
0	0	8 bits	included
0	1	not included	not included
1	0	16 bits	included

Table 1

bit 2 (noindex) and bit 3 (singleindex): must not both be set ('1'). Otherwise, they are interpreted according to Table 2.

noindex	singleindex	<index-start>	<index-stop>
0	0	included	included
0	1	included	not included
1	0	not included	not included

Table 2

bit 4 (multivalue): this bit serves to specify how the value field is interpreted as specified below. This bit MUST be cleared ('0') for packet or message TLVs, if the singleindex bit is set ('1'), or if the novalue bit is set ('1').

bits 5-7: are RESERVED and MUST each be cleared ('0') to be in accordance with this version of the specification.

<index-start> and <index-stop> are each an 8 bit field, interpreted as follows:

index-start and index-stop are variables, defined according to Table 3. The variable end-index is calculated as follows:

+ For message and packet TLVs:

  - end-index = 0

+ For address block TLVs:

- end-index = <num-addr> - 1

noindex	singleindex	index-start =	index-stop =
0	0	<index-start>	<index-stop>
0	1	<index-start>	<index-start>
1	0	0	end-index

Table 3

For an address block TLV, the TLV applies to the addresses from position index-start to position index-stop (inclusive) in the address block.

number-values is a variable, calculated by:

number-values = index-stop - index-start + 1

<length> is omitted or is a 8 or 16 bit field according to Table 1. If the multivalue bit is set ('1') then <length> MUST be an integral multiple of number-values, and the variable single-length is calculated by:

single-length = <length> / number-values

if the multivalue bit is cleared ('0'), the variable single-length is defined by:

single-length = <length>

<value> if present (see Table 1), this is a field of length <length> octets. In an address block TLV, <value> is associated with the addresses from index-start to index-stop, inclusive. If the multivalue bit is cleared ('0') then the whole of this field is associated with each of the indicated addresses. If the multivalue bit is set ('1') then this field is divided equally into number-values fields, each of length single-length octets and these are associated, in order, with the indicated addresses.

5.3.2. Constraints

TLVs in the same tlv block MUST be sorted in ascending TLV type order.

Two or more TLVs of the same type associated with the same address block MUST NOT both cover any address.

TLVs of the same type associated with the same address block MUST be sorted in ascending index-start order.

5.4. Padding

Packet headers and messages can be padded to ensure 32 bit alignment of each message contained within the packet and of the overall packet length.

All syntactical elements are an integer multiple of octets, hence padding can be accomplished by inserting an integer number of <pad-octets> after the syntactical element that is to be 32 bit aligned.

The number of <pad-octet>s required to achieve this 32 bit alignment is calculated as the smallest number (0 to 3) that, when added to the size of the preceding elements produces an integer multiple of 4.

<pad-octet> is an 8 bit field with all bits cleared ('0').

There is no need to indicate if padding is included, since a <pad-octet> will always precede either a message or the end of the packet. In the former case, the start of a message is indicated by the next non-zero octet parsed.

The padding after a message may be freely changed when a message is forwarded without affecting the message.

6. TLV specification

This document specifies one address block TLV, which is included to allow a standardized way of representing network addresses.

6.1. Address Block TLV Specification

Name	Type	Length	Value
PREFIX_LENGTH	0	8 bits	Indicates that the address is a network address, rather than a host address. The value is the length of the prefix/netmask.

Table 4

An address in an address block without an associated PREFIX_LENGTH TLV may be considered to have a prefix length equal to the address length (in bits).

7. IANA Considerations

A new registry for message types must be created with initial assignments as specified in Table 5. Future values in the range 5-127 of the Message Type can be allocated using standards action [3]. Additionally, values in the range 128-255 are reserved for private/local use.

A new registry for packet TLV types must be created, with no initial assignments. Future values in the range 0-127 of the Message Type can be allocated using standards action [3]. Additionally, values in the range 128-255 are reserved for private/local use.

A new registry for message TLV types must be created with no initial assignments. Future values in the range 0-127 of the Message Type can be allocated using standards action [3]. Additionally, values in the range 128-255 are reserved for private/local use.

A new registry for address block TLV types must be created with initial assignments as specified in Table 6. Future values in the range 1-127 of the Message Type can be allocated using standards action [3]. Additionally, values in the range 128-255 are reserved for private/local use.

Value	Description
0	MUST NOT be allocated.
1-4	RESERVED

Table 5

Message type 0 MUST NOT be allocated because a zero-octet signifies a packet header and zero-octets are used for padding. Message types 1 to 4 are reserved because they are used by OLSR [4], which uses a compatible packet/message header format.

Mnemonic	Value	Description
PREFIX_LENGTH	0	Indicates that associated addresses are network addresses, with given prefix length.

Table 6

8. Security Considerations

Packets are designed to be transmitted only one hop, and not forwarded. Hop-by-hop packet level security MAY be implemented, between nodes with an existing security association, by including a suitable packet TLV containing a cryptographic signature to the packet. Since packets are received as transmitted, signatures can be calculated based on the entire packet content, or on parts thereof as appropriate.

Messages at each hop MAY be forwarded and/or processed, according to the information in the message header and the protocol employing this specification. With immutable messages, end-to-end security MAY be implemented, between nodes with an existing security association, by including a suitable message TLV containing a cryptographic signature to the message. Since <hop-count> and <hop-limit> are the only fields that may be modified when such a message is forwarded, signatures can be calculated based on the entire message, including the message header with the <hop-count> and <hop-limit> fields set to zero ('0').

9. References

9.1. Normative References

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

[2] Hinden, R. and S. Deering, "Internet Protocol Version 6 (IPv6) Addressing Architecture", RFC 3513, April 2003.

[3] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", October 1998.

9.2. Informative References

[4] Clausen, T. and P. Jacquet, "The Optimized Link State Routing Protocol", RFC 3626, October 2003.

Appendix A. Illustrations

This informative appendix illustrates the elements, which are normatively specified in Section 5 using regular expressions.

Bits labeled Reserved or Resv are cleared ('0'). Bits labeled N and M may be cleared ('0') or set ('1'). Octets labeled Padding are cleared ('0'), and are optional.

Appendix A.1. Packet

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|0 0 0 0 0 0 0 0| Reserved  |0|0|    Packet Sequence Number     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                       Message + Padding                       |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
:                              ...                              :
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                       Message + Padding                       |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|0 0 0 0 0 0 0 0| Reserved  |0|1|            Padding            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                       Message + Padding                       |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
:                              ...                              :
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                       Message + Padding                       |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

```
     0                   1                   2                   3
     0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |0 0 0 0 0 0 0 0| Reserved  |1|0|    Packet Sequence Number     |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                                                               |
    |                        Packet TLV Block                       |
    |                                                               |
    |                               +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                               |            Padding            |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                                                               |
    |                        Message + Padding                      |
    |                                                               |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                                                               |
    :                               ...                             :
    |                                                               |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                                                               |
    |                        Message + Padding                      |
    |                                                               |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

```
     0                   1                   2                   3
     0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |0 0 0 0 0 0 0 0| Reserved  |1|1|                               |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+                               |
    |                                                               |
    |                        Packet TLV Block                       |
    |                                                               |
    |                               +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                               |            Padding            |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                                                               |
    |                        Message + Padding                      |
    |                                                               |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                                                               |
    :                               ...                             :
    |                                                               |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                                                               |
    |                        Message + Padding                      |
    |                                                               |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

```
     0                   1                   2                   3
     0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                                                               |
    |                       Message + Padding                       |
    |                                                               |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                                                               |
    :                              ...                              :
    |                                                               |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                                                               |
    |                       Message + Padding                       |
    |                                                               |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Appendix A.2. Message and Padding

```
     0                   1                   2                   3
     0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    | Message Type  |  Resv   |N|0|0|         Message Size          |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                      Originator Address                       |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |   Hop Limit   |   Hop Count   |    Message Sequence Number    |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                                                               |
    |                         Message Body                          |
    |                                                               |
    |                               +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                               |            Padding            |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+


     0                   1                   2                   3
     0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    | Message Type  |  Resv   |0|0|1|         Message Size          |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |   Hop Limit   |   Hop Count   |                               |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+                               |
    |                                                               |
    |                         Message Body                          |
    |                                                               |
    |                               +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                               |            Padding            |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

```
     0                   1                   2                   3
     0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    | Message Type  |  Resv   |N|1|0|         Message Size          |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                      Originator Address                       |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |    Message Sequence Number    |                               |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+                               |
    |                                                               |
    |                         Message Body                          |
    |                                                               |
    |                               +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                               |            Padding            |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+


     0                   1                   2                   3
     0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    | Message Type  |  Resv   |0|1|1|         Message Size          |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                                                               |
    |                         Message Body                          |
    |                                                               |
    |                               +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                               |            Padding            |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Appendix A.3. Message Body

```
     0                   1                   2                   3
     0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                                                               |
    |                       Message TLV Block                       |
    |                                               +-+-+-+-+-+-+-+-+
    |                                               |               |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+               |
    |                                                               |
    |                         Address Block                         |
    |                                                               |
    |               +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |               |                                               |
    +-+-+-+-+-+-+-+-+                                               |
    |                       Address TLV Block                       |
    |                                                               |
    |                               +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                               |                               |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+                               |
    |                                                               |
    :                              ...                              :
    |                                                               |
    |                                               +-+-+-+-+-+-+-+-+
    |                                               |               |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+               |
    |                                                               |
    |                         Address Block                         |
    |                                                               |
    |                               +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                               |                               |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+                               |
    |                                                               |
    |                       Address TLV Block                       |
    |                                                               |
    |                               +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                               |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Appendix A.4. Address Block

```
     0                   1                   2                   3
     0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    | Number Addrs  |0| Head Length |             Head              |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |             Mid               |                               |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+                               |
    |                                                               |
    :                              ...                              :
    |                                                               |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |             Mid               |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+


     0                   1                   2                   3
     0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    | Number Addrs  |1| Head Length |             Head              |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |0| Tail Length |             Tail              |      Mid      |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |  Mid (cont)   |                                               |
    +-+-+-+-+-+-+-+-+                                               |
    |                                                               |
    :                              ...                              :
    |                                                               |
    |               +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |               |              Mid              |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+


     0                   1                   2                   3
     0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    | Number Addrs  |1| Head Length |             Head              |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |1| Tail Length |              Mid              |               |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+               |
    |                                                               |
    :                              ...                              :
    |                                                               |
    |               +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |               |              Mid              |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Appendix A.5. TLV Block

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|            Length             |                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+                               |
|                                                               |
|                              TLV                              |
|                                               +-+-+-+-+-+-+-+-+
|                                               |               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+               |
|                                                               |
:                              ...                              :
|                                                               |
|                               +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                               |                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+                               |
|                                                               |
|                              TLV                              |
|                                               +-+-+-+-+-+-+-+-+
|                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Appendix A.6. TLV

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      Type     |Resv |M|0|0|0|0|  Index Start  |  Index Stop   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    Length     |                                               |
+-+-+-+-+-+-+-+-+                                               |
|                             Value                             |
|                                                               |
|               +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|               |
+-+-+-+-+-+-+-+-+


 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      Type     |Resv |0|0|0|0|1|  Index Start  |  Index Stop   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

```
     0                   1                   2                   3
     0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |     Type      |Resv |M|0|0|1|0|  Index Start  |  Index Stop   |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |            Length             |                               |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+                               |
    |                                                               |
    |                             Value                             |
    |                                                               |
    |               +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |               |
    +-+-+-+-+-+-+-+-+
```

```
     0                   1                   2                   3
     0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |     Type      |Resv |M|0|1|0|0|    Length     |               |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+               |
    |                                                               |
    |                             Value                             |
    |                                                               |
    |               +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |               |
    +-+-+-+-+-+-+-+-+
```

```
     0                   1                   2                   3
     0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |     Type      |Resv |0|0|1|0|1|
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

```
     0                   1                   2                   3
     0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |     Type      |Resv |M|0|1|1|0|            Length             |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                                                               |
    |                             Value                             |
    |                                                               |
    |               +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |               |
    +-+-+-+-+-+-+-+-+
```

```
     0                   1                   2                   3
     0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |      Type     |Resv |0|1|0|0|0|  Index Start  |    Length     |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                                                               |
    |                             Value                             |
    |                                                               |
    |               +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |               |
    +-+-+-+-+-+-+-+-+
```

```
     0                   1                   2                   3
     0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |      Type     |Resv |0|1|0|0|1|  Index Start  |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

```
     0                   1                   2                   3
     0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |      Type     |Resv |0|1|0|1|0|  Index Start  |    Length     |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    | Length (cont) |                                               |
    +-+-+-+-+-+-+-+-+                                               |
    |                                                               |
    |                             Value                             |
    |                                                               |
    |               +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |               |
    +-+-+-+-+-+-+-+-+
```

Appendix B. Complete Example

An example packet, using IPv4 addresses (length four octets) is shown. This packet has a header, with a packet sequence number but no packet TLV block, and contains a single unfragmented message. The message has a complete message header, a message TLV block of content length 9 octets containing a single TLV (with the noindex bit of its semantics set and value length 6 octets) and then two address blocks each with a following TLV block. The first address block contains 3 addresses (head length 2 octets, no tail, hence mid length two octets) and is followed by a TLV block of content length 9 octets containing two TLVs. The first of these TLVs has the noindex bit of its semantics set and has a single value of length 2 octets, which applies to all of the addresses in the preceding address block. The second of these TLVs has the novalue bit of its semantics set and hence has no length or value fields (it does have index fields, which indicate those addresses this TLV applies to). The second address block contains 2 addresses (head length 0 octets, hence no head octets, tail length 2 octets, zero-valued tail not included, hence mid length two octets) and is followed by a TLV block of content length 5 octets. This TLV block contains a single TLV of type PREFIX_LENGTH that has the multivalue and noindex bits of its semantics set and a value field length of 2 octets, indicating two values each of one octet length. There are two final padding octets that are not included in the message length of 62 octets.

```
    0                   1                   2                   3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |0 0 0 0 0 0 0 0| Reserved  |0|0|    Packet Sequence Number     |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   | Message Type  |  Resv   |N|0|0|0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 0|
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                      Originator Address                       |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |   Hop Limit   |   Hop Count   |    Message Sequence Number    |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1|   TLV Type    |Resv |0|0|1|0|0|
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |0 0 0 0 0 1 1 0|                     Value                     |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                 Value (cont)                  |0 0 0 0 0 0 1 1|
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |0 0 0 0 0 0 1 0|             Head              |      Mid      |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |  Mid (cont)   |              Mid              |      Mid      |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |  Mid (cont)   |0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1|   TLV Type    |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |Resv |0|0|1|0|0|0 0 0 0 0 0 1 0|             Value             |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |   TLV Type    |Resv |0|0|0|0|1|  Index Start  |  Index Stop   |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |0 0 0 0 0 0 1 0|1 0 0 0 0 0 0 0|1 0 0 0 0 0 1 0|      Mid      |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |  Mid (cont)   |              Mid              |0 0 0 0 0 0 0 0|
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |0 0 0 0 0 1 0 1| PREFIX_LENGTH |Resv |1|0|1|0|0|0 0 0 0 0 0 1 0|
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |    Value0     |    Value1     |0 0 0 0 0 0 0 0|0 0 0 0 0 0 0 0|
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Appendix C. Contributors

This specification is the result of the joint efforts of the following contributors from the OLSRv2 Design Team -- listed alphabetically.

- Cedric Adjih, INRIA, France, <Cedric.Adjih@inria.fr>

- Emmanuel Baccelli, Hitachi Labs Europe, France, <Emmanuel.Baccelli@inria.fr>

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

- Justin W. Dean, NRL, USA<jdean@itd.nrl.navy.mil>

- Christopher Dearlove, BAE Systems, UK, <chris.dearlove@baesystems.com>

- Satoh Hiroki, Hitachi SDL, Japan, <h-satoh@sdl.hitachi.co.jp>

- Philippe Jacquet, INRIA, France, <Philippe.Jacquet@inria.fr>

- Monden Kazuya, Hitachi SDL, Japan, <monden@sdl.hitachi.co.jp>

Appendix D. Acknowledgements

The authors would like to acknowledge the team behind OLSRv1, as specified in RFC 3626, including Anis Laouiti, Pascale Minet, Laurent Viennot (all at INRIA, France), and Amir Qayuum (Center for Advanced Research in Engineering, Pakistan) for their contributions.

The authors would like to gratefully acknowledge the following people for intense technical discussions, early reviews and comments on the specification and its components: Joe Macker (NRL), Alan Cullen (BAE Systems), Ian Chakeres (Boeing), Charlie E. Perkins (Nokia), Andreas Schjonhaug (LIX), Florent Brunneau (LIX), and the entire IETF MANET working group.

Authors' Addresses

Thomas Heide Clausen
LIX, Ecole Polytechnique, France

Phone: +33 6 6058 9349
Email: T.Clausen@computer.org
URI: http://www.lix.polytechnique.fr/Labo/Thomas.Clausen/

Christopher M. Dearlove
BAE Systems Advanced Technology Centre

Phone: +44 1245 242194
Email: chris.dearlove@baesystems.com
URI: http://www.baesystems.com/ocs/sharedservices/atc/

Justin W. Dean
Naval Research Laboratory

Phone: +1 202 767 3397
Email: jdean@itd.nrl.navy.mil
URI: http://pf.itd.nrl.navy.mil/

Cedric Adjih
INRIA Rocquencourt

Phone: +33 1 3963 5215
Email: Cedric.Adjih@inria.fr
URI: http://menetou.inria.fr/~adjih/

Intellectual Property Statement

Disclaimer of Validity

Copyright Statement

Acknowledgment

Funding for the RFC Editor function is currently provided by the Internet Society.