Mobile Ad hoc Networking (MANET)
Internet-Draft
Expires: December 21, 2006

T. Clausen
LIX, Ecole Polytechnique, France
C. Dearlove
BAE Systems Advanced Technology Centre
J. Dean
Naval Research Laboratory
The OLSRv2 Design Team
MANET Working Group

June 19, 2006

MANET Neighborhood Discovery Protocol (NHDP)
draft-ietf-manet-nhdp-00

Status of this Memo

Copyright Notice

Abstract

This document describes a neighborhood discovery protocol (NHDP) for a mobile ad hoc network (MANET). The protocol provides each MANET node with local topology up to two hops distant, describing a node's 1-hop neighbors and symmetric 2-hop neighbors. This is achieved through periodic message exchange. This neighborhood discovery protocol may be used by any MANET protocols which need neighborhood information.

The protocol imposes minimum requirements to the network by not requiring sequenced or reliable transmission of control traffic.

Table of Contents

1. Introduction

This document describes a neighborhood discovery protocol (NHDP) for a mobile ad hoc network (MANET). It was originally developed for the Optimized Link State Routing Protocol version 2 (OLSRv2) [3], based on that used in the Optimized Link State Routing Protocol version 1 [4]. It uses an exchange of HELLO messages in order that each node can determine its neighborhood up to two hops distant. This protocol is used by OLSRv2 to determine a node's 1-hop neighbors for routing, and to allow the selection of MultiPoint Relays (MPRs) for optimized flooding and topology reporting. This specification however only describes the message exchange and information storage required for 1-hop and symmetric 2-hop neighborhood discovery. It may be used by any MANET protocols which need neighborhood information, and which may add an MPR mechanism, or an alternative to it, using appropriate TLVs (type-length-value objects) as specified in [1], using which this protocol's HELLO message format is defined.

This protocol makes no assumptions about the underlying link layer, other than support of local multicast. Link layer information and notifications may be used if available and applicable to qualify the neighborhood information.

1.1. Terminology

The keywords "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC2119 [2].

Additionally, this document uses the following terminology:

node - A MANET router which implements this neighborhood discovery protocol.

MANET interface - A network device participating in a MANET and using this neighborhood discovery protocol. A node may have several MANET interfaces, each interface being assigned one or more IP addresses.

link - A link is a pair of MANET interfaces from two different nodes, where at least one interface is able to hear (i.e. receive traffic from) the other.

symmetric link - A link where both MANET interfaces are able to hear (i.e. receive traffic from) the other.

asymmetric link - A link which is not symmetric.

1-hop neighbor - A node X is a 1-hop neighbor of a node Y if node Y can hear node X (i.e. a link exists from a MANET interface on node X to a MANET interface on node Y).

symmetric 1-hop neighbor - A node X is a symmetric 1-hop neighbor of a node Y if a symmetric link exists from a MANET interface on node X to a MANET interface on node Y.

symmetric 2-hop neighbor - A node X is a symmetric 2-hop neighbor of a node Y if node X is a symmetric 1-hop neighbor of a symmetric 1-hop neighbor of node Y, but is not node Y itself.

1-hop neighborhood - The 1-hop neighborhood of a node X is the set of the 1-hop neighbors of node X.

symmetric 1-hop neighborhood - The symmetric 1-hop neighborhood of a node X is the set of the symmetric 1-hop neighbors of node X.

symmetric 2-hop neighborhood - The symmetric 2-hop neighborhood of a node X is the set of the symmetric 2-hop neighbors of node X. (This may include nodes in the 1-hop neighborhood, or even in the symmetric 1-hop neighborhood, of node X.)

1.2. Applicability Statement

This neighborhood discovery protocol supports nodes which have one or more interfaces which participate in the MANET. It provides each node with local topology information up to two hops away. This information is made available to other protocols through a collection of sets describing the node's 1-hop neighborhood and symmetric 2-hop neighborhood, collectively known as the neighborhood information base.

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 [5].

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 protocol uses the message exchange format specified in [1]. This implies that the HELLO messages specified by this protocol may be extended using the TLV mechanisms described in [1], e.g. to signal

MPR selection as required by OLSRv2 [3]. This also implies that neighborhood discovery protocol messages can be transmitted in packets with messages from other protocols, so long as these protocols also use [1].

2. Protocol Overview and Functioning

This protocol consists of a specification of local signaling, which serves to:

- advertise the presence of a node and its MANET interfaces to adjacent nodes (1-hop neighbors);

- discover links to adjacent nodes;

- perform bidirectionality (symmetry) checks on the discovered links;

- advertise discovered links and whether each is symmetric to its 1-hop neighbors and hence discover symmetric 2-hop neighbors;

- maintain an information base consisting of discovered links and their status, 1-hop neighbors and their MANET interfaces, symmetric 1-hop neighbors and symmetric 2-hop neighbors.

Signaling consists of a single type of periodically transmitted message known as a HELLO message. A HELLO message identifies its originator node and that node's MANET interfaces and addresses. As a node receives HELLO messages and populates its information base, a list of 1-hop neighbors' MANET interface addresses and their link status is included in subsequent HELLO message content. Thus, the periodic transmission allows nodes to continuously track changes in their 1-hop and symmetric 2-hop neighborhoods.

Because each node sends HELLO messages periodically, the protocol can tolerate reasonable message loss without requiring reliable transmission. Such losses can occur frequently in radio networks due to collisions or other transmission problems.

The nominal time interval of a node's periodic HELLO transmission is known as HELLO_INTERVAL and MAY be included in the HELLO message. The HELLO message MUST include a validity time value that indicates the length of time for which the message content is to be considered valid and included in the receiving node's information base. In some uses the validity time may be a multiple of HELLO_INTERVAL to allow for lossy exchange of HELLO messages.

HELLO messages MAY, in addition to periodic transmissions, also be generated as a response to some event (e.g. a change in the advertised neighborhood indicated by a received HELLO message or by a layer 2 notification, if available, indicating a change in a link to a neighbor). However a node MUST respect a minimum interval, HELLO_MIN_INTERVAL, between successive HELLO message transmissions in

order to maintain an upper bound on signaling traffic.

This protocol is designed to work in a completely distributed manner and does not depend on any central entity. This protocol does not require any changes to the format of IP packets, thus any existing IP stack can be used as is.

Each MANET node will form estimates of its 1-hop and symmetric 2-hop neighborhoods as this protocol operates. These estimates can be maintained in an information base comprised of the data sets listed below. These are defined formally in Section 3 and can be summarized as follows:

Link Set - This set records the status of all links from and, if symmetric, to 1-hop neighbors. It also records as lost links which used to be symmetric but have since failed. A node MUST record the Link Set in order to correctly send HELLO messages.

Symmetric Neighbor Set - This set records the addresses of the MANET interfaces of symmetric 1-hop neighbors, or, as lost, those which used to be. Note that if any of these nodes have more than one MANET interface then this set may record addresses that are not in the Link Set. A node MUST record the Symmetric Neighbor Set in order to correctly send HELLO messages.

Neighborhood Address Association Set - This set allows the addresses of the MANET interfaces of each 1-hop neighbor to be associated with each other. It is required for processes such as MPR selection as in [3].

2-Hop Neighbor Set - This set records the addresses of the MANET interfaces of symmetric 2-hop neighbors. It is required for processes such as MPR selection as in [3].

3. Neighborhood Information Base

The neighborhood information base stores information about the 1-hop neighborhood and the symmetric 2-hop neighborhood of a node.

Note that it is possible for a node, X, to be present in both the 1-hop and symmetric 2-hop neighborhoods of another node, Y, concurrently. If the link between node X and node Y breaks, this allows node X to be taken into consideration as a symmetric 2-hop neighbor by node Y immediately, rather than waiting for a HELLO message exchange cycle.

This specification assumes that all addresses have an associated prefix length. The prefix length of an address is, in HELLO messages, indicated using the PREFIX_LENGTH TLV specified in [1]. If no PREFIX_LENGTH TLV is present for a given address, it is assumed that the prefix length for that address is equal to the length of the address. Two addresses are identical if and only if both the addresses and their associated prefix lengths are identical.

Addresses recorded in the neighborhood information base (L_local_iface_addr, L_neighbor_iface_addr, N_local_iface_addr, N_neighbor_iface_addr, N2_local_iface_addr, N2_neighbor_iface_addr, N2_2hop_iface_addr and those listed in NA_neighbor_iface_addr_list) MUST all be recorded with prefix lengths, in order to allow comparison with addresses received in HELLO messages.

3.1. Link Set

A node records a set of "Link Tuples", recording information about its 1-hop neighborhood:

```
  (L_local_iface_addr, L_neighbor_iface_addr, L_SYM_time,
   L_ASYM_time, L_time)
```

each describing a link between a MANET interface of this node and a MANET interface of one of its 1-hop neighbors, where:

L_local_iface_addr is the address of the MANET interface of the local node on which the 1-hop neighbor is or was heard;

L_neighbor_iface_addr is the address of the MANET interface of the 1-hop neighbor;

L_SYM_time is the time until which the link to the 1-hop neighbor is considered symmetric;

L_ASYM_time is the time until which the MANET interface of the 1-hop neighbor is considered heard;

L_time specifies when this Link Tuple expires and MUST be removed.

The status of the link, denoted L_STATUS, can be derived based on the fields L_SYM_time and L_ASYM_time as defined in Table 1.

L_SYM_time	L_ASYM_time	L_STATUS
Expired	Expired	LOST
Not Expired	Expired	SYMMETRIC
Not Expired	Not Expired	SYMMETRIC
Expired	Not Expired	HEARD

Table 1

In a node, the set of Link Tuples is denoted the "Link Set".

3.2. Symmetric Neighbor Set

A node records a set of "Symmetric Neighbor Tuples", recording information about its symmetric 1-hop neighborhood:

(N_local_iface_addr, N_neighbor_iface_addr, N_SYM_time, N_time)

each describing an address of a MANET interface of this node and an address of a MANET interface of one of its symmetric 1-hop neighbors, where:

N_local_iface_addr is the address of the MANET interface of the local node to which the 1-hop neighbor has or had a symmetric link;

N_neighbor_iface_addr is an address of a MANET interface of a 1-hop neighbor which is or was in this node's symmetric 1-hop neighborhood;

N_SYM_time is the time until which the 1-hop neighbor is considered to be in this node's symmetric 1-hop neighborhood;

N_time specifies when this Symmetric Neighborhood Tuple expires and MUST be removed.

The status of the 1-hop neighbor, denoted N_STATUS, can be derived based on the field L_SYM_time as defined in Table 2.

N_SYM_time	N_STATUS
Expired	LOST
Not Expired	SYMMETRIC

Table 2

In a node, the set of Symmetric Neighbor Tuples is denoted the "Symmetric Neighbor Set".

3.3. Neighborhood Address Association Set

A node records a set of "Neighborhood Address Association Tuples", recording information about the MANET interface configuration of its 1-hop neighbors:

(NA_neighbor_iface_addr_list, NA_time)

NA_neighbor_iface_addr_list is a list of interface addresses of a single 1-hop neighbor;

NA_time specifies when this Neighborhood Address Association Tuple expires and MUST be removed.

In a node, the set of Neighborhood Address Association Tuples is denoted the "Neighborhood Address Association Set".

3.4. 2-Hop Neighbor Set

A node records a set of "2-Hop Neighbor Tuples", recording information about a its symmetric 2-hop neighborhood:

(N2_local_iface_addr, N2_neighbor_iface_addr, N2_2hop_iface_addr, N2_time)

each describing a symmetric link between an address of a MANET interface of one of this node's symmetric 1-hop neighbors and an address of a MANET interface of a node in this node's symmetric 2-hop neighborhood.

N2_local_iface_addr is the address of the local MANET interface over which the information defining this 2-Hop Neighbor Tuple was received;

N2_neighbor_iface_addr is the address of the MANET interface address of a symmetric 1-hop neighbor;

N2_2hop_iface_addr is the address of a MANET interface of a symmetric 2-hop neighbor which has a symmetric link (not necessarily using this address) to the node with MANET interface address N2_neighbor_iface_addr;

N2_time specifies the time at which this 2-Hop Neighbor Tuple expires and MUST be removed.

In a node, the set of 2-Hop Neighbor Tuples is denoted the "2-Hop Neighbor Set".

4. Packets and Messages

The packet and message format used by this neighborhood discovery protocol is defined in [1], which is used with the following considerations:

- this protocol specifies one message type, the HELLO message, in Section 4.1;
- HELLO message header options may be used as specified by the protocol which uses this neighborhood discovery protocol;
- HELLO messages are transmitted only one hop, i.e. MUST NOT be forwarded;
- multi-message packets may be created using other messages as specified by the protocol which uses this neighborhood discovery protocol;
- packet header options may be used as specified by the protocol which uses this neighborhood discovery protocol.

4.1. HELLO Messages

A HELLO message MUST contain:

- a message TLV with Type == VALIDITY_TIME and Value == H_HOLD_TIME, as specified in Section 4.1.2.1
- an address block known as the Local Interface Block, as specified in Section 4.1.1; other addresses MUST NOT be added to this address block.

A HELLO message MAY contain:

- a message TLV with Type == INTERVAL_TIME and Value == HELLO_INTERVAL, as specified in Section 4.1.2.2;
- one or more address blocks with associated address block TLVs as specified in Section 4.1.3; these address blocks contain current or former 1-hop neighbors' MANET interface addresses with associated TLVs. Other addresses (i.e. addresses of non-neighbor nodes) MAY be added to these address blocks, however if they are then these addresses MUST NOT have associated TLVs from the address block TLVs specified in Section 4.1.3.

This protocol specifies two message TLVs and three address block TLVs used in HELLO messages in Section 4.1.2 and Section 4.1.3; other TLVs

MAY be included as specified by the protocol which uses this neighborhood discovery protocol.

4.1.1. Local Interface Block

The first address block, plus following TLV block, in a HELLO message is known as the Local Interface Block. The Local Interface Block is not distinguished in any way other than by being the first address block in the message.

The first address of the Local Interface Block MUST contain the used address of the interface over which the HELLO message is transmitted. If that interface has an associated prefix different from the length of the address, a PREFIX_LENGTH TLV MUST be associated with this address. This first address, with associated prefix length, of the Local Interface Block is henceforth denoted the "Source Address".

The Local Interface Block MUST contain all of the addresses of all of the MANET interfaces of the originating node, using the standard <address-block> syntax specified in [1]. Those addresses, if any, which correspond to MANET interfaces other than that on which the HELLO message is transmitted MUST have a corresponding OTHER_IF TLV as specified in Section 4.1.3, other addresses MUST NOT use this TLV.

Note that a Local Interface Block MAY include more than one address for each MANET interface, and hence a HELLO message MAY contain more than one address without an OTHER_IF TLV. A Local Interface Block MUST NOT contain any addresses which are not of MANET interfaces of the originating node.

4.1.2. Message TLVs

This section specifies two message TLVs: VALIDITY_TIME and INTERVAL_TIME.

4.1.2.1. VALIDITY_TIME TLV

All HELLO messages MUST include a VALIDITY_TIME TLV, specifying for how long a node may, upon receiving a message, consider the message content to be valid. The VALIDITY_TIME TLV, described in this document, contains a single value since HELLO messages are transmitted only one hop. Note that [1] specifies an extended version of this VALIDITY_TIME TLV, which is compatible with the format of the VALIDITY_TIME TLV in this specification.

The VALIDITY_TIME message TLV specification is given in Table 3.

Name	Type	Length	Value
VALIDITY_TIME	TBD	8 bits	<t_default>

Table 3

where <t_default> is the period for which the information is valid, represented as specified in Section 4.1.2.3.

4.1.2.2. INTERVAL_TIME TLV

HELLO messages MAY include an INTERVAL_TIME message TLV, specifying the interval at which HELLO messages are being generated by the originator node. Note that HELLO messages which are not part of a regular schedule SHALL be ignored in defining the interval. If, for whatever reason, HELLO messages are sent in an irregular pattern, then this SHALL be the longest interval in that pattern.

The INTERVAL_TIME message TLV specification is given in Table 4.

Name	Type	Length	Value
INTERVAL_TIME	TBD	8 bits	<time>

Table 4

where <time> is the scheduled maximum time until the next transmission of a HELLO message by the originator node on the same interface, represented as specified in Section 4.1.2.3.

4.1.2.3. Representing Time

Section 4.1.2.1 and Section 4.1.2.2 specify TLVs where time is represented as a single octet. This is defined by the lowest four bits of the octet representing the mantissa (a) and the highest four bits of the octet representing the exponent (b) of the time as a multiple of a fixed constant C, yielding that:

- o time = C * (1 + a/16) * 2^b

where a is the integer represented by the four lowest bits of the time field and b the integer represented by the four highest bits of the time field. All nodes in the network MUST use the same value of C, which will be specified in seconds, hence so will be all times

(see Section 7). Note that ascending values of the octet represent ascending values of time, times may thus be compared by comparison of octets.

An algorithm for computing the representation of time t is the following:

1. find the largest integer b such that t/C >= 2^b;

2. set a = 16 * (t / (C * 2^b) - 1), rounded up to the nearest integer;

3. if a == 16 then set b = b + 1 and set a = 0;

4. if a and b are in the range 0 and 15 then t can be represented by an octet holding the value 16*b + a, otherwise it can not.

As examples, the values of 2 seconds and 6 seconds are represented by (a=0, b=5) and (a=8, b=6), respectively, i.e., by the octets 80 and 104 (hexadecimal 50 and 68).

4.1.3. Address Block TLVs

The three address block TLVs used in HELLO messages are specified in Table 5.

Name	Type	Length	Value
OTHER_IF	TBD	0 bits	Not Applicable
LINK_STATUS	TBD	8 bits	One of LOST, SYMMETRIC or HEARD.
OTHER_NEIGHB	TBD	8 bits	One of LOST or SYMMETRIC

Table 5

5. HELLO Message Generation

HELLO messages MUST be generated and transmitted independently on each MANET interface. The maximum interval between HELLO transmissions on the same MANET interface MUST NOT exceed HELLO_MIN_INTERVAL. Two successive HELLO message transmissions on the same MANET interface MUST be separated by at least HELLO_MIN_INTERVAL.

Each HELLO message MUST include a Local Interface Block as specified in Section 4.1.1 as its first address block.

On its MANET interface with address Sending Address, a node MUST report appropriate addresses with associated TLVs from the Link Set and Symmetric Neighbor Set. These addresses, with their associated TLVs, MAY be reported in any HELLO messages transmitted on that MANET interface. All such addresses, with their associated TLVs, MUST be reported in at least one HELLO message transmitted on that MANET interface within every interval of length REFRESH_INTERVAL. These addresses MUST NOT be included in the Local Interface Block, they MAY be included in any other address block.

The addresses, with their associated TLVs, which MUST be included in HELLO messages over the local MANET interface with address Sending Address, are computed thus:

1. For each Link Tuple with L_local_iface_addr == Sending Address, include:

   * L_neighbor_iface_addr with an associated TLV with:

     + Type = LINK_STATUS; AND

     + Value = L_STATUS.

2. For each address which appears as an N_neighbor_iface_addr in one or more Symmetric Neighbor Tuples:

   1. if this address has already been included with an associated TLV with Type == LINK_STATUS and Value == SYMMETRIC, do not add an associated TLV with Type == OTHER_NEIGHB;

   2. otherwise if, for one or more of these Symmetric Neighbor Tuples, N_STATUS == SYMMETRIC, then include this address with associated TLV with:

      + Type = OTHER_NEIGHB; AND

- Value = SYMMETRIC;

3. otherwise if, for all of these Symmetric Neighbor Tuples, N_STATUS == LOST, and this address has not already been included with an associated TLV with Type == LINK_STATUS and Value == LOST, then include this address with associated TLV with:

   - Type = OTHER_NEIGHB; AND

   - Value = LOST.

If an address is specified as included with more than one associated TLV, then these TLVs MAY be independently included or excluded from HELLO messages as long as all are included with any interval of length REFRESH_INTERVAL. TLVs applying to the same address MAY be applied to the same or different copies of the address in a given HELLO message.

5.1. HELLO Message: Transmission

Messages are transmitted in the packet/message format specified by [1] using the ALL-MANET-NEIGHBORS multicast address as destination IP address and with the HELLO message Hop Limit = 1.

6. HELLO Message Processing

On receiving a HELLO message, a node will update its neighborhood information base according to the specification given in this section.

For the purpose of this section, note the following definitions:

- o the "validity time" of a message is calculated from the VALIDITY_TIME TLV of the message as specified in Section 4.1.2.1;

- o the "Source Address" is the first address, including prefix length, of the Local Interface Block in the HELLO message;

- o the "Receiving Address" is the address, including prefix length, of the MANET interface on which the HELLO message was received;

- o the word EXPIRED indicates that a timer is set to a value clearly preceding the current time (e.g. current time - 1).

6.1. Populating the Link Set

On receiving a HELLO message, a node SHOULD update its Link Set:

1. If there is no Link Tuple with:

    * L_local_iface_addr == Receiving Address; AND

    * L_neighbor_iface_addr == Source Address,

    then create a new Link Tuple with

    * L_local_iface_addr = Receiving Address;

    * L_neighbor_iface_addr = Source Address;

    * L_SYM_time = EXPIRED;

    * L_time = current time + validity time.

2. This Link Tuple (existing or new) is then modified as follows:

    1. If the node finds the Receiving Address in one of the address blocks included in the HELLO message, other than the Local Interface Block, then the Link Tuple is modified as follows:

        1. If the Receiving Address in that address block is associated with a TLV with Type == LINK_STATUS and Value

```
              == LOST then:

              1.  if L_STATUS == SYMMETRIC:

                  o  L_time = current time + max(validity time,
                     L_HOLD_TIME),

                  o  L_SYM_time = EXPIRED.

          2.  Otherwise if the Receiving Address in that address block
              is associated with a TLV with Type == LINK_STATUS and
              (Value == HEARD or Value == SYMMETRIC) then:

              -  L_SYM_time = current time + validity time;

              -  L_time = L_SYM_time + L_HOLD_TIME.

      2.  L_ASYM_time = current time + validity time;

      3.  L_time = max(L_time, L_ASYM_time).
```

6.2. Populating the Symmetric Neighbor Set

On receiving a HELLO message, a node SHOULD update its Symmetric Neighbor Set:

```
   1.  If the Receiving Address is in an address block of the HELLO
       message, other than the Local Interface Block, with an associated
       TLV with Type == LINK_STATUS and (Value == HEARD or Value ==
       SYMMETRIC) then:

       1.  For each address (henceforth neighbor address) in the HELLO
           message's Local Interface Block:

           1.  If there is a Symmetric Neighbor Tuple with:

               -  N_local_iface_addr == Receiving Address; AND

               -  N_neighbor_iface_addr == neighbor address,

               then update this Symmetric Neighbor Tuple to have:

               -  N_SYM_time = current time + validity time;

               -  N_time = N_SYM_time + N_HOLD_TIME.
```

    2. Otherwise create a new Symmetric Neighbor Tuple with:

        - N_local_iface_addr = Receiving Address;

        - N_neighbor_iface_addr = neighbor address;

        - N_SYM_time = current time + validity time;

        - N_time = N_SYM_time + N_HOLD_TIME.

2. Otherwise if the Receiving Address is in an address block of the HELLO message, other than the Local Interface Block, with an associated TLV with Type == LINK_STATUS and Value == LOST, then:

    1. For each address (henceforth neighbor address) in the HELLO message Local Interface Block, if there exists a Symmetric Neighbor Tuple with:

        + N_local_iface_addr == Receiving Address; AND

        + N_neighbor_iface_addr == neighbor address,

        update this Symmetric Neighbor Tuple to have:

        + N_SYM_time = EXPIRED;

        + N_time = min(N_time, current time + N_HOLD_TIME).

6.3. Populating the Neighborhood Address Association Set

On receiving a HELLO message, the node SHOULD update its Neighborhood Address Association Set:

1. Remove all Neighborhood Address Association Tuples where:

    * NA_neighbor_iface_addr_list contains at least one address which is contained in the Local Interface Block of the received HELLO message,

    and create a new Neighborhood Address Association Tuple with:

    * NA_neighbor_iface_addr_list = list of all addresses contained in the Local Interface Block of the received HELLO message;

    * NA_time = current time + validity time.

6.4. Populating the 2-Hop Neighbor Set

On receiving a HELLO message the node SHOULD update its 2-Hop Neighbor Set:

```
1.  If there exists a Link Tuple with L_local_iface_addr == Source
    Address and L_STATUS == SYMMETRIC then:

    1.  For each address (henceforth 2-hop neighbor address) in an
        address block of the HELLO message, other than the Local
        Interface Block, which is not an interface address of the
        receiving node:

        1.  If the 2-hop neighbor address has an associated TLV with:

            -  Type == LINK_STATUS and Value == SYMMETRIC; OR

            -  Type == OTHER_NEIGHB and Value == SYMMETRIC,

            then, if there is no 2-Hop Neighbor Tuple with:

            -  N2_local_iface_addr == Receiving Address;

            -  N2_neighbor_iface_addr == Source Address;

            -  N2_2hop_iface_addr == 2-hop neighbor address;

            create a 2-Hop Neighbor Tuple with:

            -  N2_local_iface_addr = Receiving Address; AND

            -  N2_neighbor_iface_addr = Source Address; AND

            -  N2_2hop_iface_addr = 2-hop neighbor address.

            This 2-Hop Neighbor Tuple (existing or new) is then
            modified as follows:

            -  N2_time = current time + validity time.

        2.  Otherwise if the 2-hop neighbor address has a TLV with:

            -  Type == LINK_STATUS and (Value == LOST or Value ==
               HEARD); OR

            -  Type == OTHER_NEIGHB and Value == LOST;

            then remove all 2-Hop Neighbor Tuples with:
```

- N2_local_iface_addr == Receiving Address; AND
- N2_neighbor_iface_addr == Source Address; AND
- N2_2hop_iface_addr == 2-hop neighbor address.

6.5. Neighborhood Changes

If the L_SYM_time field of a Link Tuple expires (either due to timing out, or as a result of processing a TLV with Type == LINK_STATUS and Value == LOST) then all 2-Hop Neighbor Tuples with:

- N2_local_iface_addr == L_local_iface_addr from the Link Tuple, AND;
- N2_neighbor_iface_addr == L_neighbor_iface_addr from the Link Tuple,

MUST be deleted.

In this, or any other case of neighborhood change, a node MAY send a HELLO message reporting updated information. If a node does send such a HELLO message the node MUST ensure that any two successive HELLO messages are separated by at least HELLO_MIN_INTERVAL.

7. Proposed Values for Constants

This section list the values for the constants used in the description of the protocol.

7.1. Message Intervals

- HELLO_INTERVAL = 2 seconds

- REFRESH_INTERVAL = HELLO_INTERVAL

- HELLO_MIN_INTERVAL = HELLO_INTERVAL/4

7.2. Holding Times

- H_HOLD_TIME = 3 x REFRESH_INTERVAL

- L_HOLD_TIME = H_HOLD_TIME

- N_HOLD_TIME = H_HOLD_TIME

7.3. Time

- C = 0.0625 seconds (1/16 second)

In order to achieve interoperability, C MUST be the same on all nodes.

8. IANA Considerations

8.1. Multicast Addresses

A well-known multicast address, ALL-MANET-NEIGHBORS, must be registered and defined for both IPv6 and IPv4. The addressing scope is link-local, i.e. this address is similar to the all nodes/routers multicast address of IPv6 in that it targets all MANET nodes adjacent to the originator of an IP datagram.

8.2. Message Types

This specification defines one message type, which must be allocated from the "Assigned Message Types" repository of [1]

Mnemonic	Value	Description
HELLO	TBD	Local Signaling

Table 6

8.3. TLV Types

This specification defines two Message TLV types, which must be allocated from the "Assigned Message TLV Types" repository of [1]

Mnemonic	Value	Description
VALIDITY_TIME	TBD	The time (in seconds) from receipt of the message during which the information contained in the message is to be considered valid
INTERVAL_TIME	TBD	The maximum time (in seconds) between two successive transmissions of messages of the appropriate type

Table 7

This specification defines three Address Block TLV types, which must be allocated from the "Assigned Address Block TLV Types" repository of [1]

Mnemonic	Value	Description
OTHER_IF	TBD	Specifies that the address, in the Local Interface Block of the message, is an address associated with a MANET interface other than the one on which the message is transmitted
LINK_STATUS	TBD	Specifies a given link's status (LOST, SYMMETRIC or HEARD)
OTHER_NEIGHB	TBD	Specifies that the address is, or was, of a MANET interface of a symmetric 1-hop neighbor of the node transmitting the HELLO message, but does not have a matching or better LINK_STATUS TLV

Table 8

8.4. LINK_STATUS and OTHER_NEIGHB Values

The values which the LINK_STATUS TLV can use are the following:

- LOST = 0

- SYMMETRIC = 1

- HEARD = 2

The values which the OTHER_NEIGHB TLV can use are the following:

- LOST = 0

- SYMMETRIC = 1

9. References

9.1. Normative References

[1] Clausen, T., Dean, J., Dearlove, C., and C. Adjih, "Generalized MANET Packet/Message Format", Work In Progress draft-ietf-manet-packetbb-01.txt, June 2006.

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

9.2. Informative References

[3] Clausen, T., Dearlove, C., and P. Jacquet, "The Optimized Link State Routing Protocol version 2", Work In Progress draft-ietf-manet-olsrv2-02.txt, June 2006.

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

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

Appendix A. Heuristics for Generating HELLO Messages

The algorithm for generating HELLO messages in Section 5 specifies which addresses MAY be included in the address blocks after the Local Interface Block, and with which associated TLVs. These TLVs may have Type == LINK_STATUS or Type == OTHER_NEIGHB, or both. TLVs with Type == LINK_STATUS may have three possible values (Value == HEARD, Value == SYMMETRIC or Value == LOST), and TLVs of TYPE == OTHER_NEIGHB may have two possible values (Value == SYMMETRIC or Value == LOST). When both TLVs are associated with the same address only certain combinations of these TLV values are necessary, and are the only combinations generated by the algorithm in Section 5. These combinations are indicated in Table 9.

Cells labeled with "Yes" indicate the possible combinations which are generated by the algorithm in Section 5. Cells labeled with "No" indicate combinations not generated by the algorithm in Section 5, but which are correctly parsed and interpreted by the algorithm in Section 6.

	Type == OTHER_NEIGHB (absent)	Type == OTHER_NEIGHB, Value == SYMMETRIC	Type == OTHER_NEIGHB, Value == LOST
Type == LINK_STATUS (absent)	No	Yes	Yes
Type == LINK_STATUS, Value == HEARD	Yes	Yes	Yes
Type == LINK_STATUS, Value == SYMMETRIC	Yes	No	No
Type == LINK_STATUS, Value == LOST	Yes	Yes	No

Table 9

In creating the HELLO message there are three stages:

1. collect the addresses into groups, each of which will form an address block (this heuristic assumes that each address will be included only once in a HELLO message, and that all TLVs associated with it are included);

2. order the addresses in each group for most efficient TLV association;

3. add the TLVs in the most efficient manner, whether single or multiple value.

There is no straightforward way to perform these steps to create the most optimal (smallest) HELLO message. Instead the following heuristics may be considered:

1. The easiest approach to grouping addresses is to put them all in a single address block. Separate address blocks are appropriate when addresses have significantly different initial (head) bit sequences, and the address compression in the address block construction can be more efficient when addresses with different initial sequences can be compressed separately, gaining more than the overhead of multiple address blocks. Separate address blocks have a lower overhead when either they use different TLVs, or when they use multivalue TLVs. The simplest heuristic is to use a single address block, unless addresses may be divided into one or more subnets, especially if these are associated with different MANET interfaces and hence each uses either LINK_STATUS or OTHER_NEIGHB TLVs, but not both.

2. Grouping addresses that use a single TLV is straightforward, so that each TLV type and value may be applied to a continuous sequence of addresses. This can be extended to cover the case where addresses have more than one TLV. An example of how to order all TLV combinations so that each TLV type and value is applied to a continuous sequence of addresses is given. (This order is not unique.)

   * Type == LINK_STATUS, Value == LOST.

   * Type == LINK_STATUS, Value == LOST and Type == OTHER_NEIGHB, Value == SYMMETRIC.

   * Type == OTHER_NEIGHB, Value == SYMMETRIC.

   * Type == LINK_STATUS, Value == HEARD and Type == OTHER_NEIGHB, Value == SYMMETRIC.

- Type == LINK_STATUS, Value == HEARD.

- Type == LINK_STATUS, Value == HEARD and Type == OTHER_NEIGHB, Value == LOST.

- Type == OTHER_NEIGHB, Value == LOST.

- Type == LINK_STATUS, Value == SYMMETRIC.

This order is not appropriate when multiple value TLVs are to be used, then it is more important to group all TLVs of the same type together, even when having different values. A possible ordering is

- Type == LINK_STATUS, Value == HEARD.

- Type == LINK_STATUS, Value == SYMMETRIC.

- Type == LINK_STATUS, Value == LOST.

- Type == LINK_STATUS, Value == HEARD and Type == OTHER_NEIGHB, Value == SYMMETRIC.

- Type == LINK_STATUS, Value == HEARD and Type == OTHER_NEIGHB, Value == LOST.

- Type == LINK_STATUS, Value == LOST and Type == OTHER_NEIGHB, Value == SYMMETRIC.

- Type == OTHER_NEIGHB, Value == SYMMETRIC.

- Type == OTHER_NEIGHB, Value == LOST.

Where one TLV type uses single values and the other multiple values, appropriate orderings can be devised.

3. When there are many addresses in an address block, the most efficient way to add TLVs is as up to five single value TLVs, each with a single octet value field. When there are few addresses in an address block, the most efficient way to add TLVs is as up to two multiple value TLVs, with one octet of value per address each. It may be appropriate to use one approach for each TLV type. It is relatively straightforward to estimate the cost of each approach (adding TLV type, semantics, length and index overheads per TLV, and either one octet per value or per address as appropriate) and to select the probably lower cost approach. Alternatively a single decision based on the expected number of 1-hop neighbor addresses may be made.

Appendix B. HELLO Message Example

A simple example HELLO message, sent by an originator node with a single MANET interface, is as follows. The message uses IPv4 (four octet) addresses without prefix TLVs, i.e. with all addresses having maximum length prefixes. The message is sent with a full message header (message semantics octet is 0) with a hop limit of 1 and a hop count of 0. The overall message length is 48 octets (it does not need padding).

The message has a message TLV block with content length 8 octets containing two message TLVs, of types VALIDITY_TIME and INTERVAL_TIME. Each uses a TLV with semantics value 4, indicating no start and stop indexes are included, and each has a value length of 1 octet. The values included (0x68 and 0x50) represent the default values of 6 seconds and 2 seconds, respectively.

The first address block contains 1 local interface address, with head length 4 and no address tail or mid parts. This address block has no TLVs (TLV block content length 0 octets).

The second, and last, address block contains 4 neighbor interface addresses, with head length 3 octets, no address tail part and each address mid part having length one octet. The following TLV block (content length 7 octets) includes one TLV which reports the link status of all neighbors in a single multivalue TLV: the first two addresses are HEARD, the third address is SYMMETRIC and the fourth address is LOST. The TLV semantics value of 12 indicates, in addition to that this is a multivalue TLV, that no start and stop indexes are included, since values for all addresses are included. The TLV value length of 4 octets indicates one octet per value per address.

```
      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
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |     HELLO     |0 0 0 0 0 0 0 0|0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0|
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |                      Originator Address                       |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |0 0 0 0 0 0 0 1|0 0 0 0 0 0 0 0|    Message Sequence Number    |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0| VALIDITY_TIME |0 0 0 0 0 1 0 0|
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |0 0 0 0 0 0 0 1|0 1 1 0 1 0 0 0| INTERVAL_TIME |0 0 0 0 0 1 0 0|
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |0 0 0 0 0 0 0 1|0 1 0 1 0 0 0 0|0 0 0 0 0 0 0 1|0 0 0 0 0 1 0 0|
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |                             Head                              |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0|0 0 0 0 0 1 0 0|0 0 0 0 0 0 1 1|
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |                     Head                      |      Mid      |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |      Mid      |      Mid      |      Mid      |0 0 0 0 0 0 0 0|
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |0 0 0 0 0 1 1 1|  LINK_STATUS  |0 0 0 0 1 1 0 0|0 0 0 0 0 1 0 0|
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |     HEARD     |     HEARD     |   SYMMETRIC   |     LOST      |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Appendix C. Security Considerations

The objective of this protocol is to allow each node in the network to acquire information describing its 1-hop and symmetric 2-hop neighborhoods. This is acquired through periodic message exchange between neighboring nodes, and the information is made available through a collection of sets, describing the node's 1-hop neighborhood and symmetric 2-hop neighborhood.

Under normal circumstances, the information recorded in these sets is correct - i.e. corresponds to the actual network topology, apart from any changes which have not (yet) been tracked by the HELLO message exchanges. If some node for some reason, malice or malfunction, inject invalid HELLO messages, incorrect information may be recorded in the sets maintained.

A correctly formed, but still invalid, HELLO message may take any of the following forms:

1. The Local Interface Block of the HELLO message may contain addresses which do not correspond to addresses of MANET interfaces of the local node which transmits the HELLO message;

2. The Local Interface Block of the HELLO message may omit addresses of MANET interfaces of the local node which transmits the HELLO message;

3. The Local Interface Block may contain OTHER_IF TLVs, indicating incorrectly that an address is associated with a MANET interface other than the one over which the HELLO message is being transmitted;

4. The Local Interface Block may omit OTHER_IF TLVs, thereby indicating incorrect addresses associated with the MANET interface over which the HELLO message is being transmitted;

5. A present or absent address in an address block, other than in the Local Interface Block, does not in and by itself cause a problem. It is the presence, absence or incorrectness of associated LINK_STATUS and OTHER_NEIGHB TLVs that cause problems;

6. A present LINK_STATUS TLV may, incorrectly, identify an address as being of a node which is or was in the sending node's 1-hop neighborhood;

7. A consistently absent LINK_STATUS TLV may, incorrectly, fail to identify an address as being of a node which is or was in the

sending node's 1-hop neighborhood;

8. A present OTHER_NEIGHB TLV may, incorrectly, identify an address as being of a node which is or was in the sending node's symmetric 1-hop neighborhood;

9. A consistently absent OTHER_NEIGHB TLV may, incorrectly, fail to identify an address as being of a node which is or was in the sending node's symmetric 1-hop neighborhood;

10. The value of a LINK_STATUS or OTHER_NEIGHB TLV may incorrectly indicate the status (LOST, SYMMETRIC, HEARD) of a 1-hop neighbor.

Appendix D. Flow and Congestion Control

This document specifies one message type, HELLO messages. The size of each complete HELLO message is proportional to the size of the transmitting node's 1-hop neighborhood (this information may be sent distributed across multiple interfaces). HELLO messages MUST NOT be forwarded.

A node MUST report its 1-hop neighborhood in HELLO messages on each of its MANET interfaces at least each REFRESH_INTERVAL. Thus, this puts a lower bound on the control traffic, which each node employing this neighborhood discovery protocol in the network generates.

A node MUST ensure that two successive HELLO messages sent on the same MANET interface are separated by at least HELLO_MIN_INTERVAL. Thus, this puts an upper bound on the control traffic, which each node employing this neighborhood discovery protocol in the network generates.

In order for the protocol to function, each node in the network MUST employ the HELLO signaling as described in this specification.

Appendix E. Contributors

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

- Brian Adamson, NRL, USA, <adamson@itd.nrl.navy.mil>

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

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

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

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

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

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

Appendix F. Acknowledgements

The authors would like to acknowledge the team behind OLSRv1, specified in RFC3626, 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), Song-Yean Cho (Samsung Software Center) 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/

The OLSRv2 Design Team
MANET Working Group

Intellectual Property Statement

Disclaimer of Validity

This document and the information contained herein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Copyright Statement

Acknowledgment

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