Internet Engineering Task Force (IETF)
Request for Comments: 6130
Category: Standards Track
ISSN: 2070-1721

T. Clausen
LIX, Ecole Polytechnique
C. Dearlove
BAE Systems ATC
J. Dean
Naval Research Laboratory

April 2011

Mobile Ad Hoc Network (MANET) Neighborhood Discovery Protocol (NHDP)

Abstract

This document describes a 1-hop and symmetric 2-hop neighborhood discovery protocol (NHDP) for mobile ad hoc networks (MANETs).

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

This document describes a neighborhood discovery protocol (NHDP) for a mobile ad hoc network (MANET) [RFC2501]. This protocol uses a local exchange of HELLO messages so that each router can determine the presence of, and connectivity to, its 1-hop and symmetric 2-hop neighbors. Messages are defined and sent in packets according to the specification [RFC5444].

1-hop neighborhood information is recorded for use by MANET routing protocols to determine direct (1-hop) connectivity to neighboring routers. 2-hop symmetric neighborhood information is recorded so as to enable MANET routing protocols to employ flooding reduction techniques, e.g., to select reduced relay sets for efficient network-wide traffic dissemination.

1-hop and symmetric 2-hop neighborhood information is recorded in the form of Information Bases. These are available for use by other protocols, such as MANET routing protocols, that require information regarding the local network connectivity. This protocol is designed to maintain the information in these Information Bases even in the presence of a dynamic network topology and wireless communication channel characteristics.

This protocol makes no assumptions about the underlying link layer, other than support of local broadcast or multicast for communication to 1-hop neighbor routers. Link-layer information may be used if available and applicable.

This protocol is based on the neighborhood discovery process contained in the Optimized Link State Routing (OLSR) Protocol [RFC3626].

1.1. Motivation

MANETs differ from more traditional wired and infrastructure-based wireless networks due to their envisioned applicability over more challenging communication channels (e.g., wireless, lossy, broadcast channels with moderate and shared bandwidth, hidden and exposed terminals, and interference from other network devices and the environment) and in more challenging topological conditions (e.g., rapid and unpredictable mobility, dynamic and non-predetermined network membership).

Due to the properties of wireless transmissions, communication between two neighboring routers may not be bi-directional; even if router A is able to receive packets from router B, the converse is not guaranteed to be true. Furthermore, because of the localized nature of wireless broadcast communication, neighboring routers within the same communications channel may have different sets of neighbors. That is, when using the same communication channel, even if router A is able to exchange packets with router B, and router B is able to exchange packets with router C, this does not guarantee that router A and router C can exchange packets directly.

Each router in a MANET may use more than one communication channel. In particular, between the same pair of routers, more than one distinct communication channel may exist, each with different properties. This may, for example, be the case where MANET routers are equipped with multiple distinct wireless interfaces, operating at different frequencies.

For use by MANET routing protocols, as well as for establishing a router’s neighbors, a router may also need to determine whether each communication channel with that neighbor is bi-directional.

The set of neighbor routers of a given MANET router may be continuously changing, often due to router mobility or a changing physical environment in which the MANET is located. There is typically no information from lower layers that would enable an IP routing protocol to detect and, as appropriate, react to such changes. Such changes can often take place on a short timescale, such as of the order of seconds, requiring MANET routing protocols to act rapidly to ensure suitable convergence properties.

MANET routing protocols, for example [RFC3626] and [RFC5449], often employ relay set reductions in order to conserve network capacity when maintaining network-wide topological information, with calculation of these reduced relay sets employing up to two hop information.

The neighborhood discovery protocol specified in this document provides continued tracking of neighborhood changes, link bi-directionality, and local topological information up to two hops. Combined, this allows a MANET routing protocol access to information describing link establishment/disappearance and provides the necessary topological information for reduced relay set selection and other purposes.

2. Terminology

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

All terms introduced in [RFC5444], including "packet", "message", "Address Block", "TLV Block", "TLV", "address", "address prefix", and "address object" are to be interpreted as described therein.

Additionally, this document uses the following terminology:

Network Address:
: An address plus an associated prefix length. This may be an address with an associated maximum prefix length or an address prefix including a prefix length. A network address thus represents a range of addresses.

Router:
: A MANET router that implements this neighborhood discovery protocol.

Interface:
: A router’s attachment to a communications medium. An interface is assigned one or more addresses.

MANET interface:
: An interface participating in a MANET and using this neighborhood discovery protocol. A router may have several MANET interfaces.

Heard:
A MANET interface of router X is considered heard on a MANET interface of a router Y if the latter can receive control messages, according to this specification, from the former.

Link:
A link between two MANET interfaces exists if either can be heard by the other.

Symmetric link:
A symmetric link between two MANET interfaces exists if both can be heard by the other.

1-hop neighbor:
A router X is a 1-hop neighbor of a router Y if a MANET interface of router X is heard by a MANET interface of router Y.

Symmetric 1-hop neighbor:
A router X is a symmetric 1-hop neighbor of a router Y if a symmetric link exists between a MANET interface on router X and a MANET interface on router Y.

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

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

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

Symmetric 1-hop neighborhood:
The symmetric 1-hop neighborhood of a router X is the set of the symmetric 1-hop neighbors of router X.

2-hop neighborhood:
The 2-hop neighborhood of a router X is the set of the 2-hop neighbors of router X. (This may include routers in the 1-hop neighborhood of router X.)

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

Constant:
A numerical value that MUST be the same for all MANET interfaces of all routers in the MANET, at all times.

Interface parameter:
A boolean or numerical value, specified separately for each MANET interface of each router. A router MAY change interface parameter values at any time, subject to some constraints.

Router parameter:
A boolean or numerical value, specified for each router, and not specific to an interface. A router MAY change router parameter values at any time, subject to some constraints.

Information Base:
A collection of information maintained by this protocol and which is to be made available to MANET routing protocols. An Information Base may be associated with a MANET interface or with a router.

Furthermore, this document uses the following notational conventions:

X contains y, or y is contained in X
An unordered list membership operator. X is an unordered list and y is an element. "X contains y" or "y is contained in X" returns true if the unordered list X includes the element y, and returns false otherwise.

X contains Y, or Y is contained in X
An unordered list inclusion operator. X and Y are both unordered lists. "X contains Y" or "Y is contained in X" returns true if the unordered list X contains all elements y which are contained in Y, and returns false otherwise.

A overlaps B
If A and B are network addresses, and the range of addresses represented by A and the range of addresses represented by B both contain at least one common address. (This is only possible if one range is a sub-range of the other.)

a := b
An assignment operator, whereby the left side (a) is assigned the value of the right side (b). a and b may be values, network addresses, or unordered lists (they must be of the same type).

c = d
A comparison operator, returning true if the value of the left side (c) is equal to the value of the right side (d). c and d may be values, network addresses, or unordered lists (they must be of the same type). If c and d are unordered lists, then they are considered to be equal if c contains d and d contains c (i.e., they contain the same set of elements, regardless of the order in which they are recorded in the lists). If c and d are network addresses, they are considered equal only if both addresses and prefix lengths are equal (i.e., they represent the same).

e != f
A comparison operator, returning not (e = f), i.e., returning true where (e = f) would have returned false, and returning false where (e = f) would have returned true.

3. Applicability Statement

This protocol:

- Is applicable to networks, especially wireless networks, in which unknown neighbors can be reached by local broadcast or multicast packets.

- Is designed to work in networks with a dynamic topology, and in which messages may be lost, such as due to collisions in wireless networks.

- Supports routers that each have one or more participating MANET interfaces. The set of a router’s interfaces may change over time. Each interface may have one or more associated network addresses, and these may also be dynamically changing.

- Provides each router with current local topology information up to two hops away, and makes this local topology information available to MANET routing protocols in Information Bases.

- Uses the packet and message formats specified in [RFC5444]. This includes the definition of a HELLO Message Type, used for signaling local topology information.

- Allows "external" and "internal" extensibility as enabled by [RFC5444].

- May interact with, and be extended by, other protocols, such as MANET routing protocols, see Section 16.

- Can use relevant link-layer information if it is available.

- Is designed to work in a completely distributed manner, and does not depend on any central entity.

4. Protocol Overview and Functioning

The objective of this protocol is, for each router:

- To identify 1-hop neighbors and symmetric 1-hop neighbors of this router.

- To find the interface network addresses of the router's symmetric 2-hop neighbors.

- To record this information in Information Bases and thus make this information available to other protocols that use this neighborhood discovery protocol.

- To be available for use by other protocols, which MAY extend the information in these Information Bases, including adding new Sets to Information Bases, new elements to protocol Tuples and new TLVs to be included in outgoing HELLO messages and processed when in incoming HELLO messages.

These objectives are achieved using local (1-hop) signaling that:

- Advertises the presence of a router and its interface network addresses.

- Discovers links from neighboring routers.

- Performs bi-directionality checks on the discovered links.

- Advertises discovered links, and whether each is symmetric, to its 1-hop neighbors, and hence discovers symmetric 2-hop neighbors.

This specification defines, in turn:

- Parameters and constants used by the protocol. Parameters used by this protocol may, where appropriate, be specific to a given MANET interface or to a MANET router. This protocol allows all parameters to be changed dynamically, and to be set independently for each MANET router or MANET interface, as appropriate.

- o The Information Bases describing local interfaces, discovered links and their status, current and former 1-hop neighbors, and symmetric 2-hop neighbors.
- o The format of the HELLO message that is used for local signaling.
- o The generation of HELLO messages from some of the information in the Information Bases.
- o The updating of the Information Bases according to received HELLO messages and other events.
- o The response to other events, such as the expiration of information in the Information Bases.

4.1. Routers and Interfaces

In order for a router to participate in a MANET, it MUST have at least one, and possibly more, MANET interfaces.

Each MANET interface:

- o Is configured with one or more network addresses. Each address in the range of addresses represented by that network address MUST satisfy the following properties:
  - o It MUST be unique to this router, i.e., it MUST NOT be assigned to any interface of any other router.
  - o If assigned to other MANET interfaces, on the same router, these MANET interfaces MUST be isolated, i.e., addresses may only be assigned to different interfaces on the same router if no MANET interface on another router can communicate with both. For example, interfaces using distinct radio "channels" MAY be assigned the same address.
- o Has a set of interface parameters, defining the behavior of this MANET interface. Each MANET interface MAY be individually parameterized.
- o Has an Interface Information Base, recording information regarding links to this MANET interface and symmetric 2-hop neighbors that can be reached through such links.
- o Generates and processes HELLO messages.

In addition to a set of MANET interfaces as described above, each router has:

- o A Local Information Base, containing the network addresses of the interfaces (MANET and non-MANET) of this router. The contents of this Information Base are not changed by signaling.

- o A Neighbor Information Base, recording information regarding current and recently lost 1-hop neighbors of this router.

A router may have both MANET interfaces and non-MANET interfaces. Interfaces of both of these types are recorded in a router's Local Information Base, which is used, but not updated, by the signaling of this protocol.

4.2. Information Base Overview

Each router maintains protocol state using Information Bases, described in the following sections. Each Information Base consists of a number of Protocol Sets. Each Protocol Set contains a number of Protocol Tuples.

An implementation of this protocol MAY maintain this information in the indicated form, or in any other organization that offers access to this information. In particular, note that it is not necessary to remove Protocol Tuples from Protocol Sets at the exact time indicated, only to behave as if the Protocol Tuples were removed at that time.

Information in the Local Information Base is defined locally and included in HELLO messages. Information in the Interface Information Base and the Neighbor Information Base is determined from received HELLO messages; some of this information may also be included in transmitted HELLO messages. Such information has a limited duration in which it is considered valid. This duration is determined from the VALIDITY_TIME TLV in the HELLO message in which the information is received, which in turn is set by the router that originated the HELLO message, using its corresponding interface parameter H_HOLD_TIME.

Appendix E illustrates the relationship between message reception, included VALIDITY_TIME TLVs, and the duration for which information received in a HELLO message is considered valid. Appendix F illustrates some example topologies and how they correspond to information in the Information Bases.

4.2.1. Local Information Base

Each router maintains a Local Information Base, which contains the router's local configuration information, specifically:

- o The Local Interface Set, which consists of Local Interface Tuples, each of which represents an interface (MANET or non-MANET) of the router.

- o The Removed Interface Address Set, which consists of Removed Interface Address Tuples, each of which records a recently used network address of an interface (MANET or non-MANET) of the router.

The Local Interface Set is used for generating HELLO messages, specifically for determining which interface network addresses are to be included and identified as local interfaces. The Removed Interface Address Set is used to avoid incorrectly recording a formerly used network address as a symmetric 2-hop neighbor's network address.

The Local Information Base is used for generating signaling, but is not itself updated by signaling specified in this document. Updates to the Local Information Base are due to changes of the router configuration, and may be allowed to trigger signaling.

4.2.2. Interface Information Bases

Each router maintains, for each of its MANET interfaces, an Interface Information Base, which contains 1-hop neighborhood and symmetric 2-hop neighborhood information collected from this MANET interface, specifically:

- o A Link Set, which records information about current and recently lost links between this MANET interface and MANET interfaces of 1-hop neighbors. The Link Set consists of Link Tuples, each of which contains information about a single link. Link quality information (see Section 14), when used, is recorded in Link Tuples.

- o A 2-Hop Set, which records the existence of symmetric links between symmetric 1-hop neighbors of this MANET interface and other routers (symmetric 2-hop neighbors). The 2-Hop Set consists of 2-Hop Tuples, each of which records a network address of a symmetric 2-hop neighbor, and all network addresses of the corresponding symmetric 1-hop neighbor.

The Link Set for a MANET interface is used for generating HELLO messages. Specifically, the Link Set information is included to both allow other routers to identify symmetric links and to populate the 2-Hop Set. Recently lost links are recorded in the Link Set for a MANET interface so that they can be advertised in HELLO messages, accelerating their removal from relevant 1-hop neighbors' Link Sets.

The Link Set for a MANET interface is itself updated on receiving a HELLO message, including recording symmetric links as indicated above. The 2-Hop Set for a MANET interface is updated as indicated above, and is not itself used in generating HELLO messages.

4.2.3. Neighbor Information Base

Each router maintains a Neighbor Information Base, which contains collected information about current and recently lost 1-hop neighbors, specifically:

- o The Neighbor Set consists of Neighbor Tuples, each of which records all network addresses of a single 1-hop neighbor. Neighbor Tuples are maintained as long as there are corresponding Link Tuples.

- o The Lost Neighbor Set consists of Lost Neighbor Tuples, each of which records a network address of a recently lost symmetric 1-hop neighbor.

The Neighbor Set associates all network addresses of each 1-hop neighbor. These network addresses may be included when generating a HELLO message, so that other symmetric 1-hop neighbors can record this information in a 2-Hop Set. The Neighbor Set can be updated on receiving a HELLO message.

The Lost Neighbor Set is used to determine which network addresses are to be included in a HELLO message as being lost (of a recently symmetric 1-hop neighbor). The Lost Neighbor Set can itself be updated on receiving a HELLO message.

4.3. Signaling Overview

This protocol contains a signaling mechanism for maintaining the Interface Information Bases and the Neighbor Information Base. If information from the link layer, or any other source, is available and appropriate, it may also be used to update these Information Bases. Such updates are subject to the constraints specified in Appendix B.

Signaling consists of a single type of message, known as a HELLO message. Each router generates HELLO messages on each of its MANET interfaces. HELLO messages are generated independently on each MANET interface of a MANET router; the content of a given HELLO message depends on the MANET interface for which it has been generated.

Each HELLO message:

- o Identifies, as far as is required, the MANET interface for which it is generated and transmitted; this allows recipients of that HELLO message to identify that MANET interface from among those it may hear.

- o Reports the network addresses of other interfaces (MANET and non-MANET) of the router; this allows recipients of that HELLO message to associate a set of network addresses with a single 1-hop neighbor.

- o Includes 1-hop neighbor information from the Information Bases; this allows detection of local symmetric links, and symmetric 2-hop neighbors.

HELLO message generation, and the validity of the information recorded as a consequence of processing a HELLO message, is affected by timers and validity information included in HELLO messages through TLVs. The relationship between message timers and intervals is illustrated in Appendix E.

4.3.1. HELLO Message Generation

HELLO messages are generated by a router for each of its MANET interfaces, and MAY be sent:

- o Proactively, at a regular interval, known as HELLO_INTERVAL. HELLO_INTERVAL may be fixed, or may be dynamic. For example, HELLO_INTERVAL may be backed off due to congestion or network stability.

- o As a response to a change in the router itself, or its 1-hop neighborhood, for example, on first becoming active or in response to a new, lost, or changed status link.

- o In a combination of these proactive and responsive mechanisms.

Jittering of HELLO message generation and transmission SHOULD be used as described in Section 11.2, unless the medium access control mechanism in use prevents any simultaneous transmissions by potentially interfering routers.

HELLO messages MAY be scheduled independently for each MANET interface, or, interface parameters permitting, using the same schedule for all MANET interfaces of a router.

4.3.2. HELLO Message Content

The content of a HELLO message MUST satisfy the following:

- A HELLO message MUST contain all of the network addresses in the Local Interface Set of the router on which the HELLO message is being generated. This includes:
  - At least one network address of each MANET interface of the router.
  - Network addresses that include all source addresses of any IP datagrams sent by the router on any MANET interface.
  - All other network addresses of the router that are to be made known to any other router for any reason.
- For each MANET interface, within every time interval equal to the corresponding REFRESH_INTERVAL, sent HELLO messages MUST collectively include all of the relevant information in the corresponding Link Set and the Neighbor Information Base. Note that when determining whether to include information in a HELLO message, the sender MUST consider all times up to the latest time when it may send its next HELLO message on this MANET interface.
- For each MANET interface, within every time interval equal to the corresponding REFRESH_INTERVAL, sent HELLO messages MUST collectively include all of the relevant information in the corresponding Link Set and the Neighbor Information Base.
- When determining whether to include a given piece of neighbor information in a HELLO message, it is not sufficient to consider whether that information has been sent in the interval of length REFRESH_INTERVAL up to the current time. Instead, the router MUST consider the interval of length REFRESH_INTERVAL that will end at the latest possible time at which the next HELLO message will be sent on this MANET interface. (Normally, this will be HELLO_INTERVAL past the current time, but MAY be earlier if this router elects to divide its neighbor information among more than one HELLO message in order to reduce the size of its HELLO messages.) All neighbor information MUST be sent in this interval, i.e., the router MUST ensure that this HELLO message includes all neighbor information that has not already been included in any HELLO messages sent since the start of this

interval (normally, the current time - (REFRESH_INTERVAL - HELLO_INTERVAL)).

- A HELLO message MUST include exactly one VALIDITY_TIME Message TLV, as specified in [RFC5497], that indicates the length of time for which the message content is to be considered valid, and is therefore to be included in the receiving router's Interface Information Base.

- A periodically generated HELLO message SHOULD include exactly one INTERVAL_TIME Message TLV, as specified in [RFC5497], that indicates the current value of HELLO_INTERVAL for that MANET interface, i.e., the period within which a further HELLO message is guaranteed to be sent on that MANET interface.

4.3.3. HELLO Message Processing

HELLO messages received by a router are used to update the Interface Information Base for the MANET interface over which that HELLO message was received, as well as the Neighbor Information Base of the router. Specifically:

- The router ensures that there is a single Neighbor Tuple corresponding to the originator of that HELLO message.

- The router ensures that there is a Link Tuple, with appropriate status (heard or symmetric) and advertised duration, corresponding to the link between the MANET interfaces on which that HELLO message was sent and received. One or more Lost Neighbor Tuples may be created if the HELLO message reports that the link was lost.

- If the link between the MANET interfaces on which the HELLO message was sent and received is symmetric, then the router ensures that there are the appropriate 2-Hop Tuples, with advertised duration.

The processing defined in this specification handles any unexpected and erroneous information in a HELLO message, maintaining the constraints on Information Base contents specified in Appendix B.

4.4. Link Quality

Some links in a MANET may be marginal, e.g., due to adverse wireless propagation. In order to avoid using such marginal links, a link quality value is recorded in each Link Tuple. A MANET router considers links for which an insufficient link quality is recorded as lost. Modifying the recorded link quality in a Link Tuple is

OPTIONAL. If link quality is not to be modified, it MUST be set to indicate an always usable quality link.

Note that link quality is a "link admittance" mechanism, allowing a router to determine that a given link is too unstable to even consider for use. It is specifically not a link metric nor is it a substitute for one.

Link quality information is only used locally and is not used in signaling. Routers may interoperate whether they are using the same, different, or no link quality information. Link quality information is thus not equivalent to a link metric.

Link quality information is defined in this specification as a normalized, dimensionless value in the interval zero to one, inclusive, where the greater the value, the better the link quality. See Section 14 for further details.

5. Protocol Parameters and Constants

The parameters and constants used in this specification are described in this section.

5.1. Protocol and Port Numbers

This protocol specifies HELLO messages, which are included in packets as defined by [RFC5444]. These packets may be sent using either the "manet" protocol number or the "manet" well-known UDP port number, as specified in [RFC5498].

5.2. Multicast Address

This protocol specifies HELLO messages, which are included in packets as defined by [RFC5444]. These packets may be locally transmitted using the link-local multicast address "LL-MANET-Routers", as specified in [RFC5498].

5.3. Interface Parameters

The interface parameters used by this specification may be classified into the following four categories:

- o Message intervals
- o Information validity times
- o Link quality

- Jitter

These are detailed in the following sections.

Different MANET interfaces (on the same or on different routers) MAY employ different interface parameter values and MAY change their interface parameter values dynamically, subject to the constraints given in this section. A particular case is where all MANET interfaces on all MANET routers within a given MANET employ the same set of interface parameter values.

5.3.1. Message Intervals

HELLO messages serve two principal functions:

- To advertise network addresses of this router's interface to its 1-hop neighbors. The frequency of these advertisements is regulated by the interface parameters HELLO_INTERVAL and HELLO_MIN_INTERVAL.

- To advertise this router's knowledge of each of its 1-hop neighbors. The frequency of the advertisement of each such neighbor is regulated by the interface parameter REFRESH_INTERVAL.

Specifically, these parameters are as follows:

HELLO_INTERVAL:
The maximum time between the transmission of two successive HELLO messages on this MANET interface. If using periodic transmission of HELLO messages, these SHOULD be at a separation of HELLO_INTERVAL, possibly modified by jitter as specified in [RFC5148].

HELLO_MIN_INTERVAL:
The minimum interval between transmission of two successive HELLO messages on this MANET interface. (This minimum interval MAY be modified by jitter, as defined in [RFC5148].)

REFRESH_INTERVAL:
The maximum interval between advertisements, in a HELLO message on this MANET interface, of each 1-hop neighbor network address and its status. In all intervals of length REFRESH_INTERVAL, a router MUST include each 1-hop neighbor network address and its status in at least one HELLO message on this MANET interface. (This may be in the same or in different HELLO messages.)

REFRESH_INTERVAL thus represents the frequency at which a piece of information, as received in HELLO messages, can be expected to be refreshed. Thus, the REFRESH_INTERVAL is used as a basis for determining when such information expires in receiving routers (see Section 5.3.2). HELLO_INTERVAL represents the frequency of HELLO message emissions. Logically, HELLO_INTERVAL cannot be greater than the REFRESH_INTERVAL; otherwise, information cannot be refreshed in a timely manner.

HELLO messages can, however, be sent with a higher frequency. A possible use for sending HELLO messages at such a higher frequency includes sending partial HELLO messages (e.g., accommodating constraints on packet sizes from the underlying medium) refreshing only part of the information in each HELLO message. Another use is for a router to send "empty HELLO messages", advertising its own presence frequently in smaller HELLO messages (e.g., in case HELLO message exchange success rates are used for link quality estimation, or to enable rapid detection by new routers in the neighborhood) in between HELLO messages refreshing neighbor information in other routers.

The following constraints apply to these interface parameters:

- o HELLO_INTERVAL > 0
- o HELLO_MIN_INTERVAL >= 0
- o HELLO_INTERVAL >= HELLO_MIN_INTERVAL
- o REFRESH_INTERVAL >= HELLO_INTERVAL
- o If an INTERVAL_TIME Message TLV as defined in [RFC5497] is included in a HELLO message, then HELLO_INTERVAL MUST be representable as described in [RFC5497].

If REFRESH_INTERVAL > HELLO_INTERVAL, then a router may distribute its neighbor advertisements between HELLO messages in any manner, subject to the constraints above.

In the absence of any changes to the local neighborhood, a router will send a HELLO message on a MANET interface after an (possibly jittered) interval of length HELLO_INTERVAL. For a router to employ this protocol in a purely responsive manner on a MANET interface, i.e., for the router to only send HELLO messages on that MANET interface as a response to external events, HELLO_INTERVAL (and hence also REFRESH_INTERVAL) SHOULD be set sufficiently large, i.e., such that a responsive HELLO message is always expected with a shorter period than this value.

If a router has more than one MANET interface, then, even if the router configures different values of HELLO_INTERVAL on each MANET interface, the router SHOULD configure the same value of HELLO_MIN_INTERVAL on all MANET interfaces on which responsive HELLO messages may be sent. (This ensures that changes observed on one MANET interface are reported on other MANET interfaces, so that 1-hop neighbors connected to the latter can maintain up-to-date 2-hop neighborhood information.)

5.3.2. Information Validity Times

The following interface parameters manage the validity time of link information:

L_HOLD_TIME:
The period of advertisement, on this MANET interface, of former 1-hop neighbor network addresses as lost in HELLO messages, allowing recipients of these HELLO messages to accelerate removal of this information from their Link Sets. L_HOLD_TIME MAY be set to zero, if accelerated information removal is not required.

H_HOLD_TIME:
Used as the Value in the VALIDITY_TIME Message TLV included in all HELLO messages on this MANET interface. It is then used by each router receiving such a HELLO message to indicate the validity of the information taken from that HELLO message and recorded in the receiving router's Information Bases.

Note that as each item of neighbor information is included in HELLO messages within an interval of length REFRESH_INTERVAL, constraints on H_HOLD_TIME are based on REFRESH_INTERVAL, not on HELLO_INTERVAL.

The following constraints apply to these interface parameters:

- L_HOLD_TIME >= 0

- H_HOLD_TIME >= REFRESH_INTERVAL

- If HELLO messages can be lost, then both parameters SHOULD be significantly greater than REFRESH_INTERVAL.

- H_HOLD_TIME MUST be representable as described in [RFC5497].

5.3.3. Link Quality

The following interface parameters manage the usage of link quality (see Section 14):

HYST_ACCEPT:
The link quality threshold at or above which a link becomes usable, if it was not already so.

HYST_REJECT:
The link quality threshold below which a link becomes unusable, if it was not already so.

INITIAL_QUALITY:
The initial quality of a newly identified link.

INITIAL_PENDING:
If true, then a newly identified link is considered pending, and is not usable until the link quality has reached or exceeded the HYST_ACCEPT threshold.

The following constraints apply to these interface parameters:

- 0 <= HYST_REJECT <= HYST_ACCEPT <= 1
- 0 <= INITIAL_QUALITY <= 1.
- If link quality is not updated, then INITIAL_QUALITY >= HYST_ACCEPT.
- If INITIAL_QUALITY >= HYST_ACCEPT, then INITIAL_PENDING := false.
- If INITIAL_QUALITY < HYST_REJECT, then INITIAL_PENDING := true.

5.3.4. Jitter

If jitter, as defined in [RFC5148], is used, then these parameters are as follows:

HP_MAXJITTER:
Represents the value of MAXJITTER used in [RFC5148] for periodically generated HELLO messages on this MANET interface.

HT_MAXJITTER:
Represents the value of MAXJITTER used in [RFC5148] for externally triggered HELLO messages on this MANET interface.

For constraints on these interface parameters, see [RFC5148].

5.4. Router Parameters

The two router parameters defined by this specification are in the category of information validity time.

5.4.1. Information Validity Time

The following router parameter manages the validity time of lost symmetric 1-hop neighbor information:

N_HOLD_TIME:
Used as the period during which former 1-hop neighbor network addresses are advertised as lost in HELLO messages, allowing recipients of these HELLO messages to accelerate removal of this information from their 2-Hop Sets. N_HOLD_TIME MAY be set to zero, if accelerated information removal is not required.

I_HOLD_TIME:
The period for which a recently used local interface network address is recorded.

The following constraints apply to these router parameters:

- N_HOLD_TIME >= 0

- I_HOLD_TIME >= 0

5.5. Parameter Change Constraints

If protocol parameters are changed dynamically, the constraints in this section apply.

HELLO_INTERVAL

- If the HELLO_INTERVAL for a MANET interface increases, then the next HELLO message on this MANET interface MUST be generated according to the previous, shorter, HELLO_INTERVAL. A number of subsequent HELLO messages MAY be generated according to the previous, shorter, HELLO_INTERVAL (but MUST include times according to current parameters). This ensures that "promises" as to timely transmission of a future HELLO message are kept until these previous promises have expired.

- If the HELLO_INTERVAL for a MANET interface decreases, then the following HELLO messages on this MANET interface MUST be generated according to this current, shorter, HELLO_INTERVAL.

REFRESH_INTERVAL

- o If the REFRESH_INTERVAL for a MANET interface increases, then the content of subsequent HELLO messages must be organized such that the specification of the old value of REFRESH_INTERVAL is satisfied for a further period equal to the old value of REFRESH_INTERVAL.

- o If the REFRESH_INTERVAL for a MANET interface decreases, then it MAY be necessary to reschedule HELLO message generation on that MANET interface, in order for the specification of REFRESH_INTERVAL is satisfied from the time of change.

HYST_ACCEPT and HYST_REJECT

- o If HYST_ACCEPT or HYST_REJECT changes, then the appropriate actions for link quality change, as specified in Section 14.3, MUST be taken.

L_HOLD_TIME

- o If L_HOLD_TIME changes, then the expiry times of all relevant Link Tuples MUST be changed.

N_HOLD_TIME

- o If N_HOLD_TIME changes, then the expiry times of all relevant Lost Neighbor Tuples MUST be changed.

HP_MAXJITTER

- o If HP_MAXJITTER changes, then the periodic HELLO message schedule on this MANET interface MAY be changed.

HT_MAXJITTER

- o If HT_MAXJITTER changes, then externally triggered HELLO messages on this MANET interface MAY be rescheduled.

5.6. Constants

The constant C (time granularity) is used as specified in [RFC5497].

6. Local Information Base

A router maintains a Local Information Base that records information about its interfaces (MANET and non-MANET). Each interface of a router MUST be identified by at least one network address. Such network addresses MAY be specific to that interface, or MAY in some circumstances be used by more than one interface as specified below.

The Local Information Base is not modified by signaling. If a router's interface configuration changes, then the Local Information Base MUST reflect these changes. This MAY also result in signaling to advertise these changes.

It is not necessary to include all addresses of an interface in the Local Information Base, and hence in HELLO messages. However, any address that may be the source address of an IP datagram sent from that interface MUST be included (and at least one address MUST be included). A protocol using this specification MAY add additional requirements to these, e.g., that any address that may be the destination address of an IP datagram is also included.

The addresses assigned to an interface are "owned" by the router, and MUST NOT be used by any other router as an interface address. If an address has a prefix length and represents a range of addresses, this applies to all addresses in that range (i.e., such ranges MUST NOT overlap).

The addresses assigned to different interfaces on the same router MUST be distinct (hence, network address ranges MUST NOT overlap) except that:

- o The same address MAY be assigned to any number of non-MANET interfaces as well as to one (or more if the following condition also applies) MANET interface.
- o The same address MAY be assigned to two (or more if each pair satisfies this condition) MANET interfaces if those two MANET interfaces cannot communicate to, from, or one to and one from any other MANET interface of another router.

6.1. Local Interface Set

A router's Local Interface Set records its local interfaces. It consists of Local Interface Tuples, one per interface:

(I_local_iface_addr_list, I_manet)

where:

I_local_iface_addr_list is an unordered list of the network addresses of this interface.

I_manet is a boolean flag, describing if this interface is a MANET interface.

Local Interface Tuples are removed from the Local Interface Set only when the routers' interface configuration changes, subject to Section 9, i.e., they are not subject to timer-based expiration.

6.2. Removed Interface Address Set

A router's Removed Interface Address Set records network addresses that were recently used as local interface network addresses. If a router's interface network addresses are immutable, then the Removed Interface Address Set is always empty and MAY be omitted. It consists of Removed Interface Address Tuples, one per network address:

(IR_local_iface_addr, IR_time)

where:

IR_local_iface_addr is a recently used network address of an interface of this router.

IR_time specifies when this Tuple expires and MUST be removed.

7. Interface Information Bases

A router maintains an Interface Information Base for each of its MANET interfaces. This records information about links to that MANET interface and symmetric 2-hop neighbors that can be reached in two hops using those links as the first hop. Each Interface Information Base includes a Link Set and a 2-Hop Set.

A network address of a symmetric 2-hop neighbor can also be present as the network address of a 1-hop neighbor. This allows the router using this network address to be immediately considered as a symmetric 2-hop neighbor if it fails to be a symmetric 1-hop neighbor.

An element that specifies a time is considered expired if the current time is greater than or equal to that time. One such element, present in most Tuples, indicates, when expired, that the Tuple itself is considered expired and MUST be removed. Tuples that do not have such a time element are removed at other times as specified; for example, a Neighbor Tuple is removed when all corresponding Link Tuples have been removed.

7.1. Link Set

An interface's Link Set records links from other routers that are, or recently were, 1-hop neighbors. It consists of Link Tuples, each representing a single link:

(L_neighbor_iface_addr_list, L_HEARD_time, L_SYM_time, L_quality, L_pending, L_lost, L_time)

where:

L_neighbor_iface_addr_list is an unordered list of the network addresses of the MANET interface of the 1-hop neighbor;

L_HEARD_time is the time up to which the MANET interface of the 1-hop neighbor would be considered heard if not considering link quality;

L_SYM_time is the time up to which the link to the 1-hop neighbor would be considered symmetric if not considering link quality;

L_quality is a dimensionless number between 0 (inclusive) and 1 (inclusive) describing the quality of a link; a greater value of L_quality indicating a higher quality link;

L_pending is a boolean flag, describing if a link is considered pending (i.e., a candidate, but not yet established, link);

L_lost is a boolean flag, describing if a link is considered lost due to low link quality;

L_time specifies when this Tuple expires and MUST be removed.

The status of the link, as obtained through HELLO message exchange and using link quality, is denoted L_status. L_status can take the Values PENDING, HEARD, SYMMETRIC, and LOST. The values LOST, SYMMETRIC, and HEARD are defined in Section 18.5. The Value PENDING is never used in a HELLO message, it is only used locally by a router, and any Value distinct from LOST, SYMMETRIC, and HEARD may be used.

L_status is defined by:

1. If L_pending = true, then L_status := PENDING;

2. otherwise, if L_lost = true, then L_status := LOST;

3. otherwise, if L_SYM_time is not expired, then L_status := SYMMETRIC;

4. otherwise, if L_HEARD_time is not expired, then L_status := HEARD;

5. otherwise, L_status := LOST.

7.2. 2-Hop Set

An interface's 2-Hop Set records network addresses of symmetric 2-hop neighbors, and the symmetric links to symmetric 1-hop neighbors through which these symmetric 2-hop neighbors can be reached. It consists of 2-Hop Tuples, each representing a single network address of a symmetric 2-hop neighbor, and a single MANET interface of a symmetric 1-hop neighbor.

(N2_neighbor_iface_addr_list, N2_2hop_addr, N2_time)

where:

N2_neighbor_iface_addr_list is an unordered list of the network addresses of the MANET interface of the symmetric 1-hop neighbor from which this information was received;

N2_2hop_addr is a network address of a symmetric 2-hop neighbor that has a symmetric link (using any MANET interface) to the indicated symmetric 1-hop neighbor;

N2_time specifies when this Tuple expires and MUST be removed.

8. Neighbor Information Base

Each router maintains a Neighbor Information Base that records information about network addresses of current and recently symmetric 1-hop neighbors.

8.1. Neighbor Set

A router's Neighbor Set records all network addresses of each 1-hop neighbor. It consists of Neighbor Tuples, each representing a single 1-hop neighbor:

(N_neighbor_addr_list, N_symmetric)

where:

N_neighbor_addr_list is an unordered list of the network addresses of a 1-hop neighbor;

N_symmetric is a boolean flag, describing if this is a symmetric 1-hop neighbor.

Neighbor Tuples are removed from the Neighbor Set only when corresponding Link Tuples expire from the routers' Link Set, i.e., Neighbor Tuples are not directly subject to timer-based expiration.

8.2. Lost Neighbor Set

A router's Lost Neighbor Set records network addresses of routers that recently were symmetric 1-hop neighbors but that are now advertised as lost. It consists of Lost Neighbor Tuples, each representing a single such network address:

(NL_neighbor_addr, NL_time)

where:

NL_neighbor_addr is a network address of a router that recently was a symmetric 1-hop neighbor of this router;

NL_time specifies when this Tuple expires and MUST be removed.

9. Local Information Base Changes

The Local Information Base MUST be updated in response to changes in the router's local interface configuration. These MAY also change the Interface Information Base and the Neighbor Information Base. If any changes to the latter Information Bases satisfy any of the conditions described in Section 13, then those changes MUST be applied immediately, unless noted otherwise below.

A router MAY transmit HELLO messages in response to these changes.

9.1. Adding an Interface

If an interface is added to the router, then this is indicated by the addition of a Local Interface Tuple to the Local Interface Set. If the new interface is a MANET interface, then an initially empty Interface Information Base MUST be created for this new MANET interface. The actions in Section 9.3 MUST be taken for each network address of this added interface. A HELLO message MAY be sent on all MANET interfaces, it SHOULD be sent on the new interface if it is a

MANET interface. If using scheduled messages, then a message schedule MUST be established on the new MANET interface.

9.2. Removing an Interface

If an interface is removed from the router, then this MUST result in changes to the Local Information Base and to the Neighbor Information Base as follows:

1. Identify the Local Interface Tuple that corresponds to the interface to be removed.

2. For each network address (henceforth removed address) in the I_local_iface_addr_list of that Local Interface Tuple, if that network address is not in the I_local_iface_addr_list of any other Local Interface Tuple, then create a Removed Interface Address Tuple with:

   - IR_local_iface_addr := removed address;

   - IR_time := current time + I_HOLD_TIME.

3. Remove that Local Interface Tuple from the Local Interface Set.

4. If the interface to be removed is a MANET interface (i.e., with I_manet = true), then:

   1. Remove the Interface Information Base for that MANET interface;

   2. All Neighbor Tuples for which none of the network addresses in its N_neighbor_addr_list are contained in any L_neighbor_iface_addr_list in any remaining Link Tuple are removed.

If the removed interface is the last MANET interface of the router, then there will be no remaining Interface Information Bases, and the router will no longer participate in this protocol.

After removing the interface and making these changes, a HELLO message MAY be sent on all remaining MANET interfaces.

9.3. Adding a Network Address to an Interface

If a network address is added to an interface, then this is indicated by the addition of a network address to the appropriate I_local_iface_addr_list. The following changes MUST be made to the Information Bases:

1. Remove any Removed Interface Address Tuple whose IR_local_iface_addr is the added network address.

2. Remove any Neighbor Tuples whose N_neighbor_addr_list contains a network address that overlaps the added network address.

3. Remove any Link Tuples, in any Link Set, for which either:

   - L_neighbor_iface_addr_list contains any network address in the N_neighbor_addr_list of any removed Neighbor Tuple; OR

   - L_neighbor_iface_addr_list contains a network address that overlaps the added network address.

   Apply Section 13.2 but not Section 13.3.

4. Remove any Lost Neighbor Tuples whose NL_neighbor_addr overlaps the added network address.

5. Remove any 2-Hop Tuples whose N2_2hop_addr overlaps the added network address.

After adding the network address and making these changes, a HELLO message MAY be sent on all MANET interfaces.

These changes, other than to the appropriate I_local_iface_addr_list, are made in order to maintain consistency of the Information Bases. Specifically, these changes remove any record of any use of this network address by routers in the 1-hop neighborhood or in the symmetric 2-hop neighborhood of this router.

9.4. Removing a Network Address from an Interface

If a network address (henceforth removed address) is removed from an interface, then:

1. Identify the Local Interface Tuple that corresponds to the removed address.

2. If this is the only network address of that interface (the only network address in the corresponding I_local_iface_addr_list), then remove that interface as specified in Section 9.2.

3. Otherwise:

   1. Remove the removed address from this I_local_iface_addr_list.

2. If the removed address is not in the I_local_iface_addr_list of any other Local Interface Tuple, then create a Removed Interface Address Tuple with:

   - IR_local_iface_addr := removed address;

   - IR_time := current time + I_HOLD_TIME.

After removing the network address and making these changes, a HELLO message MAY be sent on all MANET interfaces.

10. Packets and Messages

The packet and message format used by this protocol is defined in [RFC5444], which is used with the following considerations:

- This protocol specifies one Message Type, the HELLO message.

- A HELLO message MAY use any combination of Message Header options specified in [RFC5444].

- HELLO messages MUST NOT be forwarded, i.e., a <msg-hop-limit>, if present, MUST have the value 1.

- HELLO messages MAY be included in multi-message packets as specified in [RFC5444].

- Received HELLO messages MUST be parsed in accordance with [RFC5444]. A HELLO message that is not in conformance with [RFC5444] MUST be discarded without being processed. A HELLO message can also be discarded without being processed for other reasons, see Section 12.1.

- This protocol specifies three Address Block TLVs. It also uses two Message TLVs defined in [RFC5497]. These five TLV Types are all defined only with Type Extension = 0. TLVs of other types, and of these types but without Type Extension = 0, are ignored by this protocol. All references in this specification to, for example, an Address Block TLV with Type = LINK_STATUS, are to be considered as referring to an Address Block TLV with Type = LINK_STATUS and Type Extension = 0.

10.1. HELLO Messages

A HELLO message MUST contain:

- Exactly one VALIDITY_TIME Message TLV as specified in [RFC5497], representing H_HOLD_TIME for the transmitting MANET interface.

The options included in [RFC5497] for representing zero and infinite times MUST NOT be used.

A HELLO message transmitted due to a periodic timer SHOULD contain, and otherwise (i.e., transmitted for any other reason, in particular in response to any Information Base changes) MAY contain:

- o Exactly one INTERVAL_TIME Message TLV as specified in [RFC5497], representing HELLO_INTERVAL for the transmitting MANET interface. The options included in [RFC5497] for representing zero and infinite times MUST NOT be used.

A HELLO message MAY contain:

- o Other Message TLVs.

- o One or more Address Blocks, each with an associated Address Block TLV Block, which MAY contain other Address Block TLVs.

10.1.1. Address Blocks

All network addresses in a router's Local Interface Set (i.e., in any I_local_iface_addr_list) MUST be represented as address objects (see [RFC5444]), and included in the Address Blocks in all generated HELLO messages, with the following permitted exception:

- o If the MANET interface on which the HELLO message is to be sent has a single address with maximum prefix length (i.e., /32 for IPv4, /128 for IPv6), then that address MAY be omitted from being included in any Address Block, provided that this address is included as the sending address of the IP datagram in which the HELLO message is included.

All network addresses of the router's interfaces included in any Address Block(s) MUST be associated with an Address Block TLV with Type = LOCAL_IF, as defined in Table 1.

Name	Value Length	Description
LOCAL_IF	1 octet	Specifies that the network address is associated with the MANET interface on which the message was sent (THIS_IF) or another interface of the sending router (OTHER_IF).

Table 1: LOCAL_IF TLV Definition

Address Blocks MAY contain current or recently lost 1-hop neighbors' network addresses, represented as address objects (see [RFC5444]), each of which is associated with one or both Address Block TLVs as described in Table 2.

Name	Value Length	Description
LINK_STATUS	1 octet	Specifies the status of the link from the indicated network address and to the MANET interface over which the HELLO message is transmitted (LOST, SYMMETRIC, or HEARD).
OTHER_NEIGHB	1 octet	Specifies that the network address is (SYMMETRIC) or was (LOST) of a MANET interface of a symmetric 1-hop neighbor of the router transmitting the HELLO message.

Table 2: LINK_STATUS and OTHER_NEIGHB TLV Definition

An Address Block TLV with Type = LINK_STATUS and Value = SYMMETRIC or Value = LOST also performs the function of an Address Block TLV with Type = OTHER_NEIGHB and the same Value. Including the latter associated with the same address object is discouraged for efficiency reasons. If an Address Block TLV with Type = LINK_STATUS and Value = SYMMETRIC is combined with an Address Block TLV with Type = OTHER_NEIGHB and Value = LOST associated with the same address object, then the latter MUST be ignored and SHOULD NOT be included. See Appendix A.

Other network addresses MAY be represented as address objects (see [RFC5444]) and included in Address Blocks, but MUST NOT be associated with any of the Address Block TLVs specified in this specification.

11. HELLO Message Generation

Each MANET interface MUST generate HELLO messages according to the specification in this section. HELLO messages are generated for each MANET interface independently. HELLO message generation and scheduling MUST be according to the interface parameters for that MANET interface, and MAY be independent for each MANET interface or, interface parameters permitting, MANET interfaces MAY use the same schedule.

If transmitting periodic HELLO messages, then, following a HELLO message transmission on a MANET interface, another HELLO message MUST be transmitted on the same MANET interface after an interval not greater than HELLO_INTERVAL. Two successive HELLO message transmissions on the same MANET interface MUST be separated by at least HELLO_MIN_INTERVAL, except as noted in Section 11.2.1.

11.1. HELLO Message Specification

HELLO messages are generated independently on each MANET interface.

All network addresses in any I_local_iface_addr_list MUST be included, represented as address objects (see [RFC5444]), except that:

- o If the interface on which the HELLO message is to be sent has a single address with maximum prefix length (i.e., /32 for IPv4, /128 for IPv6), then that address MAY be omitted, provided that this address is included as the sending address of the IP datagram in which the HELLO message is included.

All such included address objects MUST be associated with an Address Block TLV with Type := LOCAL_IF and Value according to the following:

- o If the address object represents a network address of the sending MANET interface, then Value := THIS_IF.
- o Otherwise, Value := OTHER_IF.

If such a network address is included in more than one I_local_iface_addr_list, then, for efficiency reasons, it is encouraged that the corresponding address object is associated with only one Value using an Address Block TLV with Type := LOCAL_IF; this MUST be Value := THIS_IF if the address object represents a network address of the sending MANET interface.

The following network addresses of current or former 1-hop neighbors MAY be represented as address objects (see [RFC5444]) and included in any HELLO message, respecting the parameter REFRESH_INTERVAL for each association for that MANET interface, and according to the following:

- o Network addresses of MANET interfaces of 1-hop neighbors from the Link Set of the Interface Information Base for this MANET interface (i.e., from an L_neighbor_iface_addr_list), other than those from Link Tuples with L_status = PENDING.

- o Other network addresses of symmetric 1-hop neighbors from the Neighbor Set of this router's Neighbor Information Base (i.e., from an N_neighbor_addr_list).

- o Network addresses of MANET interfaces of previously symmetric or heard 1-hop neighbors connected on this MANET interface from the Link Set of the Interface Information Base for this MANET interface (i.e., from an L_neighbor_iface_addr_list with L_status = LOST).

- o Other network addresses of previously symmetric 1-hop neighbors from the Lost Neighbor Set of this router's Neighbor Information Base (i.e., from an NL_neighbor_addr).

Each such address object (see [RFC5444]) MUST be associated with one or more appropriate Address Block TLVs. Specifically:

1. For each address object (henceforth linked address) that represents a network address contained in an L_neighbor_iface_addr_list of a Link Tuple in the Link Set for this MANET interface, for which L_status != PENDING, include the linked address with an associated Address Block TLV with:

   - o Type := LINK_STATUS; AND

   - o Value := L_status.

2. For each address object (henceforth neighbor address) that represents a network address contained in an N_neighbor_addr_list in a Neighbor Tuple with N_symmetric = true and that has not already been included with an associated Address Block TLV with Type = LINK_STATUS and Value = SYMMETRIC, include the neighbor network address with an associated Address Block TLV with:

   - o Type := OTHER_NEIGHB; AND

   - o Value := SYMMETRIC.

3. For each Lost Neighbor Tuple whose NL_neighbor_addr (henceforth lost address) has not already been represented as an address object and included, include lost address with an associated Address Block TLV with:

   - o Type := OTHER_NEIGHB; AND

   - o Value := LOST.

If any such network addresses have been added to these Information Bases since the last HELLO message was sent on this MANET interface, or if their status (as indicated by these TLVs and the Values they associate with that network address) have changed since that network address was last reported on this MANET interface, then that network address, and the indicated TLVs, SHOULD be included in the HELLO message.

If the address object (see [RFC5444]) representing a network address of a 1-hop neighbor is specified with more than one associated Address Block TLV, then these Address Block TLVs MAY be independently included or excluded from each HELLO message. Each such Address Block TLV MUST be included associated with the address object representation of that network address in a HELLO message sent on that MANET interface in every interval of length equal to that MANET interface's parameter REFRESH_INTERVAL. Address Block TLVs associated with the same address object included in the same HELLO message MAY be applied to the same or different copies of that address object.

An implementation of this protocol MAY limit the information included in each HELLO message, for example, to accommodate smaller MTU sizes. HELLO messages remain constrained by the above requirements, in particular, that all local interface information MUST be included in all HELLO messages, and that all neighbor information MUST be sent within each interval of length REFRESH_INTERVAL. This neighbor information MAY, however, be sent in the same or in different HELLO messages.

11.2. HELLO Message Transmission

HELLO messages are transmitted in the format specified by [RFC5444].

11.2.1. HELLO Message Jitter

HELLO messages MAY be sent using periodic message generation or externally triggered message generation. If using data link and physical layers that are subject to packet loss due to collisions, HELLO messages SHOULD be jittered as described in [RFC5148].

Internally triggered message generation (such as due to a change in local interfaces) MAY be treated as if externally generated message generation or MAY be not jittered.

HELLO messages MUST NOT be forwarded, and thus message forwarding jitter does not apply to HELLO messages.

Each form of jitter described in [RFC5148] requires a parameter MAXJITTER. These interface parameters may be dynamic and are specified by:

- For periodic message generation: HP_MAXJITTER.
- For externally triggered message generation: HT_MAXJITTER.

When HELLO message generation is delayed in order that a HELLO message is not sent within HELLO_MIN_INTERVAL of the previous HELLO message on the same MANET interface, then HELLO_MIN_INTERVAL SHOULD be reduced by jitter, with maximum reduction HP_MAXJITTER, as described in [RFC5148]. In this case, HP_MAXJITTER MUST NOT be greater than HELLO_MIN_INTERVAL.

12. HELLO Message Processing

On receiving a HELLO message, a router MUST first check if the message is invalid for processing by this router, as defined in Section 12.1 and, if so, discard the message without further processing. Otherwise, for each received and valid HELLO message, the receiving router MUST update its appropriate Interface Information Base and its Neighbor Information Base as specified in Section 12.3 to Section 12.6. These updates MUST be performed in the order in which they are presented in this specification. If any changes satisfy any of the conditions described in Section 13, then the indicated consequences in that section MUST be applied immediately, unless noted otherwise.

All message processing, including determination of whether a message is invalid, considers only TLVs with Type Extension = 0. TLVs with any other type extension are ignored. All references to, for example, a TLV with Type = LINK_STATUS refer to a TLV with Type = LINK_STATUS and Type Extension = 0.

12.1. Invalid Message

A received HELLO message is invalid for processing by this router if any of the following conditions are true:

- The address length as specified in the Message Header is not equal to the length of the addresses used by this router.
- The message has a <msg-hop-limit> field in its Message Header with a value other than one. (Note that the message does not need to have a <msg-hop-limit> field.)

- o The message has a <msg-hop-count> field in its Message Header with a value other than zero. (Note that the message does not need to have a <msg-hop-count> field.)

- o The message does not have a Message TLV with Type = VALIDITY_TIME in its Message TLV Block.

- o The message has more than one Message TLV with Type = VALIDITY_TIME in its Message TLV Block.

- o The message has more than one Message TLV with Type = INTERVAL_TIME in its Message TLV Block.

- o The message has any Address Block TLV(s) with Type = LOCAL_IF and any single Value v such that v != THIS_IF and v != OTHER_IF.

- o Any address object (including different address objects representing the same network address, in the same or different Address Blocks) is associated with more than one Value by one or more Address Block TLV(s) with Type = LOCAL_IF.

- o Any address object (henceforth local address) associated with an Address Block TLV with Type = LOCAL_IF represents one of the receiving router’s current or recently used network addresses (i.e., local address overlaps any network address in any I_local_iface_addr_list in the Local Interface Set or any IR_local_iface_addr in the Removed Interface Address Set).

- o The message has any Address Block TLV(s) with Type = LINK_STATUS with any single Value v such that v != LOST, v != SYMMETRIC, and v != HEARD.

- o The message has any Address Block TLV(s) with Type = OTHER_NEIGHB with any single Value v such that v != LOST and v != SYMMETRIC.

- o Any address object (including different copies of an address object, in the same or different Address Blocks) is associated with an Address Block TLV with Type = LOCAL_IF and with an Address Block TLV with Type = LINK_STATUS.

- o Any address object (including different copies of an address object, in the same or different Address Blocks) is associated with an Address Block TLV with Type = LOCAL_IF and with an Address Block TLV with Type = OTHER_NEIGHB.

- Any address object (including different copies of an address object, in the same or different Address Blocks) is associated with more than one Value by one or more Address Block TLVs with Type = LINK_STATUS.

- Any address object (including different copies of an address object, in the same or different Address Blocks) is associated with more than one Value by one or more Address Block TLVs with Type = OTHER_NEIGHB.

A router MAY recognize additional reasons for identifying that a message is badly formed and therefore invalid for processing, e.g., to allow a security protocol as suggested in Section 17 to perform verification of HELLO message signatures and prevent processing of unverifiable HELLO messages by this protocol.

An invalid message MUST be silently discarded, without updating the router's Information Bases.

12.2. Definitions

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

- "validity time" is calculated from the Message TLV with Type = VALIDITY_TIME of the HELLO message as specified in [RFC5497]. (Note that, as specified by Section 12.1, there must be exactly one such Message TLV in the HELLO message.) All information in the HELLO message used by this specification has the same validity time.

- "Receiving Address List" is the I_local_iface_addr_list corresponding to the MANET interface on which the HELLO message was received

- "Sending Address List" is an unordered list of network addresses of the MANET interface over which the HELLO message was sent, i.e., is an unordered list of the network addresses represented by address objects contained in the HELLO message with an associated Address Block TLV with Type = LOCAL_IF and Value = THIS_IF. If the Sending Address List is otherwise empty, then the Sending Address List contains a single network address with maximum prefix length (i.e., /32 for IPv4, /128 for IPv6) with an address equal to the sending address of the IP datagram in which the HELLO message was included.

- "Neighbor Address List" is an unordered list of all the network addresses of all the interfaces of the router that generated the HELLO message, i.e., is the Sending Address List, plus the network addresses represented by address objects contained in the HELLO message with an associated Address Block TLV with Type = LOCAL_IF and Value = OTHER_IF.

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

- "Removed Address List" is a list of network addresses created by this HELLO message processing that were formerly reported as local by the router originating the HELLO message but that are not included in the Neighbor Address List. This list is initialized as empty.

- "Lost Address List" is a subset of the Removed Address List containing network addresses that were formerly considered as symmetric. This list is initialized as empty.

12.3. Updating the Neighbor Set

On receiving a HELLO message, the router MUST update its Neighbor Set and populate the Removed Address List and Lost Address List:

1. Find all Neighbor Tuples (henceforth matching Neighbor Tuples) where N_neighbor_addr_list contains any network address that overlaps with any network address in the Neighbor Address List.

2. If there are no matching Neighbor Tuples, then:

    1. Create a new Neighbor Tuple with:

        - N_neighbor_addr_list := Neighbor Address List;

        - N_symmetric := false.

3. If there is one matching Neighbor Tuple, then:

    1. If the matching Neighbor Tuple's N_neighbor_addr_list != Neighbor Address List, then:

        1. For each network address (henceforth removed address) that is contained in that N_neighbor_addr_list but that is not contained in the Neighbor Address List:

            1. Add the removed address to the Removed Address List.

            2. If N_symmetric = true, then add the removed address to the Lost Address List.

2. Update the matching Neighbor Tuple by:

   - o N_neighbor_addr_list := Neighbor Address List.

4. If there are two or more matching Neighbor Tuples, then:

   1. For each network address (henceforth removed address) that is contained in the N_neighbor_addr_list of any of the matching Neighbor Tuples but is not contained in the Neighbor Address List:

      1. Add removed address to the Removed Address List.

      2. If N_symmetric = true, then add removed address to the Lost Address List.

   2. Replace the matching Neighbor Tuples by a single Neighbor Tuple with:

      - o N_neighbor_addr_list := Neighbor Address List;

      - o N_symmetric := false

12.4. Updating the Lost Neighbor Set

On receiving a HELLO message, a router MUST update its Lost Neighbor Set:

1. For each network address (henceforth lost address) that is contained in the Lost Address List, if no Lost Neighbor Tuple with NL_neighbor_addr = lost address exists, then add a Lost Neighbor Tuple with:

   - o NL_neighbor_addr := lost address;

   - o NL_time := current time + N_HOLD_TIME.

12.5. Updating the Link Sets

On receiving a HELLO message, a router MUST update its Link Sets:

1. Remove all network addresses in the Removed Address List from the L_neighbor_iface_addr_list of all Link Tuples.

2. Remove all Link Tuples whose L_neighbor_iface_addr_list is now empty; apply Section 13.2 but not Section 13.3.

The router MUST then update its Link Set for the MANET interface on which the HELLO message is received:

1. Find all Link Tuples (henceforth matching Link Tuples) where:

   - o L_neighbor_iface_addr_list contains one or more network addresses in the Sending Address List.

2. If there is more than one matching Link Tuple, then remove them all; apply Section 13.2 but not Section 13.3.

3. If no matching Link Tuples remain, then create a new matching Link Tuple with:

   - o L_neighbor_iface_addr_list := empty;

   - o L_HEARD_time := EXPIRED;

   - o L_SYM_time := EXPIRED;

   - o L_quality := INITIAL_QUALITY;

   - o L_pending := INITIAL_PENDING;

   - o L_lost := false;

   - o L_time := current time + validity time.

4. The matching Link Tuple, existing or new, is then modified as follows:

   1. If the router finds any network address (henceforth receiving address) in the Receiving Address List in an Address Block in the HELLO message, then the Link Tuple is modified as follows:

      1. If any receiving address in the HELLO message is associated with an Address Block TLV with Type = LINK_STATUS and with Value = HEARD or Value = SYMMETRIC, then:

         - o L_SYM_time := current time + validity time.

      2. Otherwise, if any receiving address in the HELLO message is associated with an Address Block TLV with Type = LINK_STATUS and Value = LOST, then:

         1. if L_SYM_time has not expired, then:

1. L_SYM_time := EXPIRED.

2. if L_status = HEARD, then:

   o L_time := current time + L_HOLD_TIME.

2. L_neighbor_iface_addr_list := Sending Address List.

3. L_HEARD_time := max(current time + validity time, L_SYM_time).

4. If L_status = PENDING, then:

   o L_time := max(L_time, L_HEARD_time).

5. Otherwise, if L_status = HEARD or L_status = SYMMETRIC, then:

   o L_time := max(L_time, L_HEARD_time + L_HOLD_TIME).

12.6. Updating the 2-Hop Set

On receiving a HELLO message, a router MUST update its 2-Hop Set for the MANET interface on which the HELLO message was received:

1. Remove all network addresses in the Removed Address List from the N2_neighbor_iface_addr_list of all 2-Hop Tuples.

2. If the Link Tuple whose L_neighbor_iface_addr_list = Sending Address List, has L_status = SYMMETRIC, then:

   1. For each network address (henceforth 2-hop address) in an Address Block of the HELLO message, where:

      o a 2-hop address is not contained in the Neighbor Address List;

      o a 2-hop address is not contained in any I_local_iface_addr_list; AND

      o a 2-hop address != any IR_local_iface_addr

      perform the following processing:

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

         o Type = LINK_STATUS and Value = SYMMETRIC; OR o Type = OTHER_NEIGHB and Value = SYMMETRIC,

then, if there is no 2-Hop Tuple such that:

o N2_neighbor_iface_addr_list contains one or more network addresses in the Sending Address List; AND

o N2_2hop_addr = 2-hop address,

then create a 2-Hop Neighbor Tuple with:

o N2_2hop_addr := 2-hop address.

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

o N2_neighbor_iface_addr_list := Sending Address List;

o N2_time := current time + validity time.

2. Otherwise, if a 2-hop address has an Address Block TLV with:

o Type = LINK_STATUS and Value = LOST or Value = HEARD; OR

o Type = OTHER_NEIGHB and Value = LOST;

then remove all 2-Hop Tuples with:

o N2_neighbor_iface_addr_list containing one or more network addresses in the Sending Address List; AND

o N2_2hop_addr = 2-hop address.

13. Other Information Base Changes

The Interface and Neighbor Information Bases MUST be changed when certain events occur. These events may result from HELLO message processing or may be otherwise generated (e.g., expiry of timers or link quality changes).

Events that cause changes in the Information Bases are:

o A Link Tuple's L_status changes to SYMMETRIC. In this case, the actions specified in Section 13.1 are performed.

- o A Link Tuple's L_status changes from SYMMETRIC, or the Link Tuple is removed. In this case, the actions specified in Section 13.2 are performed.

- o A Link Tuple's L_HEARD_time expires, or the Link Tuple is removed. In this case, the actions specified in Section 13.3 are performed.

- o Local interface network address changes. In this case, the actions specified in Section 9 are performed.

- o Link quality changes. In this case, the actions specified in Section 14 are performed.

If a Link Tuple is removed, or if L_status changes from SYMMETRIC and L_HEARD_time expires, then the actions specified in Section 13.2 MUST be performed before the actions specified in Section 13.3 are performed for that Link Tuple.

A router MAY report updated information in response to any of these changes in HELLO message(s), subject to the constraints in Section 11.

A router that transmits HELLO messages in response to such changes SHOULD transmit a HELLO message:

- o On all MANET interfaces, if the Neighbor Set changes such as to indicate the change in symmetry of any 1-hop neighbors (including addition or removal of symmetric 1-hop neighbors).

- o Otherwise, on all those MANET interfaces whose Link Set changes such as to indicate a change in L_status of any 1-hop neighbors (including the addition or removal of any 1-hop neighbors, other than those considered pending).

13.1. Link Tuple Symmetric

If, for any Link Tuple that does not have L_status = SYMMETRIC:

- o L_status changes to SYMMETRIC;

then:

1. For the Neighbor Tuple whose N_neighbor_addr_list contains L_neighbor_iface_addr_list, set:

   - o N_symmetric := true.

2. Remove all Lost Neighbor Tuples whose NL_neighbor_addr is contained in that N_neighbor_addr_list.

This includes any newly created Link Tuples whose status is immediately updated such that L_status = SYMMETRIC. (Note that in this specification, all Link Tuples are created such that their L_status is not SYMMETRIC, but a Link Tuple may be immediately updated by subsequent processing of the same HELLO message that caused the creation of the Link Tuple such that the Link Tuple's L_status changes to SYMMETRIC.)

13.2. Link Tuple Not Symmetric

If for any Link Tuple with L_status = SYMMETRIC:

- L_status changes to any other value; OR
- the Link Tuple is removed;

then:

1. All 2-Hop Tuples for the same MANET interface with:

   - N2_neighbor_iface_addr_list contains one or more network addresses in L_neighbor_iface_addr_list;

   are removed.

2. For the Neighbor Tuple whose N_neighbor_addr_list contains L_neighbor_iface_addr_list:

   1. If there are no remaining Link Tuples for any MANET interface where:

      - L_neighbor_iface_addr_list is contained in N_neighbor_addr_list; AND
      - L_status = SYMMETRIC;

      then modify the Neighbor Tuple by:

      1. N_symmetric := false.

      2. For each network address (henceforth neighbor address) in N_neighbor_addr_list, add a Lost Neighbor Tuple with:

         - NL_neighbor_addr := neighbor address;

o NL_time := current time + N_HOLD_TIME.

13.3. Link Tuple Heard Timeout

If, for any Link Tuple:

- o L_HEARD_time expires; OR
- o the Link Tuple is removed;

then:

1. For the Neighbor Tuple whose N_neighbor_addr_list contains L_neighbor_iface_addr_list, if no Link Tuples for any MANET interface remain where:

   - o L_neighbor_iface_addr_list is contained in N_neighbor_addr_list; AND
   - o L_HEARD_time is not expired;

   then remove the Neighbor Tuple.

14. Link Quality

Link quality is a mechanism whereby a router MAY take considerations other than message exchange into account for determining when a link is and is not a candidate for being considered as HEARD or SYMMETRIC. As such, it is a "link admission" mechanism.

Link quality information for a link is generated (e.g., through access to signal to noise ratio, packet loss rate, or other information from the link layer) and used only locally, i.e., is not included in signaling, and routers may interoperate whether they are using the same, different, or no, link quality information. Link quality information is specified as a normalized, dimensionless figure in the interval zero to one, inclusive, a higher value indicating a better link quality.

For deployments where no link quality is used, the considerations in Section 14.1 apply. For deployments where link quality is used, the general principles of link quality usage are described in Section 14.2. Sections 14.3 and 14.4 detail link quality functioning.

14.1. Deployment without Link Quality

In order for a router to not employ link quality, the router MUST define:

- o INITIAL_PENDING := false;

- o INITIAL_QUALITY >= HYST_REJECT (there is no reason not to define INITIAL_QUALITY := 1).

14.2. Basic Principles of Link Quality

To enable link quality usage, the L_quality value of a Link Tuple is used in conjunction with two thresholds, HYST_ACCEPT and HYST_REJECT, to set the flags L_pending and L_lost of that Link Tuple. Based on these flags, the link status to advertise for that Link Tuple is determined as described in Section 7.1.

The use of two thresholds implements link hysteresis, whereby a link that has HYST_REJECT <= L_quality < HYST_ACCEPT may be either accepted or rejected (depending on which threshold it has most recently crossed, or, if neither, on the value of parameter INITIAL_PENDING). With appropriate values of these parameters, this prevents overly rapid changes of link status.

The basic principles of link quality usage are as follows:

- o A router does not advertise a neighbor interface in any state until L_quality is acceptable:

  - o If INITIAL_PENDING = true, then the link is advertised when link L_quality >= HYST_ACCEPT.

  - o Otherwise, the link is advertised when L_quality >= HYST_REJECT.

  A link that is not yet advertised has L_pending = true.

- o Once L_quality >= HYST_ACCEPT, the router sets L_pending := false, indicating that the link can be advertised.

- o A link for which L_pending = false is advertised until its L_quality drops below HYST_REJECT.

- o If a link has L_pending = false and L_quality < HYST_REJECT, the link is LOST and is advertised as such. This link is not reconsidered as a candidate HEARD or SYMMETRIC link until L_quality >= HYST_ACCEPT.

o A link that has an acceptable quality may be advertised as HEARD, SYMMETRIC or LOST according to the exchange of HELLO messages.

In order that these principles can all hold, a router MUST NOT define:

o INITIAL_PENDING = false and INITIAL_QUALITY < HYST_REJECT; OR

o INITIAL_PENDING = true and INITIAL_QUALITY >= HYST_ACCEPT.

14.3. When Link Quality Changes

If L_quality for a link changes, then the following actions MUST be taken:

1. If L_quality >= HYST_ACCEPT, then the corresponding Link Tuple is modified by:

    1. L_pending := false;

    2. L_lost := false;

    3. If L_status = HEARD or L_status = SYMMETRIC, then:

        o L_time := max(L_time, L_HEARD_time + L_HOLD_TIME).

2. If L_status != PENDING, and L_quality < HYST_REJECT, then the corresponding Link Tuple is modified by:

    1. If L_lost = false, then:

        o L_lost := true;

        o L_time := min(L_time, current time + L_HOLD_TIME).

As a result of this processing, the L_STATUS of a Link Tuple may change. In this case, the processing actions corresponding to this change, as specified in Section 13, MUST also be taken.

If L_quality for a link is updated based on HELLO message reception, or on reception of a packet including a HELLO message, then L_quality MUST be updated prior to the HELLO message processing described in Section 12. (If the receipt of the HELLO message, or the packet containing it, creates the Link Tuple, then the Link Tuple MUST be created with the appropriately updated L_quality value, rather than with L_quality := INITIAL_QUALITY.)

14.4. Updating Link Quality

A router MAY update link quality based on any information available to it. Particular cases that MAY be used include:

- o Information from the link layer, such as signal-to-noise ratio or packet acknowledgment reception and loss information.

- o Receipt or loss of control packets. If control packets include a sequential packet sequence number, as defined in [RFC5444], then link quality can be updated when a control packet is received, whether or not it contains a HELLO message. The link quality may then, for example, be based on whether the last N out of M control packets on the link were received, or may use a "leaky integrator" tracking packet reception and loss.

- o Receipt or loss of HELLO messages. If the maximum interval between HELLO messages is known (such as by inclusion in HELLO messages of a Message TLV with Type := INTERVAL_TIME, as defined in [RFC5497]), then the loss of HELLO messages can be determined without the need to receive a later HELLO message. Note that if this case is combined with the previous case, then care must be taken to avoid "double counting" a lost HELLO message in a lost packet.

15. Proposed Values for Parameters and Constants

This section lists the parameters and constants used in the specification of the protocol, and proposed values of each that MAY be used when a single value of each is used.

15.1. Message Interval Interface Parameters

- o HELLO_INTERVAL := 2 seconds

- o HELLO_MIN_INTERVAL := HELLO_INTERVAL/4

- o REFRESH_INTERVAL := HELLO_INTERVAL

15.2. Information Validity Time Interface Parameters

- o H_HOLD_TIME := 3 x REFRESH_INTERVAL

- o L_HOLD_TIME := H_HOLD_TIME

15.3. Information Validity Time Router Parameters

- N_HOLD_TIME := L_HOLD_TIME
- I_HOLD_TIME := N_HOLD_TIME

15.4. Link Quality Interface Parameters

If link quality is changed, then parameter values will depend on the link quality process. If link quality is not changed, then:

- HYST_ACCEPT := 1
- HYST_REJECT := 0
- INITIAL_QUALITY := 1
- INITIAL_PENDING := false

15.5. Jitter Interface Parameters

- HP_MAXJITTER := HELLO_INTERVAL/4
- HT_MAXJITTER := HP_MAXJITTER

15.6. Constants

- C := 1/1024 second

16. Usage with Other Protocols

Other protocols, such as MANET routing protocols, that use neighborhood discovery, may need to interact with this protocol. This protocol is designed to permit such interactions, in particular:

- Through accessing, and possibly extending, the information in the Local Information Base (Section 6), the Interface Information Base (Section 7), and the Neighbor Information Base (Section 8). These Information Bases record the interface configuration of the router, as well as the local connectivity, up to two hops away. All updates to the elements specified in this document are subject to the constraints specified in Appendix B.
- Through accessing an outgoing HELLO message prior to it being transmitted over any MANET interface, and to add information (e.g., TLVs) as specified in [RFC5444]. This may, for example, be to allow a security protocol, as suggested in Section 17, to add a TLV containing a cryptographic signature to the message, or be to

allow inserting relay selection information into a HELLO message by way of adding a TLV to an outgoing HELLO message prior to it being transmitted.

- o Through accessing an incoming HELLO message, and potentially discarding it prior to processing by this protocol. This may, for example, allow a security protocol as suggested in Section 17 to perform verification of HELLO message signatures and prevent processing of unverifiable HELLO messages by this protocol.

- o Through accessing an incoming HELLO message after it has been completely processed by this protocol. This may, in particular, allow a protocol that has added information, such as relay selection information by way of inclusion of appropriate TLVs, access to such information after appropriate updates have been recorded in the Information Bases in this protocol.

- o Through requesting that a HELLO message be generated at a specific time. In that case, HELLO message generation MUST still respect the constraints in Appendix B.

Address objects in HELLO messages are processed according to their associated Address Block TLVs. All such address objects are to be processed according to this specification are associated with Address Block TLVs with Type of LOCAL_IF, OTHER_NEIGHB, or LINK_STATUS (and type extension zero). Address objects not associated with an Address Block TLV of any of these Types are therefore not processed by the protocol described in this specification.

A protocol, such as a MANET routing protocol, interacting with this protocol may need to add information to HELLO messages. This may be in the form of associating TLVs (of Type other than LOCAL_IF, OTHER_NEIGHB, or LINK_STATUS) to address objects already included by this specification.

A protocol, such as a MANET routing protocol, interacting with this protocol may also add information to HELLO messages by inserting address objects not already included by this specification. Such address objects are in the following called "inserted addresses". These inserted addresses may added to Address Blocks already present by virtue of the HELLO message generation in this specification, or may appear in new Address Blocks. In both cases, the following MUST be observed:

- o An inserted address MUST NOT be associated with an Address Block TLV of Type LOCAL_IF, OTHER_NEIGHB, or LINK_STATUS. Consequently, the processing in this specification will ignore such address objects.

- Each inserted address MUST be associated with an Address Block TLV, to be defined by the specification of the protocol inserting the address object. In this way, all addresses present in a HELLO message are associated with an Address Block TLV defining their semantics.

Informally speaking, Address Block TLVs define the semantics of address objects in an Address Block. If an address object in an Address Block does not have any Address Block TLVs associated, that address object has no semantics. Consequently, all protocols using the protocol defined in this specification MUST respect the following:

- An address object in an Address Block, which is not associated with any Address Block TLV, MUST be silently ignored; the mere presence of an address object without an associated Address Block TLV in a HELLO message MUST NOT cause any processing.

A protocol interacting with this protocol MAY also add an originator address to HELLO messages, as specified in [RFC5444]. Such an originator address MUST be unique to the originating router, it MAY be a local interface address of the router. It SHOULD be used consistently, but SHOULD NOT be constrained in any other way.

Strict adherence to these points will enable unambiguous coexistence of future "extensions" to HELLO messages.

In some cases, a protocol interacting with the protocol defined in this specification, may need to recognize which HELLO messages to process and which HELLO messages to discard. It is the responsibility of that protocol to ensure that such messages are suitably identifiable, e.g., through inclusion of a Message TLV or through recognizing an administrative configuration (such as address ranges). Note that such a protocol interacting with this protocol MAY specify such interaction by recognizing an additional reason for discarding a message. As suggested in Section 17 this might, for example, be a security protocol discarding a message that does not carry a Message TLV containing a cryptographic signature.

17. Security Considerations

The objective of this protocol is to allow each router in the network to acquire information describing its 1-hop neighborhood and symmetric 2-hop neighborhood. This is acquired through HELLO message exchange between neighboring routers. This information is made available through the Interface Information Bases and Neighbor Information Base, describing the router’s 1-hop neighborhood and symmetric 2-hop neighborhood.

Under normal circumstances, the information recorded in these Information Bases is correct, i.e., corresponds to the actual network topology, apart from any changes that have not (yet) been tracked by the HELLO message exchanges.

If a router for some reason, whether malice or malfunction, transmits invalid HELLO messages, incorrect information may be recorded in other routers' Information Bases. This protocol specification does, however, prevent inconsistent information from being included in the Information Bases through the specified processing, which maintains the constraints in Appendix B. The exact consequence of information inexactness depends on the use of these Information Bases, and SHOULD therefore be reflected in the specification of protocols that use information provided by this neighborhood discovery protocol.

This section, therefore, firstly outlines the ways in which correctly formed, but still invalid, HELLO messages may appear, in Section 17.1.

Injection of invalid HELLO messages into a network may be prevented in a number of ways. If, for example, a network is deployed in a site to which access is strictly regulated, so that physical access and proximity to the network is prevented, then further security mechanisms to protect against malicious routers injecting invalid HELLO messages may not be required. Similarly, if the link layer over which the network is formed provides appropriate confidentiality, authentication, and integrity, then this may, for a given deployment, suffice to appropriately protect against disclosure of information to an eavesdropper, and against a malicious router injecting invalid HELLO messages. In the latter case, the link layer would discard frames that fail the link-layer checks, without attempting to deliver such frames to IP. Finally, certain usage may be of a nature where disruption of service is of no consequence, or at least not of sufficient consequence to warrant deployment of additional security mechanisms.

A further point to stress, and which follows from the discussions above is, that it will not be the case that "one size security fits all". Different deployments may have different requirements. For example, in a deployment of a low-value sensor network, authentication using a simple message authentication code and shared symmetric keys may suffice, while anything beyond that may require too many computational resources to be viable. Conversely, in, for example, a community network, verifying not only that the originator of a HELLO message "has the right key" but also the precise identity of the originator may be required to be proved, and computational resources may be available to make such a requirement feasible.

Section 17.2, therefore, does not specify a single "one-size-fits-all" mechanism, but rather details how the security suggestions in [RFC5444] are considered for applicability within the context of this protocol, and with the purpose of aiding deployment-specific security mechanisms to be developed.

17.1. Invalid HELLO Messages

A correctly formed, but still invalid, HELLO message may take any of the following forms. Note that a present or absent address object in an Address Block, does not by itself cause a problem. It is the presence, absence, or incorrectness of associated LOCAL_IF, LINK_STATUS, and OTHER_NEIGHB Address Block TLVs that causes problems.

A router may provide false information about its own identity:

- o The HELLO message may contain address objects with an associated LOCAL_IF Address Block TLV that do not correspond to addresses of interfaces of the router transmitting the HELLO message.

- o The HELLO message may omit network addresses, or their associated LOCAL_IF Address Block TLV, of interfaces of the router transmitting the HELLO message (other than the allowed omission of the only local interface network address of the MANET interface over which the HELLO message is transmitted, if that is the case).

- o The HELLO message may incorrectly specify the LOCAL_IF Address Block TLV Value associated with one or more local interface network addresses, indicating incorrectly whether they are associated with the MANET interface over which the HELLO message is transmitted.

A router may provide false information about the identity of other routers:

- o A present LINK_STATUS Address Block TLV may, incorrectly, identify a network address as being of a MANET interface that is or was heard on the MANET interface over which the HELLO message is transmitted.

- o A consistently absent LINK_STATUS Address Block TLV may, incorrectly, fail to identify a network address as being of a MANET interface that is or was heard on the MANET interface over which the HELLO message is transmitted.

- A present OTHER_NEIGHB Address Block TLV may, incorrectly, identify a network address as being of a router that is or was in the sending router's symmetric 1-hop neighborhood.

- A consistently absent OTHER_NEIGHB Address Block TLV may, incorrectly, fail to identify a network address as being of a router that is or was in the sending router's symmetric 1-hop neighborhood.

- The Value of a LINK_STATUS Address Block TLV may incorrectly indicate the status (LOST, SYMMETRIC or HEARD) of the link from a 1-hop neighbor.

- The Value of an OTHER_NEIGHB Address Block TLV may incorrectly indicate the status (LOST or SYMMETRIC) of a symmetric 1-hop neighbor.

17.2. Authentication, Integrity, and Confidentiality Suggestions

The security suggestions in [RFC5444] regarding inclusion of a cryptographic signature in a Message TLV or a Packet TLV can be applied to this protocol. Failure to verify either form of cryptographic signature should cause a HELLO message to be rejected without being processed.

The following simplification of the suggestions for end-to-end authentication for integrity in [RFC5444] may be applied to HELLO messages:

- As the Message Header fields <msg-hop-count> and <msg-hop-limit> are either omitted or will always have the values 0 and 1, respectively, an end to end cryptographic signature can be calculated based on the entire HELLO message, including its unmodified Message Header.

The security mechanisms suggested in [RFC5444] with respect to confidentiality can be directly applied to this protocol.

18. IANA Considerations

This specification defines one Message Type, which has been allocated from the "Message Types" registry of [RFC5444], and three Address Block TLV Types, which have been allocated from the "Address Block TLV Types" registry of [RFC5444].

18.1. Expert Review: Evaluation Guidelines

For the registries where an Expert Review is required, the designated expert SHOULD take the same general recommendations into consideration as are specified by [RFC5444].

18.2. Message Types

This specification defines one Message Type, which has been allocated from the 0-223 range of the "Message Types" namespace defined in [RFC5444], as specified in Table 3.

Type	Description
0	HELLO : Local signaling

Table 3: Message Type Assignment

18.3. Message-Type-Specific TLV Type Registries

IANA has created a registry for Message-Type-specific Message TLVs for HELLO messages, in accordance with Section 6.2.1 of [RFC5444], and with initial assignments and allocation policies as specified in Table 4.

Type	Description	Allocation Policy
128-223	Unassigned	Expert Review

Table 4: HELLO Message-Type-specific Message TLV Types

IANA has created a registry for Message-Type-specific Address Block TLVs for HELLO messages, in accordance with Section 6.2.1 of [RFC5444], and with initial assignments and allocation policies as specified in Table 5.

Type	Description	Allocation Policy
128-223	Unassigned	Expert Review

Table 5: HELLO Message-Type-specific Address Block TLV Types

18.4. Address Block TLV Types

This specification defines three Address Block TLV Types, which have been allocated from the "Address Block TLV Types" namespace defined in [RFC5444]. IANA has made allocations in the 0-127 range for these types. Three new type extension registries have been created, with assignments as specified in Tables 6, 7, and 8. Specifications of these Address Block TLVs are in Section 10.1.1, with Value Constants defined in Section 18.5.

Name	Type	Type extension	Description	Allocation policy
LOCAL_IF	2	0	Specifies that the network address is associated with this local interface of the sending router (THIS_IF = 0) or another local interface of the sending router (OTHER_IF = 1)	
LOCAL_IF	2	1-255	Unassigned	Expert Review

Table 6: Address Block TLV Type Assignment: LOCAL_IF

Name	Type	Type extension	Description	Allocation policy
LINK_STATUS	3	0	Specifies the status of the link from the indicated network address (LOST = 0, SYMMETRIC = 1, or HEARD = 2)	
LINK_STATUS	3	1-255	Unassigned	Expert Review

Table 7: Address Block TLV Type Assignment: LINK_STATUS

Name	Type	Type extension	Description	Allocation policy
OTHER_NEIGHB	4	0	Specifies the status of the relationship with the router that uses the indicated network address on one or more interfaces (LOST = 0, or SYMMETRIC = 1)	
OTHER_NEIGHB	4	1-255	Unassigned	Expert Review

Table 8: Address Block TLV Type Assignment: OTHER_NEIGHB

18.5. LOCAL_IF, LINK_STATUS, and OTHER_NEIGHB Values

Note: This information is recorded here for clarity and for use elsewhere in this specification. The information required by IANA is included in the descriptions of the Address Block TLVs allocated in Section 18.4.

The Values that the LOCAL_IF Address Block TLV can use are the following:

- THIS_IF := 0
- OTHER_IF := 1

The Values that the LINK_STATUS Address Block TLV can use are the following:

- LOST := 0
- SYMMETRIC := 1
- HEARD := 2

The Values that the OTHER_NEIGHB Address Block TLV can use are the following:

- LOST := 0

- SYMMETRIC := 1

19. Contributors

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

- Brian Adamson, NRL, USA, <adamson@itd.nrl.navy.mil>
- Cedric Adjih, INRIA, France, <Cedric.Adjih@inria.fr>
- Thomas Heide Clausen, LIX, France, <T.Clausen@computer.org>
- Justin Dean, NRL, USA, <jdean@itd.nrl.navy.mil>
- Christopher Dearlove, BAE Systems ATC, UK, <chris.dearlove@baesystems.com>
- Philippe Jacquet, INRIA, France, <Philippe.Jacquet@inria.fr>

20. Acknowledgments

The authors would like to acknowledge the team behind OLSRv1, specified in RFC3626 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 (listed alphabetically): Alan Cullen (BAE Systems), Ulrich Herberg (LIX), Satoh Hiroki (Hitachi) Joe Macker (NRL), Charles E. Perkins (WiChorus), Laurent Viennot (INRIA), and the entire IETF MANET working group.

21. References

21.1. Normative References

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

[RFC5148] Clausen, T., Dearlove, C., and B. Adamson, "Jitter Considerations in Mobile Ad Hoc Networks (MANETs)", RFC 5148, February 2008.

[RFC5444] Clausen, T., Dearlove, C., Dean, J., and C. Adjih, "Generalized Mobile Ad Hoc Network (MANET) Packet/Message Format", RFC 5444, February 2009.

[RFC5497] Clausen, T. and C. Dearlove, "Representing Multi-Value Time in Mobile Ad Hoc Networks (MANETs)", RFC 5497, March 2009.

[RFC5498] Chakeres, I., "IANA Allocations for Mobile Ad Hoc Network (MANET) Protocols", RFC 5498, March 2009.

21.2. Informative References

[RFC2501] Corson, M. and J. Macker, "Mobile Ad hoc Networking (MANET): Routing Protocol Performance Issues and Evaluation Considerations", RFC 2501, January 1999.

[RFC3626] Clausen, T. and P. Jacquet, "Optimized Link State Routing Protocol (OLSR)", RFC 3626, October 2003.

[RFC5449] Baccelli, E., Jacquet, P., Nguyen, D., and T. Clausen, "OSPF Multipoint Relay (MPR) Extension for Ad Hoc Networks", RFC 5449, February 2009.

Appendix A. Address Block TLV Combinations

The algorithm for generating HELLO messages in Section 11 specifies which 1-hop neighbor network addresses may be included in the Address Blocks, and with which associated Address Block TLVs. These Address Block TLVs may have Type = LINK_STATUS or Type = OTHER_NEIGHB, or both. Address Block TLVs with Type = LINK_STATUS may have three possible Values (Value = HEARD, Value = SYMMETRIC, or Value = LOST), and Address Block TLVs of TYPE = OTHER_NEIGHB may have two possible Values (Value = SYMMETRIC or Value = LOST). When both Address Block TLVs are associated with the same network address only certain combinations of these Address Block TLV Values are necessary, and are the only combinations generated by the algorithm in Section 11. These combinations are indicated in Table 9.

Cells labeled with "Yes" indicate the possible combinations that are generated by the algorithm in Section 11. Cells labeled with "No" indicate combinations not generated by the algorithm in Section 11 but that are correctly parsed and interpreted by the algorithm in Section 12. The cell labeled with "No*" is actually inconsistent, it is handled by ignoring the Address Block TLV with Type = OTHER_NEIGHB, but SHOULD NOT be used.

	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: LINK_STATUS and OTHER_NEIGHB TLV Combinations

Appendix B. Constraints

Any process that updates the Local Information Base or the Neighbor Information Base MUST ensure that all constraints specified in this appendix are maintained.

In each Local Interface Tuple:

- I_local_iface_addr_list MUST NOT be empty.
- I_local_iface_addr_list MUST NOT contain any duplicated network addresses.
- If I_manet = true, then I_local_iface_addr_list MUST NOT contain any network address that overlaps any network address in the I_local_iface_addr_list of any other Local Interface Tuple with I_manet = true, unless it is known that the corresponding MANET interfaces will always communicate with separate sets of MANET interfaces on other routers.

In each Removed Interface Address Tuple:

- IR_local_iface_addr MUST NOT contain any network address that is in the I_local_iface_addr_list of any Local Interface Tuple.
- IR_local_iface_addr MUST NOT equal the IR_local_iface_addr of any other Removed Interface Address Tuple.

In each Link Tuple:

- L_neighbor_iface_addr_list MUST NOT be empty.
- L_neighbor_iface_addr_list MUST NOT contain any network address that overlaps any network address in the I_local_iface_addr_list of any Local Interface Tuple or the IR_local_iface_addr of any Removed Interface Address Tuple.
- L_neighbor_iface_addr_list MUST NOT contain any duplicated network addresses.
- L_neighbor_iface_addr_list MUST NOT contain any network address which is in the L_neighbor_iface_addr_list of any other Link Tuple in the same Link Set.
- If L_HEARD_time has not expired, then there MUST be a Neighbor Tuple whose N_neighbor_addr_list contains L_neighbor_iface_addr_list.

- L_HEARD_time MUST NOT be greater than L_time.

- L_SYM_time MUST NOT be greater than L_HEARD_time (unless both are expired).

- L_quality MUST NOT be less than 0 or greater than 1.

- If L_quality >= HYST_ACCEPT, then L_pending MUST be false.

- If L_quality < HYST_REJECT, then L_status MUST be PENDING or LOST.

- L_pending MUST NOT be set to true if it is currently false.

In each Neighbor Tuple:

- N_neighbor_addr_list MUST NOT contain any network address that overlaps any network address in the I_local_iface_addr_list of any Local Interface Tuple or the IR_local_iface_addr of any Removed Interface Address Tuple.

- N_neighbor_addr_list MUST NOT contain any duplicated network addresses.

- N_neighbor_addr_list MUST NOT contain any network address that is in the N_neighbor_addr_list of any other Neighbor Tuple.

- If N_symmetric is = true, then there MUST be one or more Link Tuples with:

  - L_neighbor_iface_addr_list contained in N_neighbor_addr_list; AND

  - L_status = SYMMETRIC.

- If N_symmetric is = false, then there MUST be one or more Link Tuples with:

  - L_neighbor_iface_addr_list contained in N_neighbor_addr_list.

  All such Link Tuples MUST NOT have L_status = SYMMETRIC. At least one such Link Tuple MUST have L_HEARD_time not expired.

In each Lost Neighbor Tuple:

- NL_neighbor_addr MUST NOT overlap any network address in the I_local_iface_addr_list of any Local Interface Tuple or the IR_local_iface_addr of any Removed Interface Address Tuple.

- o NL_neighbor_addr MUST NOT equal the NL_neighbor_addr of any other Lost Neighbor Tuple.

- o NL_neighbor_addr MUST NOT be in the N_neighbor_addr_list of any Neighbor Tuple with N_symmetric = true.

In each 2-Hop Tuple:

- o There MUST be a Link Tuple associated with the same MANET interface with:

  - o L_neighbor_iface_addr_list = N2_neighbor_iface_addr_list; AND

  - o L_status = SYMMETRIC.

- o N2_2hop_addr MUST NOT overlap any network address in the I_local_iface_addr_list of any Local Interface Tuple or the IR_local_iface_addr of any Removed Interface Address Tuple.

- o N2_2hop_addr MUST NOT be the N2_2hop_addr of any other 2-Hop Tuple in the same 2-Hop Set and with the same N2_neighbor_iface_addr_list.

- o N2_2hop_addr MUST NOT be in the N2_neighbor_iface_addr_list of the same 2-Hop Tuple.

Appendix C. HELLO Message Example

HELLO messages are instances of [RFC5444] messages, and this protocol supports any combination of message header options and address encodings, enabled by [RFC5444] that convey the required information. As a consequence, there is no single way to represent how all HELLO messages look. This appendix illustrates two HELLO message with similar content, the exact values included are explained in the following text.

The HELLO message’s four bit Message Flags (MF) field has value 7 indicating that the message header contains hop limit, hop count, and message sequence number fields. Its four bit Message Address Length (MAL) field has value 3, indicating addresses in the message have a length of four octets, here being IPv4 addresses. The message is as transmitted, with a hop limit of 1 and a hop count of 0. The overall message length is 45 octets.

The message contains a Message TLV Block with content length 8 octets containing two Message TLVs, of types VALIDITY_TIME and INTERVAL_TIME. Each uses a Message TLV with Flags octet (MTLVF) value 16, indicating that each has a Value, and each has a Value

Length of 1 octet. The Values included are time codes (as defined in [RFC5497]) representing the parameters H_HOLD_TIME and HELLO_INTERVAL, respectively.

The message has a single Address Block containing 5 addresses. The Address Block Flags octet (ABF) value 128 indicates an address Head but no address Tail, and no address prefixes. The Head Length of 3 octets indicates address Mid sections of one octet each (Mid 0 to Mid 4).

The following Address Block TLV Block (content length 14 octets) includes two Address Block TLVs. The first is a LOCAL_IF Address Block TLV with Flags octet (ATLVF) value 80, which indicates that a single address, with index 0 (i.e., the address Head:Mid 0) is the single local interface address of this router (the one octet Value THIS_IF indicates that this address is of this interface). The second is a LINK_STATUS Address Block TLV with Flags octet (ATLVF) value 52, which specifies the link status values of all reported neighbors in a single multivalue Address Block TLV that covers the addresses with indexes 1 to 4, inclusive. The Address Block TLV Value Length of 4 octets indicates one octet per Value per address. The last four addresses thus are of neighbors, each an with associated link status: the first and second HEARD, the third SYMMETRIC, and the fourth LOST.

```
     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     | MF=7  | MAL=3 |      Message Length = 45      |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    | Hop Limit = 1 | Hop Count = 0 |    Message Sequence Number    |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    | Message TLV Block Length = 8  | VALIDITY_TIME |  MTLVF = 16   |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    | Value Len = 1 | Value (Time)  | INTERVAL_TIME |  MTLVF = 16   |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    | Value Len = 1 | Value (Time)  | Num Addrs = 5 |   ABF = 128   |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    | Head Len = 3  |                     Head                      |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |     Mid 0     |     Mid 1     |     Mid 2     |     Mid 3     |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |     Mid 4     | Address TLV Block Length = 14 |   LOCAL_IF    |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |  ATLVF = 80   |   Index = 0   | Value Len = 1 |    THIS_IF    |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |  LINK_STATUS  |   ATLV = 52   | Strt Indx = 1 | Stop Indx = 4 |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    | Value Len = 4 |     HEARD     |     HEARD     |   SYMMETRIC   |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |     LOST      |
    +-+-+-+-+-+-+-+-+
```

Note that this example is for illustrative purposes. The essential information can be conveyed, more efficiently (assuming that the local interface address will be supplied by IP, and that the INTERVAL_TIME TLV is not needed) by the 29 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
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |     HELLO     |0 0 0 0 0 0 1 1|0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 1|
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0| VALIDITY_TIME |0 0 0 1 0 0 0 0|
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |0 0 0 0 0 0 0 1|0 1 1 0 0 1 0 0|0 0 0 0 0 1 0 0|1 0 0 0 0 0 0 0|
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |0 0 0 0 0 0 1 1|                    Head                       |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |     Mid 1     |     Mid 2     |     Mid 3     |     Mid 4     |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1|  LINK_STATUS  |0 0 0 1 0 1 0 0|
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |0 0 0 0 0 1 0 0|     HEARD     |     HEARD     |   SYMMETRIC   |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |     LOST      |
    +-+-+-+-+-+-+-+-+
```

Note that the above example assumes that H_HOLD_TIME and C have their default values of 6 seconds and 1/1024 second, and thus result in a time code of 100 (hexadecimal 64).

Appendix D. Flow and Congestion Control

This protocol specifies one Message Type, the HELLO message. The maximum size of a HELLO message is proportional to the size of the originating router’s 1-hop neighborhood. HELLO messages MUST NOT be forwarded.

A router MUST report its 1-hop neighborhood in HELLO messages on each of its MANET interfaces at least each REFRESH_INTERVAL. This puts a lower bound on the control traffic generated by each router in the network employing this protocol.

A router MUST ensure that two successive HELLO messages sent on the same MANET interface are separated by at least HELLO_MIN_INTERVAL. (If using jitter, then this may be reduced to a mean minimum value of HELLO_MIN_INTERVAL - HP_MAXJITTER/2.) Thus, this puts an upper bound on the control traffic generated by each router in the network employing this protocol.

Appendix E. Interval and Timer Illustrations

This informative appendix illustrates the relationship between message timers and intervals. (The adjustments to this timing when using timing jitter, as defined in [RFC5148], are not shown.)

E.1. HELLO Message Generation Timing

Figure 1 illustrates a basic HELLO message schedule for a router, on a MANET interface, employing strictly periodic transmission of HELLO messages. The router transmits a HELLO message each HELLO_INTERVAL. Each HELLO message contains all 1-hop neighbor network addresses of the router that are to be reported in any such HELLO message. (The parameter REFRESH_INTERVAL, not shown, is in this example equal to the parameter HELLO_INTERVAL.)

The router includes a VALIDITY_TIME TLV in each HELLO message, encoding the parameter H_HOLD_TIME, the duration for which information received in the HELLO message should be considered valid by receiving routers. Receiving routers will, when recording the information received in the HELLO message, each use this for setting the L_HEARD_time, L_SYM_time and L_time elements of their corresponding Link Tuple, and the N2_time in the corresponding 2-Hop Tuple for each network address. Only L_time is illustrated in Figure 1.

```
     H_HOLD_TIME:         |-----------------------------|   ...

     HELLO_INTERVAL:      |---------|---------|---------|   ...

     Time:             ---*---------*---------*---------*------>

                          ^         ^         ^         ^
                          |         |         |         |
         HELLO (a, b, c, d)         |         |         |
                                    |         |         |
                   HELLO (a, b, c, d)         |         |   ...
                                              |         |
                             HELLO (a, b, c, d)         |
                                                        |
                                       HELLO (a, b, c, d)

     L_time:              |-----------------------------|
                                    |--------------------   ...
                                              |----------   ...
                                                        |   ...
```

Figure 1: HELLO Message Generation: Regular Schedule

Figure 2 illustrates a message schedule similar to Figure 1, where the router announces its own presence more frequently by sending additional HELLO messages. HELLO messages are still sent regularly, at a reduced interval defined by a new value of HELLO_INTERVAL. However, REFRESH_INTERVAL has not been reduced. Each 1-hop neighbor network address included in these HELLO messages need be advertised only once within each REFRESH_INTERVAL. Consequently, the additional HELLO messages due to the reduced value of HELLO_INTERVAL may therefore be empty. (This is not the only allowed distribution of 1-hop neighbor network addresses, they could, for example, be sent alternately a, b and c, d.)

```
     H_HOLD_TIME:         |-----------------------------|   ...

     REFRESH_INTERVAL:    |---------|---------|---------|   ...

     HELLO_INTERVAL:      |----|----|----|----|----|----|   ...

     Time:             ---*---------*---------*---------*------>

                          ^    ^    ^    ^    ^    ^    ^
                          |    |    |    |    |    |    |
         HELLO (a, b, c, d)    |    |    |    |    |    |
                               |    |    |    |    |    |
                        HELLO ()    |    |    |    |    |
                                    |    |    |    |    |
                   HELLO (a, b, c, d)    |    |    |    |
                                         |    |    |    |   ...
                                  HELLO ()    |    |    |
                                              |    |    |
                             HELLO (a, b, c, d)    |    |
                                                   |    |
                                            HELLO ()    |
                                                        |
                                       HELLO (a, b, c, d)

     L_time:              |-----------------------------|
                               |-------------------------   ...
                                    |--------------------   ...
                                         |---------------   ...
                                              |----------   ...
                                                   |-----   ...
                                                        |   ...
```

Figure 2: HELLO Message Generation: Regular Schedule with Different HELLO_INTERVAL and REFRESH_INTERVAL

HELLO messages may also be sent in response to events. The minimal interval between two successive HELLO message transmissions by a router is HELLO_MIN_INTERVAL, setting an upper bound of the HELLO message emission rate. Hence, for each HELLO message transmission, a router must wait at least HELLO_MIN_INTERVAL before the next HELLO message transmission. Similarly, the maximum interval between two successive HELLO message transmissions is HELLO_INTERVAL, setting a lower bound on the message transmission rate. Hence, for each HELLO message transmission, the router must ensure that the next HELLO message transmission must not wait more than HELLO_INTERVAL.

Figure 3 illustrates a message schedule similar to Figure 1, but with HELLO messages responding to events at maximum rate, i.e., with HELLO messages being sent each HELLO_MIN_INTERVAL. Note that when a HELLO message is sent, it is assumed that the following messages may all be scheduled at an interval of HELLO_INTERVAL, and hence each HELLO message contains all 1-hop neighbor network addresses. In each HELLO message in this example, a new 1-hop neighbor network address is added, reflecting the changes occurring since the last HELLO message was sent. HELLO messages are sent at the maximum allowed rate in order to signal these changes as rapidly as possible.

```
     H_HOLD_TIME:         |-----------------------------|   ...

     HELLO_INTERVAL:      |---------|---------|---------|   ...

     HELLO_MIN_INTERVAL:  |----|----|----|----|----|----|   ...

     Time:             ---*---------*---------*---------*------>

                          ^    ^    ^    ^    ^    ^    ^
                          |    |    |    |    |    |    |
                   HELLO ()    |    |    |    |    |    |
                               |    |    |    |    |    |
                       HELLO (a)    |    |    |    |    |
                                    |    |    |    |    |
                         HELLO (a, b)    |    |    |    |
                                         |    |    |    |   ...
                           HELLO (a, b, c)    |    |    |
                                              |    |    |
                             HELLO (a, b, c, d)    |    |
                                                   |    |
                               HELLO (a, b, c, d, e)    |
                                                        |
                                 HELLO (a, b, c, d, e, f)

     L_time:              |-----------------------------|
                               |-------------------------   ...
                                    |--------------------   ...
                                         |---------------   ...
                                              |----------   ...
                                                   |-----   ...
                                                        |   ...
```

Figure 3: HELLO Message Generation: Regular Schedule with Responsive Messages

Figure 4 shows the same example as Figure 3, but with an increased REFRESH_INTERVAL, and showing partial HELLO messages that include only the necessary network addresses.

```
     H_HOLD_TIME:         |-----------------------------|   ...

     REFRESH_INTERVAL:    |-------------------|----------   ...
                               |-------------------|-----   ...
                                    |-------------------|   ...
                                         |---------------   ...
                                              |----------   ...
                                                   |-----   ...
                                                        |   ...

     HELLO_INTERVAL:      |---------|---------|---------|   ...

     HELLO_MIN_INTERVAL:  |----|----|----|----|----|----|   ...

     Time:             ---*---------*---------*---------*------>

                          ^    ^    ^    ^    ^    ^    ^
                          |    |    |    |    |    |    |
                   HELLO ()    |    |    |    |    |    |
                               |    |    |    |    |    |
                       HELLO (a)    |    |    |    |    |
                                    |    |    |    |    |
                            HELLO (b)    |    |    |    |
                                         |    |    |    |   ...
                                 HELLO (c)    |    |    |
                                              |    |    |
                                   HELLO (a, d)    |    |
                                                   |    |
                                        HELLO (b, e)    |
                                                        |
                                             HELLO (c, f)

     L_time:              |-----------------------------|
                               |-------------------------   ...
                                    |--------------------   ...
                                         |---------------   ...
                                              |----------   ...
                                                   |-----   ...
                                                        |   ...
```

Figure 4: HELLO Message Generation: Regular Schedule with Responsive Messages and Different HELLO_INTERVAL and REFRESH_INTERVAL

Figure 5 summarizes the overall relationship between the intervals governing HELLO message transmissions by a router.

```
     H_HOLD_TIME:         |------------------------------------|

     REFRESH_INTERVAL:    |----------------|

     HELLO_INTERVAL:      |----------|

     HELLO_MIN_INTERVAL:  |---|

     Time:             ----------------------------------------------->

                          ^   ^      ^     ^                   ^
                          |   |      |     |                   |
                          |   |      |     |           Time up to which
              HELLO message   |      |     |           received HELLO
              transmission    |      |     |           message content
                              |      |     |           is valid.
                              |      |     |
                              |      |     Time before which all
                              |      |     neighbor information must
                              |      |     be transmitted in HELLO
                              |      |     messages (one or more)
                              |      |
                              |      Latest time for next HELLO message
                              |      transmission
                              |
                              Earliest time for next HELLO message
                              transmission
```

Figure 5: HELLO Message Generation Intervals

E.2. HELLO Message Processing Timing

Figure 6 illustrates the Link Set timers when receiving a HELLO message not including the network address of the receiving MANET interface.

```
  VALIDITY_TIME:    |-------------------------|

  L_time:           |-------------------------|

  L_HEARD_time:     |-------------------------|

  L_SYM_time:     *-| (i.e.,  expired)

  Time:          ---*--------------------------------->
                    ^
                    |
           HELLO () received
```

Figure 6: HELLO Message Processing: Network Address Not Present

Figure 7 illustrates the Link Set timers when, following the received HELLO message illustrated in Figure 6, a router receives a HELLO message including the network address (a) of the receiving interface with link status = HEARD (or SYM).

```
  VALIDITY_TIME:    |-------------------------|
                          |-------------------------|

  L_time:           |-------------------------|
                          |-------------------------|

  L_HEARD_time:     |-------------------------|
                          |-------------------------|

  L_SYM_time:     *-| (i.e.,  expired)
  L_SYM_time:             |-------------------------|

  Time:            -*-----*---------------------------------->
                    ^     ^
                    |     |
   HELLO () received      |
                          |
   HELLO (a:HEARD) received
```

Figure 7: HELLO Message Processing: Network Address Present

Figure 8 illustrates the Link Set timers when, following the received HELLO messages illustrated in Figure 7, a router receives a HELLO message including the network address (a) of the receiving interface with link status = LOST.

```
     VALIDITY_TIME:    |--------------------------|
                             |--------------------------|
                                   |--------------------------|

     L_time:           |--------------------------|
                             |--------------------------|
                                   |--------------------------|

     L_HEARD_time:     |--------------------------|
                             |--------------------------|
                                   |--------------------------|

     L_SYM_time:     *-| (i.e.,  expired)
                             |--------------------------|
                                 *-| (i.e.,  expired)

     Time:            -*-----*-----*--------------------------------->
                       ^     ^     ^
                       |     |     |
       HELLO () received     |     |
                             |     |
      HELLO (a:HEARD) received     |
                                   |
             HELLO (a:LOST) received
```

Figure 8: HELLO Message Processing: Network Address Lost

E.3. Other HELLO Message Timing

There are three other timing parameters that are used by a router to control HELLO message generation and processing.

Figure 9 illustrates the time, with duration L_HOLD_TIME, during which the appropriate network addresses of a formerly, but no longer, symmetric 1-hop neighbor, as connected by this MANET interface, are advertised as LOST using a LINK_STATUS TLV in HELLO messages on this MANET interface, thus allowing that 1-hop neighbor to update its Link Set accordingly.

```
     L_HOLD_TIME:   |----------------------------|

     Time:       ---*------------------------------------->
                    ^                            ^
                    |                            |
         Formerly symmetric 1-hop neighbor       |
         ceases to be symmetric on this          |
         MANET interface                         |
                                                 |
                     Time up to which network addresses of
                     this neighbor connected using this MANET
                     interface are advertised in HELLO
                     messages on this MANET interface
                     using a LINK_STATUS TLV, Value = LOST
```

Figure 9: HELLO Message Generation: Advertisement of Formerly Symmetric 1-Hop Neighbor on This MANET Interface as Lost

Figure 10 illustrates the time, with duration N_HOLD_TIME, during which all network addresses of a formerly, but no longer, symmetric 1-hop neighbor, are advertised as LOST in HELLO messages on all MANET interfaces using an OTHER_NEIGHB TLV (if not also reported using a LINK_STATUS TLV) thus allowing all other symmetric 1-hop neighbors to update their 2-Hop Sets accordingly.

```
     L_HOLD_TIME:   |----------------------------|

     Time:       ---*------------------------------------->
                    ^                            ^
                    |                            |
         Formerly symmetric 1-hop neighbor       |
         ceases to be symmetric                  |
                                                 |
                     Time up to which network addresses of
                     this neighbor are advertised in HELLO
                     messages on all MANET interfaces
                     using an OTHER_NEIGHB TLV,
                     Value = LOST
```

Figure 10: HELLO Message Generation: Advertisement of Formerly Symmetric 1-Hop Neighbor on Any MANET Interface as Lost

Figure 11 illustrates the time, with duration I_HOLD_TIME, during which a formerly, but no longer, used local interface network address is excluded from being considered as a 2-hop neighbor network address (in order that a router is not recorded as its own 2-hop neighbor during that period).

```
     I_HOLD_TIME:      |-----------------------------|

     Time:          ---*------------------------------------->
                       ^                             ^
                       |                             |
       Formerly used local interface                 |
       network address ceases to be                  |
       assigned to a local interface                 |
                                                     |
                               Time up to which this network
                               address is excluded from being
                               included in this router's 2-Hop Set
```

Figure 11: Local Interface Removed Network Address

Appendix F. Topology Pictures

This appendix illustrates various examples of physical topologies, as well as how these are logically recorded by NHDP from the point of view of the router A. This representation is a composite of information that would be contained within A's various Information Bases after NHDP has been running for sufficiently long time for the state to converge.

Note that the examples given in this appendix are NOT exhaustive, but are selected to be illustrative of NHDP neighborhood representations of physical MANET topologies.

The example topologies presented contain 3 physical routers A, B, and C. Each of these routers has one or two distinct interfaces, denoted "top" and "bottom". Each interface has one or two addresses, and symmetric connectivity between a pair of interfaces is illustrated by these being connected by a line.

In all examples, the topology is described as it is recorded by NHDP in router A.

F.1. Example 1: Standard Single Interface Topology

In Figure 12, each router has a single interface, each with a single IP address. This is the simplest possible network, and the resulting representation is given to the right in Figure 12.

```
          ___________ ___________
         |           |           |
        {1}         {2}         {3}
         |           |           |          {1}--------{2}--------{3}
      +--'--+     +--'--+     +--'--+
      |  A  |     |  B  |     |  C  |
      +-----+     +-----+     +-----+
```

Figure 12: Standard Single Interface Topology (Left), and Corresponding NHDP Representation (Right)

The Local Information Set in A contains a single Local Interface Tuple that has an I_local_iface_addr_list of {1}. This value is denoted with a {1} on the leftmost part of the resulting representation.

The Interface Information Base has only one Link Set, which represents the "top" interface of A, or {1}. This Link Set's only Link Tuple has an L_neighbor_iface_addr_list containing {2}; this corresponds to the dashed line from {1} to {2} to the right in Figure 12. The 2-Hop Set contains a single 2-Hop Tuple, with N2_neighbor_iface_addr_list being {2} and N2_2hop_addr being {3}; this corresponds to the dashed line from {2} to {3} to the right in Figure 12.

The descriptions of the following examples in this appendix will be derived similarly, and use the same notational conventions.

F.2. Example 2: Dual Addressed Interface on 1-Hop Neighbor

In Figure 13, the network is identical to that in Example 1, except that the middle router, B, has two IP addresses on its single interface.

```
          ___________ ___________
         |           |           |
        {1}        {2,4}        {3}
         |           |           |          {1}-------{2,4}-------{3}
      +--'--+     +--'--+     +--'--+
      |  A  |     |  B  |     |  C  |
      +-----+     +-----+     +-----+
```

Figure 13: Single Interfaces, with 1-Hop Neighbor B Having Two Addresses (Left), and Corresponding NHDP Representation (Right)

The content of the Interface Information Base is in this case identical to Example 1, except that L_neighbor_iface_addr_list is {2,4} and N2_neighbor_iface_addr_list is {2,4}.

F.3. Example 3: Dual Addressed Interface on 2-Hop Neighbor

In Figure 14, the network is identical to that in Example 1, except that the rightmost router, C, has two IP addresses on its interface.

```
          __________ __________
         |          |          |
        {1}        {2}       {3,4}                      +----{3}
         |          |          |         {1}--------{2}---+
      +--'--+    +--'--+    +--'--+                      +----{4}
      |  A  |    |  B  |    |  C  |
      +-----+    +-----+    +-----+
```

Figure 14: Single Interfaces, with 2-Hop Neighbor C Having Two Addresses (Left), and Corresponding NHDP Representation (Right)

The content of the Interface Information Base is in this case identical to than in Example 1, except that the 2-Hop Set contains an extra 2-Hop Tuple with N2_neighbor_iface_addr_list being {2} and N2_2hop_addr being {4}. These two 2-Hop Tuples are illustrated by the two lines from {2} to {3} and (2) to {4}, respectively.

F.4. Example 4: Dual Addressed Interfaces

In Figure 15, the network is identical to that in Example 1, except that all routers have two IP addresses on their interface. The Local Information Base in router A is the same as in Example 1, except that I_local_iface_addr_list is {1,5}.

```
          __________ __________
         |          |          |
       {1,5}      {2,6}      {3,4}                      +----{3}
         |          |          |        {1,5}------{2,6}--+
      +--'--+    +--'--+    +--'--+                      +----{4}
      |  A  |    |  B  |    |  C  |
      +-----+    +-----+    +-----+
```

Figure 15: Single interfaces, all routers having two addresses (left), and corresponding NHDP representation (right)

The content of the Interface Information Base is in this case a combination of the Interface Information Bases from Examples 1, 2, and 3.

F.5. Example 5: Dual Interface on 2-Hop Neighbor

In Figure 16, all routers have a single IP address on each interface, and with the 2-hop neighbor having two interfaces.

```
         __________ __________
        |          |          |
       {1}        {2}        {3}                              +----{3}
        |          |          |              {1}--------{2}---+
     +--'--+    +--'--+    +-----+                            +----{4}
     |  A  |    |  B  |    |  C  |
     +-----+    +-----+    +-----+
                              |
                             {4}
```

Figure 16: Single Addresses, with 2-Hop Neighbor C Having Two Interfaces (Left), and Corresponding NHDP Representation (Right)

The Interface Information Base is identical to that in Example 3; NHDP does not distinguish topologically between this example and Example 3.

F.6. Example 6: Dual interface on 1-Hop Neighbor

In Figure 17, all routers have a single IP address on each interface, and with the 1-hop neighbor having two interfaces.

```
         __________
        |          |
       {1}        {2}                                  +-----+
        |          |                         {1}-------| {2} |------{4}
     +--'--+    +--'--+    +-----+                     | {5} |
     |  A  |    |  B  |    |  C  |                     +-----+
     +-----+    +-----+    +-----+
                   |          |
                  {5}        {4}
                   |__________|
```

Figure 17: Single Addresses, with 1-Hop Neighbor B Having Two Interfaces (Left), and Corresponding NHDP Representation (Right)

The Local Information Base is identical to that in Example 1.

The Interface Information Base has only one Link Set containing one Link Tuple with L_neighbor_iface_addr_list being {2}. The 2-Hop Set contains a single 2-Hop Tuple, with N2_neighbor_iface_addr_list being {2} and N2_2hop_addr being {4}.

The Neighbor Information Base contains a Neighbor Set containing a single Neighbor Tuple, which represents router B, with N_neighbor_addr_list being {2,5}. This entry is represented on the right of Figure 17 by boxing {2} with {5}.

Note that router A does not have information indicating which of router B's interfaces is connected to router C. However, router A knows that the address {4} (and thus router C) is reachable by using {2} as next hop.

F.7. Example 7: Dual Interface on 1-Hop and 2-Hop Neighbors

In Figure 18, all routers have a single IP address on each interface, and both the 1-hop and 2-hop neighbors have two interfaces. Furthermore, there are now two physical links between routers B and C, over distinct interface pairs.

```
         __________ __________
        |          |          |
       {1}        {2}        {3}                      +-----+   +----{3}
        |          |          |             {1}-------| {2} |---+
     +--'--+    +--'--+    +-----+                    | {5} |   +----{4}
     |  A  |    |  B  |    |  C  |                    +-----+
     +-----+    +-----+    +-----+
                   |          |
                  {5}        {4}
                   |__________|
```

Figure 18: Single Addresses, with 1-Hop and 2-Hop Neighbors B and C Having Two Interfaces (Left), and Corresponding NHDP Representation (Right)

The Local Information Base is identical to that in Example 1.

The Link Set is identical to that in Example 6, and the 2-Hop Set contains, as in Example 5 (and similarly to Examples 3 and 4), two 2-Hop Tuples representing the two links between routers B and C.

Note that router A does not have information describing which of router B's interfaces is connected to which interfaces of router C, or even that the interfaces with addresses {3} and {4} are interfaces of the same router. However, router A knows that the addresses {3} and {4} (and thus router C) are reachable using {2} as next hop.

F.8. Example 8: Dual Interface Locally and on 1-Hop Neighbor

In Figure 19, all routers have a single IP address on each interface, and both A and its the 1-hop neighbor B have two interfaces. Furthermore, there are now two physical links between routers A and B, over distinct interface pairs.

```
         __________ __________
        |          |          |                        +-----+
       {1}        {2}        {3}              {1}-------| {2} |--------{3}
        |          |          |                        | {5} |
     +--'--+    +--'--+    +-----+                     +-----+
     |  A  |    |  B  |    |  C  |
     +-----+    +-----+    +-----+                     +-----+
        |          |                                   | {2} |
       {6}        {5}                         {6}-------| {5} |--------{3}
        |__________|                                   +-----+
```

Figure 19: Single Addresses, with Both A and 1-Hop Neighbor B Having Two Interfaces (Left), and Corresponding NHDP Representation (Right)

The Local Information Set contains two Local Interface Tuples, with I_local_iface_addr_list of {1} and {6}, respectively.

Each Interface Information Base's Link Set contains one Link Tuple, representing the links between {1} and {2}, and between {6} and {5}, respectively. The 2-Hop Set for interface {1} contains a single 2-Hop Tuple, with N2_neighbor_iface_addr_list being {2} and N2_2hop_addr being {3}. The 2-Hop Set for interface {6} contains a single 2-Hop Tuple, with N2_neighbor_iface_addr_list being {5} and N2_2hop_addr being {3}.

The Neighbor Information Base contains a Neighbor Set containing a single Neighbor Tuple, which represents router B, with N_neighbor_addr_list being {2,5}. This entry is denoted by boxing {2} with {5}.

F.9. Example 9: Dual Interface on All Routers

In Figure 20, all routers have a single IP address on each interface, and all routers have two interfaces. Furthermore, there are now two physical links between A and B, over distinct interface pairs, and two physical links between B and C, also over distinct interface pairs.

```
         __________ __________
        |          |          |                       +-----+   +----{3}
       {1}        {2}        {3}            {1}-------| {2} |---+
        |          |          |                       | {5} |   +----{4}
     +--'--+    +--'--+    +-----+                    +-----+
     |  A  |    |  B  |    |  C  |
     +-----+    +-----+    +-----+                    +-----+
        |          |          |                       | {2} |   +----{3}
       {6}        {5}        {4}            {6}-------| {5} |---+
        |__________|__________|                       +-----+   +----{4}
```

Figure 20: Single Addresses, with All Routers Having Two Interfaces (Left), and Corresponding NHDP Representation (Right)

The Local Information Set is identical to that in Example 8. The Interface Information Base for each interface in A is also identical to that in Example 8, except that an additional 2-Hop Tuple is present in each 2-Hop Set, each representing the link between router B and the interface of router C with address {4}.

As in Example 7, router A does not have information describing which of router B's interfaces is connected to which interface of C, or even that the interfaces with addresses {3} and {4} are interfaces of the same router. However, router A knows that the addresses {3} and {4} (and router C) are reachable by using {2} or {5} (depending on via which of its local interfaces) as next hop.

F.10. Example 10: Dual Addressed Dual Interfaces on All Routers

In Figure 21, all routers have two IP addresses on each interface, and all routers have two interfaces. Furthermore, there are now two physical links between A and B, over distinct interface pairs, and two physical links between B and C, also over distinct interface pairs.

```
                                                                 +--{9}
         __________ __________                            +------|
        |          |          |                 +-----+   |      +--{10}
      {1,2}      {5,6}      {9,10}       {1,2}--|{5,6}|---+
        |          |          |                 |{7,8}|   |      +--{11}
     +--'--+    +--'--+    +-----+              +-----+   +------|
     |  A  |    |  B  |    |  C  |                               +--{12}
     +-----+    +-----+    +-----+
        |          |          |                                  +--{9}
        |          |          |                 +-----+   +------|
        |          |          |                 |{5,6}|   |      +--{10}
      {3,4}      {7,8}     {11,12}       {3,4}--|{7,8}|---+
        |__________|__________|                 +-----+   |      +--{11}
                                                          +------|
                                                                 +--{12}
```

Figure 21: Dual Addresses, with All Routers Having Two Interfaces (Left) and Corresponding NHDP Representation (Right)

This example is the combination of all the preceding examples in this appendix. The Local Information Set in A contains Local Information Tuples for each of its interfaces, and each Interface Information Base contains in its Link Set a representation of links {1,2}-{5,6} or {3,4}-{7,8}, respectively. The Neighbor Set (in the Neighbor Information Base) records the existence of router B and all of its addresses on all its interfaces, i.e., {5,6,7,8}.

As in Example 9, each interface address of router C is represented in the 2-Hop Set of each Interface Information Base as a link from router B to each of these addresses. Router A does not have information describing which of router B's interfaces is connected to which interface of C, nor that the addresses {9}, {10}, {11}, and {12} are addresses of the same router (or that some of these, such as {9} and {10}, are addresses on the same interface of the router).

F.11. Example 11: Single Addressed Dual Interface Locally

In Figure 22, all routers have a single interface, except for router A which has two. Each of A's two interfaces has a link with the single interface of router B. All interfaces have a single address.

```
      __________ __________
     |     _____|          |
    {1}   |    {2}        {3}              {1}--------{2}---------{3}
     |    |     |          |
  +--'--+ |  +--'--+    +-----+
  |  A  | |  |  B  |    |  C  |
  +-----+ |  +-----+    +-----+
     |    |
    {6}   |                               {6}--------{2}---------{3}
     |____|
```

Figure 22: Single Addresses, with A Having Two Interfaces, Both Linked to the Single Interface of B (Left), and Corresponding NHDP Representation (Right)

This is similar to Example 8, except that the single address {2} also replaces the address {5}. In particular, both Link Tuples (one in each Link Set, each in its corresponding Interface Information Base) have L_neighbor_iface_addr_list being {2}, the Neighbor Set (in the Neighbor Information Base) contains a single Neighbor Tuple with N_neighbor_addr_list being {2}, and both 2-Hop Tuples (one in each 2-Hop Set, each in its corresponding Interface Information Base) have N2_neighbor_iface_addr_list being {2} and N2_2hop_addr being {3}.

Authors' Addresses

Thomas Heide Clausen
LIX, Ecole Polytechnique

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

Christopher Dearlove
BAE Systems ATC

Phone: +44 1245 242194
EMail: chris.dearlove@baesystems.com
URI: http://www.baesystems.com/

Justin W. Dean
Naval Research Laboratory

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