Mobile Ad hoc Networking (MANET)
Internet-Draft
Intended status: Standards Track
Expires: March 29, 2010

T. Clausen
LIX, Ecole Polytechnique
C. Dearlove
BAE Systems ATC
P. Jacquet
Project Hipercom, INRIA
The OLSRv2 Design Team
MANET Working Group

September 25, 2009

The Optimized Link State Routing Protocol version 2
draft-ietf-manet-olsrv2-10

Status of This Memo

Copyright Notice

Abstract

This document describes version 2 of the Optimized Link State Routing (OLSRv2) protocol for Mobile Ad hoc NETworks (MANETs).

Table of Contents

1. Introduction

The Optimized Link State Routing protocol version 2 (OLSRv2) is an update to OLSRv1 as published in [RFC3626]. Compared to [RFC3626], OLSRv2 retains the same basic mechanisms and algorithms, while using a more flexible and efficient signaling framework, and includes some simplification of the messages being exchanged.

OLSRv2 is developed for mobile ad hoc networks. It operates as a table driven, proactive protocol, i.e. it exchanges topology information with other routers in the network regularly. OLSRv2 is an optimization of the classical link state routing protocol. Its key concept is that of MultiPoint Relays (MPRs). Each router selects a set of its neighbor routers (which "cover" all of its symmetrically connected 2-hop neighbor routers) as MPRs. MPRs are then used to achieve both flooding reduction and topology reduction.

Flooding reduction is achieved by control traffic being flooded through the network using hop by hop forwarding, but with a router only needing to forward control traffic which is first received directly from one of the routers which have selected it as an MPR (its "MPR selectors"). This mechanism, denoted "MPR flooding", provides an efficient mechanism for information distribution within the MANET by reducing the number of transmissions required.

Topology redction is achieved by a mechanism where the routers selected as MPRs have a special responsibility when declaring link state information in the network. A sufficient requirement for OLSRv2 to provide shortest (lowest hop count) routes to all destinations is that routers declare link state information for their MPR selectors, if any. Routers which are not selected as MPRs need not send any link state information. Additional available link state information may be transmitted, e.g. for redundancy. Thus the use of MPRs allows reduction of the number and the size of link state messages, and in the amount of link state information maintained in each router. Based on this reduced link state information, MPRs are used as intermediate routers in multi-hop routes.

A router selects MPRs from among its one hop neighbors connected by "symmetric", i.e. bidirectional, links. Therefore, selecting routes through MPRs avoids the problems associated with data packet transfer over unidirectional links (such as the problem of not getting link layer acknowledgments at each hop, for link layers employing this technique).

OLSRv2 uses and extends [NHDP] and uses [RFC5444], [RFC5497] and, optionally, [RFC5148]. These other protocols and specifications were all originally created as part of OLSRv2, but have been specified

separately for wider use.

OLSRv2 makes no assumptions about the underlying link layer. OLSRv2, through its use of [NHDP], may use link layer information and notifications when available and applicable.

OLSRv2, as OLSRv1, inherits its concept of forwarding and relaying from HIPERLAN (a MAC layer protocol) which is standardized by ETSI [HIPERLAN], [HIPERLAN2].

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", and "TLV", are to be interpreted as described there.

All terms introduced in [NHDP], including "interface", "MANET interface", "address", "symmetric link", "symmetric 1-hop neighbor", "symmetric 2-hop neighbor", "constant", "interface parameter", and "router parameter", are to be interpreted as described there.

Additionally, this document uses the following terminology:

Router - A MANET router which implements the protocol specified in this document.

OLSRv2 interface - A MANET interface running this protocol.

Routable address - An address which may be used as the destination of a packet. A router MUST be able to distinguish a routable address from a non-routable address by direct inpsection of the address, based on global scope address allocations by IANA and/or administrative configuration. Broadcast, multicast and anycast addresses, and addresses which are limited in scope to less than the entire MANET, MUST NOT be considered as routable addresses.

Originator address - An address which is unique (within the MANET) to a router. A router MUST select an originator address; it MAY choose one of its interface addresses as its originator address. If it selects a routable address then this MUST be one which this router will accept as destination. An originator address MUST NOT have a prefix length.

Message originator address - The originator address of the router which created a message, as deduced from that message by its recipient. The message originator address will usually be included in the message as its <msg-orig-addr> element as defined in [RFC5444]. However an exceptional case in a HELLO message is also allowed by this specification when a router only uses a single address. All messages used in this specification, including HELLO messages defined in [NHDP], MUST have a message originator address.

Willingness - A numerical value between WILL_NEVER and WILL_ALWAYS (both inclusive), which represents the router's willingness to be selected as an MPR.

Willing symmetric 1-hop neighbor - A symmetric 1-hop neighbor of this router which has willingness not equal to WILL_NEVER.

Symmetric 1-hop neighbor through OLSRv2 interface I - A symmetric 1-hop neighbor of the router via a symmetric link using OLSRv2 interface I of the router.

Symmetric strict 2-hop neighbor - A router, X, is a symmetric strict 2-hop neighbor of a router Y, if router X is a symmetric 2-hop neighbor of router Y and if router X is not also a willing symmetric 1-hop neighbor of router Y.

Symmetric strict 2-hop neighbor through OLSRv2 interface I - A symmetric strict 2-hop neighbor of the router which is a symmetric 1-hop neighbor of a willing symmetric 1-hop neighbor through OLSRv2 interface I. The router MAY elect to consider only information received over OLSRv2 interface I in making this determination.

Symmetric strict 2-hop neighborhood - The symmetric strict 2-hop neighborhood of a router X is the set of symmetric strict 2-hop neighbors of router X.

Multipoint relay (MPR) - A router, X, is an MPR for a router, Y, if router Y has selected router X to "re-transmit" all the broadcast messages that it receives from router X, provided that the message is not a duplicate, and that the hop limit field of the message is greater than one.

MPR selector - A router, Y, is an MPR selector of router X if router Y has selected router X as MPR.

MPR flooding - The optimized MANET-wide information distribution mechanism, employed by this protocol, in which a message is relayed by only a reduced subset of the routers in the network. MPR flooding is the mechanism by which flooding reduction is achieved.

This document employs the same notational conventions as in [RFC5444] and [NHDP].

3. Applicability Statement

This protocol:

- Is a proactive routing protocol for mobile ad hoc networks (MANETs) [RFC2501].

- 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 OLSRv2 interfaces. The set of a router's interfaces may change over time. Each OLSRv2 interface may have one or more addresses (which may have prefix lengths), and these may also be dynamically changing.

- Enables hop-by-hop routing, i.e., each router can use its local information provided by this protocol to route packets.

- Continuously maintains routes to all destinations in the network, i.e., routes are instantly available and data traffic is subject to no delays due to route discovery. Consequently, no data traffic buffering is required.

- Supports routers which have non-OLSRv2 interfaces which may be local to a router or which can serve as gateways towards other networks.

- Is optimized for large and dense networks: the larger and more dense a network, the more optimization can be achieved by using MPRs, compared to the classic link state algorithm.

- Uses the message format specified in [RFC5444]. This includes the definition of a TC Message Type, used for MANET wide signaling of network topology information.

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

o Uses [NHDP] for discovering each router's 1-hop and symmetric 2-hop neighbors, and extends [NHDP] by addition of MPR and willingness information.

o 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, independently:

o Identify all destinations in the network.

o Identify a sufficient subset of links in the network, in order that shortest paths can be calculated to all available destinations.

o Provide a Routing Set, containing these shortest paths from this router to all destinations (routable addresses and local links).

4.1. Overview

These objectives are achieved, for each router, by:

o Using [NHDP] to identify symmetric 1-hop neighbors and symmetric 2-hop neighbors.

o Independently selecting MPRs from among its symmetric 1-hop neighbors such that all symmetric 2-hop neighbors are reachable via at least one symmetric 1-hop neighbor. An analysis and examples of MPR selection algorithms is given in [MPR], a suggested algorithm is included in this specification. Note that it is not necessary for routers to use the same algorithm in order to interoperate in the same MANET.

o Signaling its MPR selection by extending [NHDP] to include this information in outgoing HELLO messages, by the addition of MPR Address Block TLV(s) associated with appropriate addresses.

o Extracting its MPR selectors from received HELLO messages, using the included MPR Address Block TLV(s).

o Reporting its willingness to be an MPR in HELLO messages, by the addition on an MPR_WILLING Message TLV. The router's willingness to be an MPR indicates how willing it is to participate in MPR flooding and to be an intermediate node for routing. A node can absolutely decline to perform either role.

o Periodically signaling links between MPR selectors and itself throughout the MANET, by using TC (Topology Control) messages, defined in this specification.

o Diffusing TC messages by using flooding reduction mechanism, denoted "MPR flooding": only the MPRs of a router will retransmit messages received from (i.e., originated or last relayed by) that router.

Note that the indicated extensions to [NHDP] are of forms permitted by that specification.

This specification defines, in turn:

o Parameters and constants used by this protocol, in addition to those specified in [NHDP]. Parameters used by this protocol may, where appropriate, be specific to a given OLSRv2 interface, or to a router. This protocol allows all parameters to be changed dynamically, and to be set independently for each router or each OLSRv2 interface, as appropriate.

o Extensions to the Information Bases specified in [NHDP].

o Two new Information Bases: the Topology Information Base and the Received Message Information Base.

o A requirement for each router to have an originator address to be included in the HELLO messages of [NHDP].

o A Message TLV, to be included in the HELLO messages of [NHDP], allowing a router to indicate its willingness to be an MPR.

o An Address Block TLV, to be included in the HELLO messages of [NHDP], allowing a router to signal its MPR selection.

o The MPR flooding mechanism, including the inclusion of message originator address and sequence number to manage duplicate messages.

o TC messages, which are used for MANET wide signaling (using MPR flooding) of selected topology (link state) information.

o The specification of new Message TLVs and Address Block TLVs which are used in TC messages.

o The generation of TC messages from the appropriate information in the Information Bases.

- The updating of the Topology Information Base according to received TC messages.

- The response to other events, such as the expiration of information in the Information Bases.

This protocol inherits the stability of a link state algorithm and has the advantage of having routes immediately available when needed, due to its proactive nature.

This protocol only interacts with IP through routing table management, and the use of the sending IP address for IP datagrams containing OLSRv2 packets.

4.2. Routers and Interfaces

In order for a router to participate in a MANET, it MUST have at least one, and possibly more, OLSRv2 interfaces. Each OLSRv2 interface:

- Is configured with one or more addresses, as specified in [NHDP]. These addresses MUST each be unique within the MANET and MUST include any address that will be used as the sending address of any IP packet sent on this OLSRv2 interface.

- Has a number of interface parameters, adding to those specified in [NHDP].

- Has an Interface Information Base, extending that specified in [NHDP].

- Generates and processes HELLO messages according to [NHDP], extended as specified in Section 9 and Section 10.

In addition to a set of OLSRv2 interfaces as described above, each router:

- May have one or more non-OLSRv2 interfaces and/or local attached networks which this router can accept packets destined for. All routable addresses of the router for which it is to accept packets as destination MUST be used as an (OLSRv2 or non-OLSRv2) interface address or of a local attached network.

- Has a number of router parameters, adding to those specified in [NHDP].

- Has a Local Information Base, extending that specified in [NHDP], including selection of an originator address and recording any

locally attached networks.

- o Has a Neighbor Information Base, extending that specified in [NHDP] to record MPR selection and advertisement information.
- o Has a Topology Information Base, recording information received in TC messages and derived therefrom.
- o Has a Received Message Information Base, recording information about received messages to ensure that each TC message is only processed once, and forwarded at most once on each OLSRv2 interface, by a router.
- o Generates and processes TC messages.

4.3. Information Base Overview

Each router maintains the Information Bases described in the following sections. These are used for describing the protocol in this document. An implementation of this protocol MAY maintain this information in the indicated form, or in any other organization which offers access to this information. In particular, note that it is not necessary to remove Tuples from Sets at the exact time indicated, only to behave as if the Tuples were removed at that time.

4.3.1. Local Information Base

The Local Information Base is specified in [NHDP] and contains a router's local configuration. It is extended in this specification to also record an originator address and to include a router's:

- o Originator Set, containing addresses that were recently used as this router's originator address, and is used to enable a router to recognize and discard control traffic which was originated by the router itself.
- o Local Attached Network Set, containing addresses of networks to which this router can act as a gateway, and advertises in its TC messages.

4.3.2. Interface Information Bases

The Interface Information Bases, one for each OLSRv2 interface, are specified in [NHDP]. In addition to the uses in [NHDP], information recorded in the Interface Information Bases is used for completing the Routing Set.

4.3.3. Neighbor Information Base

The Neighbor Information Base is specified in [NHDP], and is extended to also record each neighbor's originator address, the willingness of each neighbor to be an MPR, as well as this router's MPR relationships with each neighbor (whether an MPR and/or an MPR selector of that neighbor) and whether that neighbor is to be advertised in TC messages.

A router selects some of its symmetric 1-hop neighbors as MPRs (see Section 14). That selection is recorded in the Neighbor Set. This selection is then reported in the router's HELLO messages, extending the specification in [NHDP], by using an MPR Address Block TLV. In making that selection a router MUST consider its 1-hop neighbors' willingness to be an MPR, which (unless having default value) is reported using an Address Block TLV in HELLO messages and recorded in the receiving router's Neighbor Set.

A router also records in the Neighbor Set which symmetric 1-hop neighbors have selected it as an MPR (i.e. its MPR selectors). This is determined from the MPR TLVs in received HELLO messages. It also records which symmetric 1-hop neighbors that it is to advertise connectivity to in its TC messages; this MUST include all of its MPR selectors.

The Neighbor Set finally records each 1-hop neighbor's originator address, as included in its HELLO messages in an extension to [NHDP]. This, and other information in the Neighbor Set, including each 1-hop neighbor's routable addresses, is used in advertising the selected symmetric 1-hop neighbors in TC messages.

4.3.4. Topology Information Base

The purpose of the Topology Information Base is to record information used, in addition to that in the Local Information Base, the Interface Information Bases and the Neighbor Information Base, to construct the Routing Set (which is also included in the Topology Information Base).

This specification describes the calculation of the Routing Set based on a Topology Graph constructed in two phases. First, a "backbone" graph representing the routers in the MANET, and the connectivity between them, is constructed from the Local Information Base, the Neighbor Information Base and the Router Topology Set in the Topology Information Base. Second, this graph is "decorated" with additional destination addresses using the Local Information Base, and the Routable Address Topology Set and the Attached Network Set in the Topology Information Base.

The Topology Graph does not need to be recorded in the Topology Information Base, it can either be constructed as required when the Routing Set is to be changed, or need not be explicitly constructed (as illustrated in Appendix B. An implementation MAY construct and retain the Topology Graph if preferred.

The Topology Information Base in each router contains:

- An Advertising Remote Router Set, recording each other router from which TC messages have been received. This is used in order to determine if a received TC messages contains fresh or outdated information; the TC message is ignored in the latter case.

- A Router Topology Set, recording links between routers in the MANET, as described by received TC messages.

- A Routable Address Topology Set, recording routable addresses in the MANET (available as packet destinations) and from which other router these addresses can be directly reached (i.e. in a single IP hop) as reported by received TC messages.

- An Attached Network Set, recording networks to which a remote router has advertised that it may act as a gateway. These networks may be reached in one or more IP hops.

- A Routing Set, recording routes from this router to all available destinations. The IP routing table is to be updated using this Routing Set. (A router MAY choose to use any or all destination addresses in the Routing Set to update the IP routing table, this selection is outside the scope of this protocol.)

4.3.5. Received Message Information Base

The Received Message Information Base in each router contains:

- A Received Set for each OLSRv2 interface, describing TC messages received by this router on that OLSRv2 interface.

- A Processed Set, describing TC messages processed by this router.

- A Forwarded Set, describing TC messages forwarded by this router.

The Received Message Information Base serves the MPR flooding mechanism by ensuring that received messages are forwarded at most once by a router, and also ensures that received messages are processed exactly once by a router.

4.4. Signaling Overview

This protocol generates and processes HELLO messages according to [NHDP], extended according to Section 9 and Section 10 of this specification to include an originator address and MPR selection information.

This protocol specifies a single message type, the TC message.

This protocol is tolerant of unreliable transmissions of TC messages; each router sends TC messages periodically, and can therefore sustain a reasonable loss of some such messages. Such losses may occur more frequently in wireless networks due to collisions or other transmission problems. This protocol MAY use "jitter", randomized adjustments to message transmission times, to reduce the incidence of collisions as specified in [RFC5148].

This protocol is tolerant of out of sequence delivery of TC messages due to in transit message reordering (possibly due to message alternative routing by flooding and message loss). Each router maintains an Advertised Neighbor Sequence Number (ANSN) which is incremented when its recorded neighbor information that is to be included in its TC messages changes. This ANSN is included in the router's TC messages. The recipient of a TC message can used this included ANSN to identify which of the information it has received is most recent, even if messages have been re-ordered while in transit. Only the most recent information received is used, older information received later is discarded.

TC messages may be "complete" or "incomplete". A complete TC message advertises all of the originating router's MPR selectors, it may also advertise other symmetric 1-hop neighbors. Complete TC messages are generated periodically (and also, optionally, in response to neighborhood changes). Incomplete TC messages may be used to report additions to advertised information without repeating unchanged information.

TC messages, and HELLO messages as extended by this specification, include an originator address for the router that created the message. A TC message reports both the originator addresses and routable addresses of its advertised neighbors, distinguishing the two using a TLV for this purpose (an address may be both).

TC messages also report the originator's locally attached networks.

TC messages are MPR flooded throughout the MANET. A router retransmits a TC message only if it is received from (i.e., originated from or was last relayed by) one of that router's MPR

selectors.

Some TC messages may be MPR flooded over only part of the network, e.g., allowing a router to ensure that nearer routers are kept more up to date than distant routers, such as is used in Fisheye State Routing [FSR] and Fuzzy Sighted Link State routing [FSLS]. This is enabled using [RFC5497].

4.5. Routing Set

The purpose of the Routing Set is to determine and record routes (local interface address and next hop interface address) to all possible routable addresses and of all destinations that are local, i.e. within one hop, to the router (whether using routable addresses or not). Only symmetric links are used in such routes.

It is intended that the Routing Set can be used for packet routing, by using its contents to update IP's routing tables. That update, and whether any Routing Tuples are not used in IP's routing table, is outside the scope of this specification.

The signaling in this specification has been designed so that a "backbone" Topology Graph of routers, each identified by its originator address, with at most one direct connection between any pair of routers, can be constructed (from the Neighbor Set and the Router Topology Set) using a suitable minimum path length algorithm, and then this Topology Graph can have other addresses (routable, or of symmetric 1-hop neighbors) added to it (using the Interface Information Base, the Routable Address Topology Set and the Attached Network Set).

5. Protocol Parameters and Constants

The parameters and constants used in this specification are those defined in [NHDP] plus those defined in this section. The separation in [NHDP] into interface parameters, router parameters and constants is also used in this specification, however all but one (RX_HOLD_TIME) of the parameters added by this protocol are router parameters. Parameters may be categorized as follows:

- Local history times

- Message intervals

- Advertised information validity times

- Received message validity times

- Jitter times
- Hop limits
- Willingness

In addition, constants for particular cases of a router's willingness to be an MPR are defined. These parameters and constants are detailed in the following sections. As for the parameters in [NHDP], parameters defined in this document may be changed dynamically by a router, and need not be the same on different routers, even in the same MANET, or, for interface parameters, on different interfaces of the same router.

5.1. Protocol and Port Numbers

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

TC messages and HELLO messages [NHDP] SHOULD, in a given deployment of this protocol, both be using the same of either of IP or UDP, in order that it is possible to combine messages of both protocols into the same [RFC5444] packet for transmission.

5.2. Multicast Address

This protocol specifies TC 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. Local History Times

The following router parameter manages the time for which local information is retained:

O_HOLD_TIME - is used to define the time for which a recently used and replaced originator address is used to recognize the router's own messages.

The following constraint applies to this parameter:

- O_HOLD_TIME >= 0

5.4. Message Intervals

The following router parameters regulate TC message transmissions by a router. TC messages are usually sent periodically, but MAY also be sent in response to changes in the router's Neighbor Set and/or Local Attached Network Set. With a larger value of the parameter TC_INTERVAL, and a smaller value of the parameter TC_MIN_INTERVAL, TC messages may more often be transmitted in response to changes in a highly dynamic network. However because a router has no knowledge of, for example, routers remote to it (i.e. beyond 2 hops away) joining the network, TC messages MUST NOT be sent purely responsively.

TC_INTERVAL - is the maximum time between the transmission of two successive TC messages by this router. When no TC messages are sent in response to local network changes (by design, or because the local network is not changing) then TC messages SHOULD be sent at a regular interval TC_INTERVAL, possibly modified by jitter as specified in [RFC5148].

TC_MIN_INTERVAL - is the minimum interval between transmission of two successive TC messages by this router. (This minimum interval MAY be modified by jitter, as specified in [RFC5148].)

The following constraints apply to these parameters:

- TC_INTERVAL > 0
- TC_MIN_INTERVAL >= 0
- TC_INTERVAL >= TC_MIN_INTERVAL
- If INTERVAL_TIME TLVs as defined in [RFC5497] are included in TC messages, then TC_INTERVAL MUST be representable as described in [RFC5497].

5.5. Advertised Information Validity Times

The following router parameters manage the validity time of information advertised in TC messages:

T_HOLD_TIME - is used to define the minimum Value in the VALIDITY_TIME TLV included in all TC messages sent by this router. If a single value of parameter TC_HOP_LIMIT (see Section 5.8) is used then this will be the only Value in that TLV.

A_HOLD_TIME - is the period during which TC messages are sent after they no longer have any advertised information to report, but are sent in order to accelerate outdated information removal by other routers.

The following constraints apply to these parameters:

- T_HOLD_TIME > 0
- A_HOLD_TIME >= 0
- T_HOLD_TIME >= TC_INTERVAL
- If TC messages can be lost, then both T_HOLD_TIME and A_HOLD_TIME SHOULD be significantly greater than TC_INTERVAL; a value >= 3 x TC_INTERVAL is RECOMMENDED.
- T_HOLD_TIME MUST be representable as described in [RFC5497].

5.6. Received Message Validity Times

The following parameters manage the validity time of recorded received message information:

RX_HOLD_TIME - is an interface parameter, and is the period after receipt of a message by the appropriate OLSRv2 interface of this router for which that information is recorded, in order that the message is recognized as having been previously received on this OLSRv2 interface.

P_HOLD_TIME - is a router parameter, and is the period after receipt of a message which is processed by this router for which that information is recorded, in order that the message is not processed again if received again.

F_HOLD_TIME - is a router parameter, and is the period after receipt of a message which is forwarded by this router for which that information is recorded, in order that the message is not forwarded again if received again.

The following constraints apply to these parameters:

- RX_HOLD_TIME > 0
- P_HOLD_TIME > 0
- F_HOLD_TIME > 0

o All of these parameters SHOULD be greater than the maximum difference in time that a message may take to traverse the MANET, taking into account any message forwarding jitter as well as propagation, queuing, and processing delays.

5.7. Jitter

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

TP_MAXJITTER - represents the value of MAXJITTER used in [RFC5148] for periodically generated TC messages sent by this router.

TT_MAXJITTER - represents the value of MAXJITTER used in [RFC5148] for externally triggered TC messages sent by this router.

F_MAXJITTER - represents the default value of MAXJITTER used in [RFC5148] for messages forwarded by this router. However before using F_MAXJITTER a router MAY attempt to deduce a more appropriate value of MAXJITTER, for example based on any INTERVAL_TIME or VALIDITY_TIME TLVs contained in the message to be forwarded.

For constraints on these parameters see [RFC5148].

5.8. Hop Limit Parameter

The parameter TC_HOP_LIMIT is the hop limit set in each TC message. TC_HOP_LIMIT MAY be a single fixed value, or MAY be different in TC messages sent by the same router. However each other router, at any hop count distance, SHOULD see a regular pattern of TC messages, in order that meaningful Values of INTERVAL_TIME and VALIDITY_TIME TLVs at each hop count distance can be included as defined in [RFC5497]. Thus the pattern of TC_HOP_LIMIT SHOULD be defined to have this property. For example the repeating pattern (255 4 4) satisfies this property (having period TC_INTERVAL at hop counts up to 4, inclusive, and 3 x TC_INTERVAL at hop counts greater than 4), but the repeating pattern (255 255 4 4) does not satisfy this property because at hop counts greater than 4, message intervals are alternately TC_INTERVAL and 3 x TC_INTERVAL.

The following constraints apply to this parameter:

o The maximum value of TC_HOP_LIMIT >= the network diameter in hops, a value of 255 is RECOMMENDED. Note that if using a pattern of different values of TC_HOP_LIMIT as described above, then only the maximum value in the patttern is so constrained.

o All values of TC_HOP_LIMIT >= 2.

5.9. Willingness

Each router has a WILLINGNESS parameter, which MUST be in the range WILL_NEVER to WILL_ALWAYS, inclusive, and represents the router's willingness to be an MPR, and hence its willingness to forward messages and be an intermediate router on routes. If a router has WILLINGNESS = WILL_NEVER it does not perform these tasks. A MANET using this protocol with too many routers having WILLINGNESS = WILL_NEVER will not function; it MUST be ensured, by administrative or other means, that this does not happen.

Routers MAY have different WILLINGNESS values; however the three constants WILL_NEVER, WILL_DEFAULT and WILL_ALWAYS MUST have the values defined in Section 16. (Use of WILLINGNESS = WILL_DEFAULT allows a router to avoid including an MPR_WILLING TLV in its TC messages, use of WILLINGNESS = WILL_ALWAYS means that a router will always be selected as an MPR by all symmetric 1-hop neighbors.)

The following constraints apply to this parameter:

o WILLINGNESS >= WILL_NEVER

o WILLINGNESS <= WILL_ALWAYS

5.10. Parameter Change Constraints

This section presents guidelines, applicable if protocol parameters are changed dynamically.

O_HOLD_TIME

* If O_HOLD_TIME for a router changes, then O_time for all Originator Tuples MAY be changed.

TC_INTERVAL

* If the TC_INTERVAL for a router increases, then the next TC message generated by this router MUST be generated according to the previous, shorter, TC_INTERVAL. Additional subsequent TC messages MAY be generated according to the previous, shorter, TC_INTERVAL.

* If the TC_INTERVAL for a router decreases, then the following TC messages from this router MUST be generated according to the current, shorter, TC_INTERVAL.

RX_HOLD_TIME

* If RX_HOLD_TIME for an OLSRv2 interface changes, then RX_time for all Received Tuples for that OLSRv2 interface MAY be changed.

P_HOLD_TIME

* If P_HOLD_TIME changes, then P_time for all Processed Tuples MAY be changed.

F_HOLD_TIME

* If F_HOLD_TIME changes, then F_time for all Forwarded Tuples MAY be changed.

TP_MAXJITTER

* If TP_MAXJITTER changes, then the periodic TC message schedule on this router MAY be changed immediately.

TT_MAXJITTER

* If TT_MAXJITTER changes, then externally triggered TC messages on this router MAY be rescheduled.

F_MAXJITTER

* If F_MAXJITTER changes, then TC messages waiting to be forwarded with a delay based on this parameter MAY be rescheduled.

TC_HOP_LIMIT

* If TC_HOP_LIMIT changes, and the router uses multiple values after the change, then message intervals and validity times included in TC messages MUST be respected. The simplest way to do this is to start any new repeating pattern of TC_HOP_LIMIT values with its largest value.

6. Information Bases

The purpose of this protocol is to determine the Routing Set, which may be used to update IP's Routing Table, providing "next hop" routing information for IP packets. This specification includes the following Information Bases:

Local Information Base - as defined in [NHDP], extended by the inclusion of the router's originator address and the addition of an Originator Set, defined in Section 6.1.1, and a Local Attached Network Set, defined in Section 6.1.2.

Interface Information Bases - as defined in [NHDP], an Interface Information Base for each OLSRv2 interface.

Neighbor Information Base - as defined in [NHDP], extended by the addition of five elements to each Neighbor Tuple, and the inclusion of an Advertised Neighbor Sequence Number (ANSN), both as defined in Section 6.2.

Topology Information Base - this Information Base is specific to this protocol, and is defined in Section 6.3.

Received Message Information Base - this Information Base is specific to this protocol, and is defined in Section 6.4.

The ordering of sequence numbers, when considering which is the greater, is as defined in Section 17.

6.1. Local Information Base

The Local Information Base as defined in [NHDP] is extended by:

o Recording the router's originator address. Note that this MAY be equal to any address in any I_local_iface_addr_list in a Local Interface Tuple, but MUST NOT be equal to the AL_net_addr in a Local Attached Network Tuple.

o The addition of an Originator Set, defined in Section 6.1.1, and a Local Attached Network Set, defined in Section 6.1.2.

All routable addresses of the router for which it is to accept packets as destination MUST be included in the Local Interface Set or the Local Attached Network Set.

6.1.1. Originator Set

A router's Originator Set records addresses that were recently used as originator addresses by this router. If a router's originator address is immutable then this set is always empty and MAY be omitted. It consists of Originator Tuples:

(O_orig_addr, O_time)

where:

O_orig_addr is a recently used originator address, note that this does not include a prefix length;

O_time specifies the time at which this Tuple expires and MUST be removed.

6.1.2. Local Attached Network Set

A router's Local Attached Network Set records its local non-OLSRv2 interfaces via which it can act as gateways to other networks. The Local Attached Network Set is not modified by this protocol. This protocol MAY respond to changes to the Local Attached Network Set, which MUST reflect corresponding changes in the router's status. It consists of Local Attached Network Tuples:

(AL_net_addr, AL_dist)

where:

AL_net_addr is the network address of an attached network which can be reached via this router. This SHOULD be a routable address, and MUST NOT be an interface address, or the originator address, of this router.

AL_dist is the number of hops to the network with address AL_net_addr from this router.

Attached networks local to this router only (i.e. not reachable except via this router) SHOULD be treated as local non-MANET interfaces, and added to the Local Interface Set, as specified in [NHDP], rather than be added to the Local Attached Network Set.

Because an attached network is not specific to the router, and may be outside the MANET, an attached network MAY also be attached to other routers.

It is not the responsibility of this protocol to maintain routes from this router to networks recorded in the Local Attached Network Set.

Local Attached Neighbor Tuples are removed from the Local Attached Network Set only when the routers' local attached network configuration changes, i.e., they are not subject to timer-based expiration or changes due to received messages.

6.2. Neighbor Information Base

Each Neighbor Tuple in the Neighbor Set, defined in [NHDP], has these additional elements:

N_orig_addr is the neighbor's originator address, which may be unknown. Note that this originator address does not include a prefix length;

N_willingness is the neighbor's willingness to be selected as an MPR, in the range from WILL_NEVER to WILL_ALWAYS, both inclusive;

N_mpr is a boolean flag, describing if this neighbor is selected as an MPR by this router;

N_mpr_selector is a boolean flag, describing if this neighbor has selected this router as an MPR, i.e., is an MPR selector of this router.

N_advertised is a boolean flag, describing if this router has elected to advertise a link to this neighbor in its TC messages.

A Neighbor Tuple created (but not updated) by [NHDP] MUST set:

N_orig_addr := unknown;

N_willingness := WILL_NEVER;

N_mpr := false;

N_mpr_selector := false;

N_advertised := false.

The Neighbor Information Base also includes a variable, the Advertised Neighbor Sequence Number (ANSN), whose value is included in TC messages to indicate the freshness of the information transmitted. The ANSN is incremented whenever advertised information (the originator and routable addresses included in Neighbor Tuples with N_advertised = true, and local attached networks recorded in the Local Attached Network Set in the Local Information Base) changes.

6.3. Topology Information Base

The Topology Information Base stores information received in TC messages, in the Advertising Remote Router Set, the Router Topology Set, the Routable Address Topology Set and the Attached Network Set.

Additionally, a Routing Set is maintained, derived from the information recorded in the Local Information Base, the Interface Information Bases, the Neighbor Information Base and the rest of the Topology Information Base.

6.3.1. Advertising Remote Router Set

A router's Advertising Remote Router Set records information describing each remote router in the network that transmits TC messages, allowing outdated TC messages to be recognized and discarded. It consists of Advertising Remote Router Tuples:

(AR_orig_addr, AR_seq_number, AR_time)

where:

AR_orig_addr is the originator address of a received TC message, note that this does not include a prefix length;

AR_seq_number is the greatest ANSN in any TC message received which originated from the router with originator address AR_orig_addr (i.e., which contributed to the information contained in this Tuple);

AR_time is the time at which this Tuple expires and MUST be removed.

6.3.2. Router Topology Set

A router's Topology Set records topology information about the links between routers in the MANET, allowing a "backbone" graph of all routers to be constructed using a minimum distance algorithm. It consists of Router Topology Tuples:

(TR_from_orig_addr, TR_to_orig_addr, TR_seq_number, TR_time)

where:

TR_from_orig_addr is the originator address of a router which can reach the router with originator address TR_to_orig_addr in one hop, note that this does not include a prefix length;

TR_to_orig_addr is the originator address of a router which can be reached by the router with originator address TR_to_orig_addr in one hop, note that this does not include a prefix length;

TR_seq_number is the greatest ANSN in any TC message received which originated from the router with originator address TR_from_orig_addr (i.e., which contributed to the information contained in this Tuple);

TR_time specifies the time at which this Tuple expires and MUST be removed.

6.3.3. Routable Address Topology Set

A router's Routable Address Topology Set records topology information about the routable addresses within the MANET, and via which routers they may be reached. It consists of Routable Address Topology Tuples:

(TA_from_orig_addr, TA_dest_addr, TA_seq_number, TA_time)

where:

TA_from_orig_addr is the originator address of a router which can reach the router with routable address TA_dest_addr in one hop, note that this does not include a prefix length;

TA_dest_addr is a routable address of a router which can be reached by the router with originator address TA_from_orig_addr in one hop;

TA_seq_number is the greatest ANSN in any TC message received which originated from the router with originator address TA_from_orig_addr (i.e., which contributed to the information contained in this Tuple);

TA_time specifies the time at which this Tuple expires and MUST be removed.

6.3.4. Attached Network Set

A router's Attached Network Set records information about networks (which may be outside the MANET) attached to other routers and their routable addresses. It consists of Attached Network Tuples:

(AN_orig_addr, AN_net_addr, AN_dist, AN_seq_number, AN_time)

where:

AN_orig_addr is the originator address of a router which can act as gateway to the network with address AN_net_addr, note that this does not include a prefix length;

AN_net_addr is the network address of an attached network, which may be reached via the router with originator address AN_orig_addr;

AN_dist is the number of hops to the network with address AN_net_addr from the router with originator address AN_orig_addr;

AN_seq_number is the greatest ANSN in any TC message received which originated from the router with originator address AN_orig_addr (i.e., which contributed to the information contained in this Tuple);

AN_time specifies the time at which this Tuple expires and MUST be removed.

6.3.5. Routing Set

A router's Routing Set records the first hop along a selected path to each destination for which any such path is known. It consists of Routing Tuples:

(R_dest_addr, R_next_iface_addr, R_local_iface_addr, R_dist)

where:

R_dest_addr is the address of the destination, either the address of an interface of a destination router, or the network address of an attached network;

R_next_iface_addr is the address of the "next hop" on the selected path to the destination;

R_local_iface_addr is the address of the local OLSRv2 interface over which a packet MUST be sent to reach the destination by the selected path.

R_dist is the number of hops on the selected path to the destination;

The Routing Set for a router is derived from the contents of other protocol Sets of the router (the Link Sets, the Neighbor Set, the Router Topology Set, the Routable Address Topology Set, the Attached Network Set, and OPTIONALLY the Two Hop Sets). The Routing Set is updated (Routing Tuples added or removed, or the complete Routing Set recalculated) when routing paths are calculated, based on changes to these other protocol Sets. Routing Tuples are not subject to timer-based expiration.

6.4. Received Message Information Base

The Received Message Information Base records information required to ensure that a message is processed at most once and is forwarded at

most once per OLSRv2 interface of a router, using MPR flooding.

6.4.1. Received Set

A router has a Received Set per OLSRv2 interface. Each Received Set records the signatures of messages which have been received over that OLSRv2 interface. Each consists of Received Tuples:

(RX_type, RX_orig_addr, RX_seq_number, RX_time)

where:

RX_type is the received Message Type;

RX_orig_addr is the originator address of the received message, note that this does not include a prefix length;

RX_seq_number is the message sequence number of the received message;

RX_time specifies the time at which this Tuple expires and MUST be removed.

6.4.2. Processed Set

A router has a single Processed Set which records signatures of messages which have been processed by the router. It consists of Processed Tuples:

(P_type, P_orig_addr, P_seq_number, P_time)

where:

P_type is the processed Message Type;

P_orig_addr is the originator address of the processed message, note that this does not include a prefix length;

P_seq_number is the message sequence number of the processed message;

P_time specifies the time at which this Tuple expires and MUST be removed.

6.4.3. Forwarded Set

A router has a single Forwarded Set which records signatures of messages which have been forwarded by the router. It consists of Forwarded Tuples:

(F_type, F_orig_addr, F_seq_number, F_time)

where:

F_type is the forwarded Message Type;

F_orig_addr is the originator address of the forwarded message, note that this does not include a prefix length;

F_seq_number is the message sequence number of the forwarded message;

F_time specifies the time at which this Tuple expires and MUST be removed.

6.5. Corresponding Protocol Tuples

In a number of cases there is a natural correspondence from a Protocol Tuple in a Protocol Set to a single Protocol Tuple in another Protocol Set. The latter Protocol Tuple is referred to as "corresponding" to the former.

Specific examples include:

- There is a Local Interface Tuple corresponding to each Link Tuple, where the Link Tuple is in the Link Set for an OLSRv2 interface, and the Local Interface Tuple represents that OLSRv2 interface.

- There is a Neighbor Tuple corresponding to each Link Tuple which has L_HEARD_time not expired, such that N_neighbor_addr_list contains L_neighbor_iface_addr_list.

- There is a Link Tuple (in the Link Set in the same Interface Information Base) corresponding to each 2-Hop Tuple such that L_neighbor_iface_addr_list = N2_neighbor_iface_addr_list.

- There is a Neighbor Tuple corresponding to each 2-Hop Tuple, such that N_neighbor_addr_list contains N2_neighbor_iface_addr_list.

- There is an Advertising Remote Router Tuple corresponding to each Router Topology Tuple such that AR_orig_addr = TR_from_orig_addr.

- o There is an Advertising Remote Router Tuple corresponding to each Routable Address Topology Tuple such that AR_orig_addr = TA_from_orig_addr.

- o There is an Advertising Remote Router Tuple corresponding to each Attached Network Tuple such that AR_orig_addr = AN_orig_addr.

- o There is an Neighbor Tuple corresponding to each Routing Tuple such that N_neighbor_addr_list contains R_next_iface_addr.

7. Message Processing and Forwarding

This protocol defines, and hence owns, the TC message type (see Section 20). Thus, as specified in [RFC5444], this protocol receives all TC messages and is responsible for determining whether and how each TC message is to be processed (updating Information Bases) and/or forwarded, according to this specification. OLSRv2 does not require any part of the Packet Header.

This protocol also receives HELLO messages, which are defined, and hence owned, by [NHDP]. Such messages, when received on an OLSRv2 interface, are made available to this protocol in two ways, both as permitted by [NHDP]. First, such received HELLO messages MUST be made available to this protocol on reception, which allows them to be discarded before being processed by [NHDP], for example if the information added to the HELLO message by this protocol is inconsistent. Second, such received HELLO messages MUST be made available to OLSRv2 after [NHDP] has completed its processing thereof, unless discarded as malformed by [NHDP], for processing by this protocol. HELLO messages are not forwarded by this protocol.

Extensions to this protocol which define, and hence own, other Messages Types, MAY manage the processing and/or forwarding of these messages using the same mechanism as for TC messages. These mechanisms contain elements (P_type, RX_type, F_type) required only for such usage.

The processing selection and forwarding mechanisms are designed to only need to parse the Message Header in order to determine whether a message is to be processed and/or forwarded, and not to have to parse the Message Body even if the message is forwarded (but not processed). An implementation MAY either only parse the Message Body if necessary, or MAY always parse the Message Body.

An implementation MUST discard the message silently if it is unable to parse the Message Header or (if attempted) the Message Body.

7.1. Actions when Receiving a Message

If the router receives a HELLO message from [NHDP], then the message may be rejected before processing by [NHDP] or processed after processing by [NHDP], both according to Section 10.

A router MUST perform the following tasks for each received TC message (or other Message Type defined by an extension to this protocol and specified to use this process):

1. If the router recognizes from the originator address of the message that the message is one which the receiving router itself originated (i.e. is the originator address of this router, or is an O_orig_addr in an Originator Tuple) then the message MUST be silently discarded.

2. Otherwise:

    1. If the message is of a type which may be processed, including being a TC message, then the message is considered for processing according to Section 7.2, AND;

    2. If the message is of a type which may be forwarded, including being a TC message, AND:

        + <msg-hop-limit> is present and <msg-hop-limit> > 1, AND;

        + <msg-hop-count> is not present or <msg-hop-count> < 255

        then the message is considered for forwarding according to Section 7.3.

7.2. Message Considered for Processing

If a message (the "current message") is considered for processing, then the following tasks MUST be performed:

1. If a Processed Tuple exists with:

    * P_type = the Message Type of the current message, AND;

    * P_orig_addr = the originator address of the current message, AND;

    * P_seq_number = the message sequence number of the current message;

    then the current message MUST NOT be processed.

2. Otherwise:

   1. Create a Processed Tuple with:

      + P_type := the Message Type of the current message;

      + P_orig_addr := the originator address of the current message;

      + P_seq_number := the sequence number of the current message;

      + P_time := current time + P_HOLD_TIME.

   2. Process the current message according to its type. For a TC message this is as defined in Section 12.

7.3. Message Considered for Forwarding

If a message (the "current message") is considered for forwarding, then the following tasks MUST be performed:

1. If the sending address (i.e., the source address of the IP datagram containing the current message) does not match (taking into account any address prefix) an address in an L_neighbor_iface_addr_list of a Link Tuple, with L_status = SYMMETRIC, in the Link Set for the OLSRv2 interface on which the current message was received (the "receiving interface") then the current message MUST be silently discarded.

2. Otherwise:

   1. If a Received Tuple exists in the Received Set for the receiving interface, with:

      + RX_type = the Message Type of the current message, AND;

      + RX_orig_addr = the originator address of the current message, AND;

      + RX_seq_number = the sequence number of the current message;

      then the current message MUST be silently discarded.

   2. Otherwise:

1. Create a Received Tuple in the Received Set for the receiving interface with:

   - RX_type := the Message Type of the current message;

   - RX_orig_addr := originator address of the current message;

   - RX_seq_number := sequence number of the current message;

   - RX_time := current time + RX_HOLD_TIME.

2. If a Forwarded Tuple exists with:

   - F_type = the Message Type of the current message, AND;

   - F_orig_addr = the originator address of the current message, AND;

   - F_seq_number = the sequence number of the current message.

   then the current message MUST be silently discarded.

3. Otherwise if the sending address matches (taking account of any address prefix) any address in an L_neighbor_iface_addr_list of a Link Tuple in the Link Set for the receiving OLSRv2 interface which has L_status = SYMMETRIC and whose corresponding Neighbor Tuple has N_mpr_selector = true, then:

   1. Create a Forwarded Tuple with:

      o F_type := the Message Type of the current message;

      o F_orig_addr := originator address of the current message;

      o F_seq_number := sequence number of the current message;

      o F_time := current time + F_HOLD_TIME.

   2. The Message Header of the current message is modified by:

o if present, decrement <msg-hop-limit> in the Message Header by 1, AND;

o if present, increment <msg-hop-count> in the Message Header by 1.

3. For each OLSRv2 interface of the router, include the message in a packet to be transmitted on that OLSRv2 interface, as described in Section 8. This packet MAY contain other forwarded messages and/or messages generated by this router, including by other protocols using [RFC5444]. Forwarded messages MAY be jittered as described in [RFC5148]. The value of MAXJITTER used in jittering a forwarded message MAY be based on information in that message (in particular any INTERVAL_TIME or VALIDITY_TIME TLVs in that message) or otherwise SHOULD be with a maximum delay of F_MAXJITTER. A router MAY modify the jitter applied to a message in order to more efficiently combine messages in packets, as long as the maximum jitter is not exceeded.

8. Packets and Messages

The packet and message format used by this protocol is defined in [RFC5444]. Except as otherwise noted, options defined in [RFC5444] may be freely used, in particular alternative formats defined by packet, message, Address Block and TLV flags.

This protocol may extend HELLO messages (owned by [NHDP]) by adding a message originator address and/or TLVs to these messages when sent over OLSRv2 interfaces, and processes these HELLO messages after their processing by NHDP, as permitted by [NHDP].

This protocol defines and owns the TC Message Type. Extensions to this protocol MAY define additions to TC messages. These MAY include new Message TLVs and/or Address Block TLVs. Extensions MAY also include new Messsage Types to be handled similarly to TC messages. See Section 18.

Routers using this protocol exchange information through messages. One or more messages sent by a router at the same time SHOULD be combined into a single packet (size permitting). These messages may have originated at the sending router, or have originated at another router and are forwarded by the sending router. Messages with different originating routers MAY be combined for transmission within the same packet. Messages from other protocols defined using [RFC5444] MAY be combined for transmission within the same packet.

The remainder of this section defines, within the framework of [RFC5444], Message Types and TLVs specific to this protocol. All references in this specification to TLVs that do not indicate a type extension, assume Type Extension = 0. TLVs in processed messages with a type extension which is neither zero as so assumed, nor a specifically indicated non-zero type extension, are ignored.

8.1. HELLO Messages

A HELLO message is generated as specified in [NHDP]. In addition, a router using this protocol MUST be able to add information to such messages, prior to these being sent on an OLSRv2 interface, as permitted by [NHDP], so that all HELLO messages sent on an OLSRv2 interface:

- MUST allow a message originator address to be determined. This will usually use the message's <msg-orig-addr> element as defined in [RFC5444]. There are two permitted exceptions when the router MAY omit a <msg-orig-addr> element, but an originator address of the message is still correctly defined:

  * If the message contains only a single local interface address, and that address is equal to this router's originator address, then that local interface address is the message originator address.

  * If the message contains no local interface addresses, then, as specified in [NHDP], the source address of the IP datagram containing the message is recognised as the only interface address of the router. In this case, that address is also the message originator address.

- MUST, if it is including any addresses from an N_neighbor_addr_list that has N_mpr = true and are associated with a TLV with Type = LINK_STATUS and Value = SYMMETRIC, include TLV(s) with Type := MPR associated with at least one such address from each such N_neighbor_addr_list.

- MUST NOT include any TLVs with Type = MPR associated with any other addresses.

- MAY include a message TLV with Type := MPR_WILLING, indicating the router's willingness to be selected as an MPR.

An router using this protocol MUST also be able to access any incoming HELLO message received on an OLSRv2 interface, subsequent to the processing specified in [NHDP], as permitted by [NHDP].

8.1.1. HELLO Message TLVs

In a HELLO message, a router MUST include an MPR_WILLING Message TLV as specified in Table 1, unless WILLINGNESS = WILL_DEFAULT (in which case it MAY be included). A router MUST NOT include more than one MPR_WILLING Message TLV.

Type	Value Length	Value
MPR_WILLING	1 octet	Router parameter WILLINGNESS; unused bits (based on the maximum willingness value WILL_ALWAYS) are RESERVED and SHOULD be set to zero.

Table 1: MPR_WILLING TLV definition

If a router does not advertise an MPR_WILLING TLV in a HELLO message, then the router MUST be assumed to have WILLINGNESS equal to WILL_DEFAULT.

8.1.2. HELLO Message Address Block TLVs

In a HELLO message, a router MAY include MPR Address Block TLV(s) as specified in Table 2.

Type	Value Length	Value
MPR	0 octets	None.

Table 2: MPR TLV definition

8.2. TC Messages

A TC message MUST contain:

o A message originator address, using the message's <msg-orig-addr> element as defined in [RFC5444].

o <msg-seq-num> and <msg-hop-limit> elements, as specified in [RFC5444].

o A <msg-hop-count> element in its Message Header if the message contains a TLV with either Type = VALIDITY_TIME or Type = INTERVAL_TIME indicating more than one time value according to

distance. (A TC message MAY contain <msg-hop-count> even if it does not need to.)

- A single Message TLV with Type := CONT_SEQ_NUM, and Type Extension := COMPLETE or Type Extension := INCOMPLETE, as specified in Section 8.2.1 (for complete and incomplete TC messages, respectively) except that the latter MAY be omitted if the message does not contain any addresses associated with a TLV with Type = NBR_ADDR_TYPE or Type = GATEWAY.

- A Message TLV with Type := VALIDITY_TIME, as specified in [RFC5497]. The options included in [RFC5497] for representing zero and infinite times MUST NOT be used.

- If the TC message is complete, all addresses which are the N_orig_addr of a Neighbor Tuple with N_advertised = true, each associated with a TLV with Type = NBR_ADDR_TYPE, and Value = ORIGINATOR, or with Value = ROUTABLE_ORIG if also to be associated with Value = ROUTABLE, see Section 8.2.2. If the TC message is incomplete then any such addresses MAY be included; if any such addresses are included then this MUST be with the appropriate associated TLV(s).

- If the TC message is complete, all routable addresses which are in the N_neighbor_addr_list of a Neighbor Tuple with N_advertised = true. Each such address MUST be associated with a TLV with Type = NBR_ADDR_TYPE, and Value = ROUTABLE, or with Value = ROUTABLE_ORIG if also to be associated with Value = ORIGINATOR, see Section 8.2.2. If the TC message is incomplete then any such addresses MAY be included; if any such addresses are included then this MUST be with the appropriate associated TLV(s).

- If the TC message is complete, all addresses which are the AL_net_addr of a Local Attached Network Tuple. Each such address MUST be associated with a TLV with Type = GATEWAY, and Value = AN_dist as specified in Section 8.2.2. If the TC message is incomplete then any such addresses MAY be included; if included then this MUST be with the appropriate associated TLV.

A TC message MAY contain:

- A Message TLV with Type := INTERVAL_TIME, as specified in [RFC5497]. The options included in [RFC5497] for representing zero and infinite times MUST NOT be used.

8.2.1. TC Message TLVs

In each TC message which contains any addresses associated with a TLV with Type = NBR_ADDR_TYPE or Type = GATEWAY, a router MUST include a single CONT_SEQ_NUM Message TLV, as specified in Table 3, and with Type Extension = COMPLETE or Type Extension = INCOMPLETE, according to whether the TC message is complete or incomplete.

Type	Value Length	Value
CONT_SEQ_NUM	2 octets	The ANSN contained in the Neighbor Information Base.

Table 3: CONT_SEQ_NUM TLV definition

8.2.2. TC Message Address Block TLVs

In a TC message, a router MAY include NBR_ADDR_TYPE Address Block TLV(s) as specified in Table 4.

Type	Value Length	Value
NBR_ADDR_TYPE	1 octet	ORIGINATOR indicates that the address is an originator address, ROUTABLE indicates that the address is a routable address of an interface, ROUTABLE_ORIG indicates that the address is both

Table 4: NBR_ADDR_TYPE TLV definition

If an address is both a originator address and a routable interface address, then it may be associated, using a TLV with Type = NBR_ADDR_TYPE, with either a Value = ROUTABLE_ORIG, or (using two separate TLVs) both with Value = ORIGINATOR and with Value = ROUTABLE.

In a TC message, a router MAY include GATEWAY Address Block TLV(s) as specified in Table 5.

Type	Value Length	Value
GATEWAY	1 octet	Number of hops to attached network.

Table 5

All addresses included in a TC message according to this specification MUST be associated with either at least one TLV with Type = NBR_ADDR_TYPE or a TLV with Type = GATEWAY, but not both. Other addresses MAY be included in the TC message, but (other than the message originator address) are ignored by this specification.

9. HELLO Message Generation

An HELLO message is composed and generated as defined in [NHDP], extended by the following being added to the HELLO message by this protocol before being sent over an OLSRv2 interface, as permitted by [NHDP]:

- A message originator address, using a <msg-orig-addr> element, unless:

  * The message contains only a single local interface address, which is then interpreted as the message originator address, OR;

  * The message does not include any local interface addresses, as permitted by the specification in [NHDP] when the router that generated the HELLO message has only one interface address, and will use that as the sending address of the IP datagram in which the HELLO message is contained. In this case that address MAY also be used as the message originator address.

- A Message TLV with Type := MPR_WILLING and Value := WILLINGNESS MUST be included, unless WILLINGNESS = WILL_DEFAULT (in which case it MAY be included).

- For each Neighbor Tuple with N_mpr = true, and for which one or more addresses in its N_neighbor_addr_list are included with an associated TLV with Type = LINK_STATUS and Value = SYMMETRIC, at least one of these addresses (including a different copy of that address, in the same or a different Address Block) MUST be associated with an Address Block TLV with Type := MPR. Note that other addresses (which do not meet this specification) MUST NOT be associated with an Address Block TLV with Type = MPR, but that more than one address from the same qualifying

N_neighbor_addr_list MAY be associated with an Address Block TLV with Type := MPR.

o An additional HELLO message MAY be sent when the router's set of MPRs changes, in addition to the cases specified in [NHDP], and subject to the same constraints.

9.1. HELLO Message: Transmission

HELLO messages are included in packets as specified in [RFC5444]. These packets may contain other messages, including TC messages.

10. HELLO Message Processing

All HELLO 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 = MPR_WILLING refer to a TLV with Type = MPR_WILLING and Type Extension = 0.

In addition to the reasons specified in [NHDP] for discarding a HELLO message on reception, a HELLO message MUST be discarded before processing by [NHDP] or this specification if it:

o Has more than one TLV with Type = MPR_WILLING in its Message TLV Block.

o Has a message originator address, or any address associated with a TLV with Type = LOCAL_IF, that the receiving router has recorded as:

  * its originator address, OR;

  * as the O_orig_addr in an Originator Tuple, OR;

  * in an I_local_iface_addr_list in a Local Interface Tuple, OR;

  * as the IR_local_iface_addr in a Removed Interface Address Tuple, OR;

  * as the AL_net_addr in a Local Attached Network Tuple.

  Note that some of these cases are already excluded by [NHDP].

o Includes any address associated with a TLV with Type = LINK_STATUS or Type = OTHER_NEIGHB that is also the message's originator address.

o Contains any address associated with a TLV with Type = MPR, where that address (including a different copy of that address, in the same or a different Address Block) is not also associated with a TLV with Type = LINK_STATUS and Value = SYMMETRIC.

HELLO messages are first processed as specified in [NHDP]. That processing includes identifying (or creating) a Neighbor Tuple corresponding to the originator of the HELLO message (the "current Neighbor Tuple"). After this, the following MUST be performed:

1. If the HELLO message has a well-defined message originator address, i.e., has an <msg-orig-addr> element or has zero or one addresses associated with a TLV with Type = LOCAL_IF:

    1. Remove any other Neighbor Tuples with N_orig_addr = message originator address, taking any consequent action (including removing one or more Link Tuples) as specified in [NHDP].

    2. The current Neighbor Tuple is then updated according to:

        1. N_orig_addr := message originator address;

        2. Update N_willingness as described in Section 10.1;

        3. Update N_mpr_selector as described in Section 10.2.

2. If there are any changes to the router's Information Bases, then perform the processing defined in Section 13.

10.1. Updating Willingness

N_willingness in the current Neighbor Tuple is updated as follows:

1. If the HELLO message contains a Message TLV with Type = MPR_WILLING then N_willingness := the Value of that TLV;

2. Otherwise, N_willingness := WILL_DEFAULT.

10.2. Updating MPR Selectors

N_mpr_selector is updated as follows:

1. If a router finds any of its local interface addresses (i.e., those contained in the I_local_iface_addr_list of an OLSRv2 interface) with an associated TLV with Type = MPR in the HELLO message (indicating that the originating router has selected the receiving router as an MPR) then, for the current Neighbor Tuple:

* N_mpr_selector := true

2. Otherwise (i.e., if no such address and TLV were found) if a router finds any of its local interface addresses with an associated TLV with Type = LINK_STATUS and Value = SYMMETRIC in the HELLO message, then for the current Neighbor Tuple:

   * N_mpr_selector := false

   * N_advertised := false

11. TC Message Generation

A router with one or more OLSRv2 interfaces, and with any Neighbor Tuples with N_advertised = true, or with a non-empty Local Attached Network Set MUST generate TC messages. A router which does not have such information to advertise SHOULD also generate "empty" TC messages for a period A_HOLD_TIME after it last generated a non-empty TC message. TC messages (non-empty and empty) are generated according to the following:

1. The message originator address MUST be the router's originator address.

2. The message hop count, if included, MUST be set to zero.

3. The message hop limit MUST be set to a value greater than 1. A router MAY use the same hop limit TC_HOP_LIMIT in all TC messages, or use different values of the hop limit TC_HOP_LIMIT in TC messages, see Section 5.8.

4. The message MUST contain a Message TLV with Type := CONT_SEQ_NUM and Value := ANSN from the Neighbor Information Base. If the TC message is complete then this Message TLV MUST have Type Extension := COMPLETE, otherwise it MUST have Type Extension := INCOMPLETE. (Exception: a TC message MAY omit such a Message TLV if the TC message is not reporting any addresses with associated TLV with Type = NBR_ADDR_TYPE or Type = GATEWAY.)

5. The message MUST contain a Message TLV with Type := VALIDITY_TIME, as specified in [RFC5497]. If all TC messages are sent with the same hop limit then this TLV MUST have Value := T_HOLD_TIME. If TC messages are sent with different hop limits (more than one value of TC_HOP_LIMIT) then this TLV MUST specify times which vary with the number of hops distance appropriate to the chosen pattern of TC message hop limits, as specified in [RFC5497], these times SHOULD be appropriate multiples of T_HOLD_TIME.

6. The message MAY contain a Message TLV with Type := INTERVAL_TIME, as specified in [RFC5497]. If all TC messages are sent with the same hop limit then this TLV MUST have Value := TC_INTERVAL. If TC messages are sent with different hop limits, then this TLV MUST specify times which vary with the number of hops distance appropriate to the chosen pattern of TC message hop limits, as specified in [RFC5497], these times SHOULD be appropriate multiples of TC_INTERVAL.

7. A complete message MUST include, and an incomplete message MAY include, in its Address Blocks:

    1. N_orig_addr in each Neighbor Tuple with N_advertised = true, associated with a TLV with Type := NBR_ADDR_TYPE and Value := ORIGINATOR (or Value := ROUTABLE_ORIG if also to be associated with Value = ROUTABLE).

    2. Each routable address in an N_neighbor_addr_list in each Neighbor Tuple with N_advertised = true, associated with a TLV with Type := NBR_ADDR_TYPE and Value := ROUTABLE (or Value := ROUTABLE_ORIG if also to be associated with Value = ORIGINATOR).

    3. AL_net_addr in each Local Attached Neighbor Tuple, each associated with a TLV with Type := GATEWAY and Value := AL_dist.

11.1. TC Message Transmission

Complete TC messages are generated and transmitted periodically on all OLSRv2 interfaces, with a default interval between two consecutive TC transmissions by the same router of TC_INTERVAL.

TC messages MAY be generated in response to a change in the information which they are to advertise, indicated by a change in ANSN. In this case a router MAY send a complete TC message, and if so MAY re-start its TC message schedule. Alternatively a router MAY send an incomplete TC message with at least the newly advertised addresses (i.e. not previously, but now, an N_orig_addr or an N_neighbor_addr_list in a Neighbor Tuple with N_advertised = true, or in an AL_net_addr) in its Address Blocks, with associated TLV(s). Note that a router cannot report removal of advertised content using an incomplete TC message.

When sending a TC message in response to a change of advertised addresses, a router must respect a minimum interval of TC_MIN_INTERVAL between generated TC messages. Sending an incomplete TC message MUST NOT cause the interval between complete TC messages

to be increased, and thus a router MUST NOT send an incomplete TC message if within TC_MIN_INTERVAL of the next scheduled complete TC message.

The generation of TC messages, whether scheduled or triggered by a change of contents MAY be jittered as described in [RFC5148]. The values of MAXJITTER used SHOULD be:

- o TP_MAXJITTER for periodic TC message generation;

- o TT_MAXJITTER for responsive TC message generation.

TC messages are included in packets as specified in [RFC5444]. These packets MAY contain other messages, including HELLO messages and TC messages with different originator addresses. TC messages are forwarded according to the specification in Section 7.3.

12. TC Message Processing

On receiving a TC message, a router MUST first check if the message is invalid for processing by this router, as defined in Section 12.1. Otherwise the receiving router MUST update its appropriate Interface Information Base and its Router Information Base as specified in Section 12.2.

All TC message processing, including determination of whether a message is invalid, unless otherwise noted considers only TLVs with Type Extension = 0. TLVs with any other type extension (or any unmentioned type extension when other type extensions are considered) are ignored. All references to, for example, a TLV with Type = VALIDITY_TIME refer to a TLV with Type = VALIDITY_TIME and Type Extension = 0.

Following TC message processing, if there are any changes in the router's Information Bases, then the processing in Section 13 MUST be performed.

12.1. Invalid Message

A received TC message is invalid for processing by this router if the message:

- o Does not include a message originator address, a message sequence number, and a hop limit.

- o Does not include a hop count, and contains a multi-value TLV with Type = VALIDITY_TIME or Type = INTERVAL_TIME, as defined in [RFC5497].

- o Does not have exactly one TLV with Type = VALIDITY_TIME in its Message TLV Block.

- o Has more than one TLV with Type = INTERVAL_TIME in its Message TLV Block.

- o Does not have a TLV with Type = CONT_SEQ_NUM and Type Extension = COMPLETE or Type Extension = INCOMPLETE in its Message TLV Block, and contains at least one address associated with a TLV with Type = NBR_ADDR_TYPE or Type = GATEWAY.

- o Has more than one TLV with Type = CONT_SEQ_NUM and Type Extension = COMPLETE or Type Extension = INCOMPLETE in its Message TLV Block.

- o Has a message originator address, or any address associated with a TLV with Type = NBR_ADDR_TYPE or Type = GATEWAY, that the receiving router has recorded as:

  - * its originator address, OR;

  - * as the O_orig_addr in an Originator Tuple, OR;

  - * in an I_local_iface_addr_list in a Local Interface Tuple, OR;

  - * the IR_local_iface_addr in a Removed Interface Address Tuple.

- o Has a message originator address, or any address associated with a TLV with Type = NBR_ADDR_TYPE, that the receiving router has recorded as the AL_net_addr in a Local Attached Network Tuple.

- o Includes any address with a prefix length which is not maximal (equal to the address length, in bits) associated with a TLV with Type = NBR_ADDR_TYPE and Value = ORIGINATOR or Value = ROUTABLE_ORIG.

- o Includes any non-routable address associated with a TLV with Type = NBR_ADDR_TYPE and Value = ROUTABLE or Value = ROUTABLE_ORIG.

- o Includes any address associated with a TLV with Type = NBR_ADDR_TYPE or Type = GATEWAY that is also the message's originator address.

- o Associates any address (including different copies of an address, in the same or different Address Blocks) with more than one single Value using one or more TLV(s) with Type = GATEWAY.

o Associates any address (including different copies of an address, in the same or different Address Blocks) with TLVs with Type = NBR_ADDR_TYPE and Type = GATEWAY.

A router MAY recognize additional reasons for identifying that a message is invalid. An invalid message MUST be silently discarded, without updating the router's Information Bases.

12.2. TC Message Processing Definitions

When, according to Section 7.2, a TC message is to be "processed according to its type", this means that:

o If the TC message contains a Message TLV with Type = CONT_SEQ_NUM and Type Extension = COMPLETE, then processing according to Section 12.3 and then according to Section 12.4 is carried out.

o If the TC message contains a Message TLV with Type = CONT_SEQ_NUM and Type Extension = INCOMPLETE, then only processing according to Section 12.3 is carried out.

For the purposes of this section:

o "validity time" is calculated from a VALIDITY_TIME Message TLV in the TC message according to the specification in [RFC5497]. All information in the TC message has the same validity time.

o "received ANSN" is defined as being the Value of a Message TLV with Type = CONT_SEQ_NUM.

o Comparisons of sequence numbers are carried out as specified in Section 17.

12.3. Initial TC Message Processing

The TC message is processed as follows:

1. The Advertising Remote Router Set is updated according to Section 12.3.1. If the TC message is indicated as discarded in that processing then the following steps are not carried out.

2. The Router Topology Set is updated according to Section 12.3.2.

3. The Routable Address Topology Set is updated according to Section 12.3.3.

4. The Attached Network Set is updated according to Section 12.3.4.

12.3.1. Populating the Advertising Remote Router Set

The router MUST update its Advertising Remote Router Set as follows:

1. If there is an Advertising Remote Router Tuple with:

   * AR_orig_addr = message originator address; AND

   * AR_seq_number > received ANSN

   then the TC message MUST be discarded.

2. Otherwise:

   1. If there is no Advertising Remote Router Tuple such that:

      + AR_orig_addr = message originator address;

      then create an Advertising Remote Router Tuple with:

      + AR_orig_addr := message originator address.

   2. This Advertising Remote Router Tuple (existing or new) is then modified as follows:

      + AR_seq_number := received ANSN;

      + AR_time := current time + validity time.

12.3.2. Populating the Router Topology Set

The router MUST update its Router Topology Set as follows:

1. For each address (henceforth advertised address) in an Address Block that has an associated TLV with Type = NBR_ADDR_TYPE and Value = ORIGINATOR or Value = ROUTABLE_ORIG, perform the following processing:

   1. If there is no Router Topology Tuple such that:

      + TR_from_orig_addr = message originator address; AND

      + TR_to_orig_addr = advertised address

      then create a new Router Topology Tuple with:

      + TR_from_orig_addr := message originator address

+ TR_to_orig_addr := advertised address.

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

   + TR_seq_number := received ANSN;

   + TR_time := current time + validity time.

12.3.3. Populating the Routable Address Topology Set

The router MUST update its Routable Address Topology Set as follows:

1. For each address (henceforth advertised address) in an Address Block that has an associated TLV with Type = NBR_ADDR_TYPE and Value = ROUTABLE or Value = ROUTABLE_ORIG, perform the following processing:

   1. If there is no Routable Address Topology Tuple such that:

      + TA_from_orig_addr = message originator address; AND

      + TA_dest_addr = advertised address

      then create a new Routable Address Topology Tuple with:

      + TA_from_orig_addr := message originator address;

      + TA_dest_addr := advertised address.

   2. This Routable Address Topology Tuple (existing or new) is then modified as follows:

      + TA_seq_number := received ANSN;

      + TA_time := current time + validity time.

12.3.4. Populating the Attached Network Set

The router MUST update its Attached Network Set as follows:

1. For each address (henceforth advertised address) in an Address Block that has an associated TLV with Type = GATEWAY, and is not an AL_net_addr in a Local Attached Network Tuple, perform the following processing:

   1. If there is no Attached Network Tuple such that:

+ AN_net_addr = network address; AND

+ AN_orig_addr = message originator address

then create a new Attached Network Tuple with:

+ AN_net_addr := network address;

+ AN_orig_addr := message originator address.

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

   + AN_dist := the Value of the associated GATEWAY TLV;

   + AN_seq_number := received ANSN;

   + AN_time := current time + validity time.

12.4. Completing TC Message Processing

The TC message is processed as follows:

1. The Router Topology Set is updated according to Section 12.4.1.

2. The Routable Address Topology Set is updated according to Section 12.4.2.

3. The Attached Network Set is updated according to Section 12.4.3.

12.4.1. Purging the Router Topology Set

The Router Topology Set MUST be updated as follows:

1. Any Router Topology Tuples with:

   * TR_from_orig_addr = message originator address; AND

   * TR_seq_number < received ANSN

   MUST be removed.

12.4.2. Purging the Routable Address Topology Set

The Routable Address Topology Set MUST be updated as follows:

1. Any Routable Address Topology Tuples with:

   * TA_from_orig_addr = message originator address; AND

   * TA_seq_number < received ANSN

   MUST be removed.

12.4.3. Purging the Attached Network Set

The Attached Network Set MUST be updated as follows:

1. Any Attached Network Tuples with:

   * AN_orig_addr = message originator address; AND

   * AN_seq_number < received ANSN

   MUST be removed.

13. Information Base Changes

The changes described in the following sections MUST be carried out when any Information Base changes as indicated.

13.1. Originator Address Changes

If the router changes originator address, then:

1. If there is no Originator Tuple with:

   * O_orig_addr = old originator address

   then create an Originator Tuple with:

   * O_orig_addr := old originator address

   The Originator Tuple (existing or new) with:

   * O_orig_addr = new originator address

   is then modified as follows:

   * O_time := current time + O_HOLD_TIME

13.2. Neighbor State Changes

The N_mpr_selector and N_advertised flags in Neighbor Tuples MUST be maintained according to the following rules:

1. If N_symmetric = false, then N_mpr_selector = false and N_advertised = false.

2. If N_mpr_selector = true, then N_advertised = true.

3. In other cases (i.e. N_symmetric = true and N_mpr_selector = false) a router MAY select N_advertised = true or N_advertised = false. The more neighbors that are advertised, the larger TC messages become, but the more redundancy is available for routing. A router SHOULD consider the nature of its network in making such a decision, and SHOULD avoid unnecessary changes in advertising status, which may result both in additional TC messages having to be sent by its neighbors, and in unnecessary changes to routing, which will have similar effects to other forms of topology changes in the MANET.

13.3. Advertised Neighbor Changes

The router MUST increment the ANSN in the Neighbor Information Base whenever:

1. Any Neighbor Tuple changes its N_advertised value.

2. N_orig_addr is changed, or any routable address is added to or removed from any Neighbor Tuple with N_advertised = true.

3. There is any change to the Local Attached Network Set.

13.4. Advertising Remote Router Tuple Expires

The Router Topology Set, the Routable Address Topology Set and the Attached Network Set MUST be changed when an Advertising Remote Router Tuple expires (AR_time is reached). The following changes are required before the Advertising Remote Router Tuple is removed:

1. All Router Topology Tuples with:

   * TR_from_orig_addr = AR_orig_addr of the Advertising Remote Router Tuple

   are removed.

2. All Routable Address Topology Tuples with:

   * TA_from_orig_addr = AR_orig_addr of the Advertising Remote Router Tuple

   are removed.

3. All Attached Network Tuples with:

   * AN_orig_addr = AR_orig_addr of the Advertising Remote Router Tuple

   are removed.

13.5. Neighborhood Changes and MPR Updates

The set of symmetric 1-hop neighbors selected as MPRs MUST satisfy the conditions defined in Section 14. To ensure this:

1. The set of MPRs of a router MUST be recalculated if:

   * a Link Tuple is added with L_status = SYMMETRIC, OR;

   * a Link Tuple with L_status = SYMMETRIC is removed, OR;

   * a Link Tuple with L_status = SYMMETRIC changes to having L_status = HEARD or L_status = LOST, OR;

   * a Link Tuple with L_status = HEARD or L_status = LOST changes to having L_status = SYMMETRIC, OR;

   * a 2-Hop Tuple is added or removed, OR;

   * the N_willingness of a Neighbor Tuple with N_symmetric = true changes from WILL_NEVER to any other value, OR;

   * the N_willingness of a Neighbor Tuple with N_symmetric = true and N_mpr = true changes to WILL_NEVER from any other value, OR;

   * the N_willingness of a Neighbor Tuple with N_symmetric = true and N_mpr = false changes to WILL_ALWAYS from any other value.

2. Otherwise, the set of MPRs of a router MAY be recalculated if the N_willingness of a Neighbor Tuple with N_symmetric = true changes in any other way; it SHOULD be recalculated if N_mpr = false and this is an increase in N_willingness or if N_mpr = true and this is a decrease in N_willingness.

If the set of MPRs of a router is recalculated, this MUST be as described in Section 14. Before that calculation, the N_mpr of all Neighbor Tuples are set false (although the previous values of N_mpr MAY be used by an algorithm that minimises changes to the set of MPRs). After that calculation the N_mpr of all Neighbor Tuples representing symmetric 1-hop neighbors which are chosen as MPRs, are

set true.

13.6. Routing Set Updates

The Routing Set MUST be updated, as described in Section 15 when changes in the Local Information Base, the Neighborhood Information Base or the Topology Information Base indicate a change of the known symmetric links and/or attached networks in the MANET, hence changing the Topology Graph. It is sufficient to consider only changes which affect at least one of:

- The Local Interface Set, if the change removes any address in an I_local_iface_addr_list. In this case, unless the OLSRv2 interface is removed, it may not be necessary to do more than replace such addresses, if used, by an alternative address from the same I_local_iface_addr_list.

- The Local Attached Set, if the change removes any AL_net_addr which is also an AN_net_addr. In this case it may not be necessary to do more than add and remove Routing Tuples with R_dest_addr equal to that AN_net_addr.

- The Link Set of any OLSRv2 interface, and to consider only Link Tuples which have, or just had, L_status = SYMMETRIC (including removal of such Link Tuples).

- The Neighbor Set of the router, and to consider only Neighbor Tuples that have, or just had, N_symmetric = true, and do not have N_orig_addr = unknown.

- The 2-Hop Set of any OLSRv2 interface, if used in the creation of the Routing Set.

- The Router Topology Set of the router.

- The Routable Address Topology Set of the router.

- The Attached Network Set of the router.

14. Selecting MPRs

Each router MUST select, from among its willing symmetric 1-hop neighbors, a subset of these routers as MPRs. Only MPRs forward control messages flooded through the MANET, thus effecting a flooding reduction, an optimization of the classical flooding mechanism, known as MPR flooding. MPRs MAY also be used to effect a topology reduction in the MANET. Consequently, while it is not essential that the set of MPRs is minimal, keeping the number of MPRs small ensures

that the overhead is kept at a minimum.

A router MUST select MPRs for each of its OLSRv2 interfaces, but then forms the union of those sets as its single set of MPRs. This union MUST include all symmetric 1-hop neighbors with willingness WILL_ALWAYS. Only this overall set of MPRs is relevant, the recorded and used MPR relationship is one of routers, not interfaces. Routers MAY select their MPRs by any process which satisfies the conditions which follow. Routers can freely interoperate whether they use the same or different MPR selection algorithms.

For each OLSRv2 interface a router MUST select a set of MPRs. This set MUST have the properties that:

- o All of the selected MPRs are willing symmetric 1-hop neighbors, AND;

- o If the selecting router sends a message on that OLSRv2 interface, and that message is successfully forwarded by all of the selected MPRs for that interface, then all symmetric strict 2-hop neighbors of the selecting router through that OLSRv2 interface will receive that message over a symmetric link.

Note that it is always possible to select a valid set of MPRs. The set of all willing symmetric 1-hop neighbors of a router is a (maximal) valid set of MPRs for that router. However a router SHOULD NOT select a symmetric 1-hop neighbor with Willingness != WILL_ALWAYS as an MPR if there are no symmetric strict 2-hop neighbors with a symmetric link to that symmetric 1-hop neighbor. Thus a router with no symmetric 1-hop neighbors with willingness WILL_ALWAYS and with no symmetric strict 2-hop neighbors SHOULD NOT select any MPRs.

A router MAY select its MPRs for each OLSRv2 interface independently, or it MAY coordinate its MPR selections across its OLSRv2 interfaces, as long as the required condition is satisfied for each OLSRv2 interface. Each router MAY select its MPRs independently from the MPR selection by other routers, or it MAY, for example, give preference to routers that either are, or are not, already selected as MPRs by other routers.

When selecting MPRs for each OLSRv2 interface independently, this MAY be done using information from the Link Set and 2-Hop Set of that OLSRv2 interface only, and the Neighbor Set of the router (specifically the N_willingness elements).

The selection of MPRs (overall, not per OLSRv2 interface) is recorded in the Neighbor Set of the router (using the N_mpr elements). A selected MPR MUST be a willing symmetric 1-hop neighbor (i.e. the

corresponding N_symmetric = true, and the corresponding N_willingness != WILL_NEVER).

A router MUST recalculate its MPRs whenever the currently selected set of MPRs does not still satisfy the required conditions. It MAY recalculate its MPRs if the current set of MPRs is still valid, but could be more efficient. Sufficient conditions to recalculate a router's set of MPRs are given in Section 13.5.

An example algorithm that creates a set of MPRs that satisfies the required conditions is given in Appendix A.

15. Routing Set Calculation

The Routing Set of a router is populated with Routing Tuples that represent paths from that router to all destinations in the network. These paths are calculated based on the Network Topology Graph, which is constructed from information in the Information Bases, obtained via HELLO and TC message exchange.

Changes to the Routing Set do not require any messages to be transmitted. The state of the Routing Set SHOULD, however, be reflected in IP's routing table by adding and removing entries from IP's routing table as appropriate. Only appropriate Routing Tuples (in particular only those that represent local links or paths to routable addresses) need be reflected in IP's routing table.

15.1. Network Topology Graph

The Network Topology Graph is formed from information from the router's Local Interface Set, Link Sets, Neighbor Set, Router Topology Set, Routable Address Topology Set and Attached Network Set. The Network Topology Graph MAY also use information from the router's 2-Hop Sets. The Network Topology Graph forms the router's topological view of the network in form of a directed graph. The Network Topology Graph has a "backbone" (within which minimum distance routes will be constructed) containing the following edges:

o Edges X -> Y for all possible Y, and one X per Y, such that:

  * Y is the N_orig_addr of a Neighbor Tuple, AND;

  * N_orig_addr is not unknown;

  * X is in the I_local_iface_addr_list of a Local Interface Tuple, AND;

   * There is a Link Tuple with L_status = SYMMETRIC such that this Neighbor Tuple and this Local Interface Tuple correspond to it. An address from L_neighbor_iface_addr_list will be denoted R in this case.

   It SHOULD be preferred, where possible, to select R = S and X from the Local Interface Tuple corresponding to the Link Tuple from which R was selected.

o All edges W -> U such that:

   * W is the TR_from_orig_addr of a Router Topology Tuple, AND;

   * U is the TR_to_orig_addr of the same Router Topology Tuple.

The Network Topology Graph is further "decorated" with the following edges. If an address S, V, Z or T equals an address Y or W, then the edge terminating in the address S, V, Z or T MUST NOT be used in any path.

o Edges X -> S for all possible S, and one X per S, such that:

   * S is in the N_neighbor_addr_list of a Neighbor Tuple, AND;

   * X is in the I_local_iface_addr_list of a Local Interface Tuple, AND;

   * There is a Link Tuple with L_status = SYMMETRIC such that this Neighbor Tuple and this Local Interface Tuple correspond to it. An address from L_neighbor_iface_addr_list will be denoted R in this case.

   It SHOULD be preferred, where possible, to select R = S and X from the Local Interface Tuple corresponding to the Link Tuple from which R was selected.

o All edges W -> V such that:

   * W is the TA_from_orig_addr of a Routable Address Topology Tuple, AND;

   * V is the TA_dest_addr of the same Routable Address Topology Tuple.

o All edges W -> T such that:

   * W is the AN_orig_addr of an Attached Network Tuple, AND;

   * T is the AN_net_addr of the same Attached Network Tuple.

o OPTIONALLY, all edges Y -> Z such that:

   * Z is a routable address and is the N2_2hop_addr of a 2-Hop Tuple, AND;

   * Y is the N_orig_addr of the corresponding Neighbor Tuple, AND;

   * This Neighbor Tuple has N_willingness not equal to WILL_NEVER.

   A path terminating with such an edge SHOULD NOT be used in preference to any other path.

Any part of the Topology Graph which is not connected to an address X is not used. Only one selection X need be made from each I_local_iface_addr_list, and only one selection R need be made from any L_neighbor_iface_addr_list. All edges have a cost (hop count) of one, except edges W -> T which each have a cost (hop count) equal to the appropriate value of AN_dist.

15.2. Populating the Routing Set

The Routing Set MUST contain the shortest paths for all destinations from all local OLSRv2 interfaces using the Network Topology Graph. This calculation MAY use any algorithm, including any means of choosing between paths of equal length.

Using the notation of Section 15.1, initially "backbone" paths using only edges X -> Y and W -> U need be constructed (using a minimum distance algorithm). Then paths using only a final edge of the other types may be added. These MUST NOT replace backbone paths with the same destination (and paths terminating in an edge Y -> Z SHOULD NOT replace paths with any other form of terminating edge).

Each path will correspond to a Routing Tuple. These will be of two types. The first type will represent single edge paths, of type X -> S or X -> Y, by:

o R_local_iface_addr := X;

o R_next_iface_addr := R;

o R_dest_addr := S or Y;

o R_dist := 1,

where R is as defined in Section 15.1 for these types of edges.

The second type will represent a multiple edge path, which will always have first edge of type X -> Y, and will have final edge of type W -> U, W -> V, W -> T or Y -> Z. The Routing Tuple will be:

- R_local_iface_addr := X;
- R_next_iface_addr := Y;
- R_dest_addr := U, V, T or Z;
- R_dist := the total hop count of the path.

Finally, Routing Tuples of the second type whose R_dest_addr is not routable MAY be discarded.

An example algorithm for calculating the Routing Set of a router is given in Appendix B.

16. Proposed Values for Parameters and Constants

This protocol uses all parameters and constants defined in [NHDP] and additional parameters and constants defined in this document. All but one (RX_HOLD_TIME) of these additional parameters are router parameters as defined in [NHDP]. These proposed values of the additional parameters are appropriate to the case where all parameters (including those defined in [NHDP]) have a single value. Proposed values for parameters defined in [NHDP] are given in that document.

16.1. Local History Time Parameters

- O_HOLD_TIME := 30 seconds

16.2. Message Interval Parameters

- TC_INTERVAL := 5 seconds
- TC_MIN_INTERVAL := TC_INTERVAL/4

16.3. Advertised Information Validity Time Parameters

- T_HOLD_TIME := 3 x TC_INTERVAL
- A_HOLD_TIME := T_HOLD_TIME

16.4. Received Message Validity Time Parameters

- o RX_HOLD_TIME := 30 seconds

- o P_HOLD_TIME := 30 seconds

- o F_HOLD_TIME := 30 seconds

16.5. Jitter Time Parameters

- o TP_MAXJITTER := HP_MAXJITTER

- o TT_MAXJITTER := HT_MAXJITTER

- o F_MAXJITTER := TT_MAXJITTER

16.6. Hop Limit Parameter

- o TC_HOP_LIMIT := 255

16.7. Willingness Parameter and Constants

- o WILLINGNESS := WILL_DEFAULT

- o WILL_NEVER := 0

- o WILL_DEFAULT := 3

- o WILL_ALWAYS := 7

17. Sequence Numbers

Sequence numbers are used in this specification for the purpose of discarding "old" information, i.e. messages received out of order. However with a limited number of bits for representing sequence numbers, wrap-around (that the sequence number is incremented from the maximum possible value to zero) will occur. To prevent this from interfering with the operation of this protocol, the following MUST be observed when determining the ordering of sequence numbers.

The term MAXVALUE designates in the following one more than the largest possible value for a sequence number. For a 16 bit sequence number (as are those defined in this specification) MAXVALUE is 65536.

The sequence number S1 is said to be "greater than" the sequence number S2 if:

o S1 > S2 AND S1 - S2 < MAXVALUE/2 OR

o S2 > S1 AND S2 - S1 > MAXVALUE/2

When sequence numbers S1 and S2 differ by MAXVALUE/2 their ordering cannot be determined. In this case, which should not occur, either ordering may be assumed.

Thus when comparing two messages, it is possible - even in the presence of wrap-around - to determine which message contains the most recent information.

18. Extensions

An extension to this protocol will need to interact with this specification, and possibly also with [NHDP]. This protocol is designed to permit such interactions, in particular:

o Through accessing, and possibly extending, the information in the Information Bases. All updates to the elements specified in this document are subject to the constraints specified in [NHDP] and Appendix D.

o Through accessing an outgoing message prior to it being transmitted over any OLSRv2 interface, and to add information to it as specified in [RFC5444]. This MAY include Message TLVs and/or addresses with associated Address Block TLVs. (Addresses without new associated TLVs SHOULD NOT be added to messages.) This may, for example, be to allow a security protocol, as suggested in Section 19, to add a TLV containing a cryptographic signature to the message.

o Through accessing an incoming message, and potentially discarding it prior to processing by this protocol. This may, for example, allow a security protocol as suggested in Section 19 to perform verification of message signatures and prevent processing and/or forwarding of unverifiable messages by this protocol.

o Through accessing an incoming message after it has been completely processed by this protocol. This may, in particular, allow a protocol which has added information, by way of inclusion of appropriate TLVs, or of addresses associated with new TLVs, access to such information after appropriate updates have been recorded in the Information Bases in this protocol.

o Through requesting that a message be generated at a specific time. In that case, message generation MUST still respect the constraints in [NHDP] and Section 5.4.

19. Security Considerations

Currently, this protocol does not specify any special security measures. As a proactive routing protocol, this protocol is a potential target for various attacks. Various possible vulnerabilities are discussed in this section.

19.1. Confidentiality

This protocol periodically MPR floods topological information to all routers in the network. Hence, if used in an unprotected wireless network, the network topology is revealed to anyone who listens to the control messages.

In situations where the confidentiality of the network topology is of importance, regular cryptographic techniques, such as exchange of OLSRv2 control traffic messages encrypted by PGP [RFC4880] or encrypted by some shared secret key, can be applied to ensure that control traffic can be read and interpreted by only those authorized to do so.

19.2. Integrity

Each router is injecting topological information into the network through transmitting HELLO messages and, for some routers, TC messages. If some routers for some reason, malicious or malfunction, inject invalid control traffic, network integrity may be compromised. Therefore, message authentication is recommended.

Different such situations may occur, for instance:

1. a router generates TC messages, advertising links to non-neighbor routers;

2. a router generates TC messages, pretending to be another router;

3. a router generates HELLO messages, advertising non-neighbor routers;

4. a router generates HELLO messages, pretending to be another router;

5. a router forwards altered control messages;

6. a router does not forward control messages;

7. a router does not select multipoint relays correctly;

8. a router forwards broadcast control messages unaltered, but does not forward unicast data traffic;

9. a router "replays" previously recorded control traffic from another router.

Authentication of the originator router for control messages (for situations 2, 4 and 5) and on the individual links announced in the control messages (for situations 1 and 3) may be used as a countermeasure. However to prevent routers from repeating old (and correctly authenticated) information (situation 9) temporal information is required, allowing a router to positively identify such delayed messages.

In general, digital signatures and other required security information may be transmitted as a separate Message Type, or signatures and security information may be transmitted within the HELLO and TC messages, using the TLV mechanism. Either option permits that "secured" and "unsecured" routers can coexist in the same network, if desired,

Specifically, the authenticity of entire control packets can be established through employing IPsec authentication headers, whereas authenticity of individual links (situations 1 and 3) require additional security information to be distributed.

An important consideration is that all control messages are transmitted either to all routers in the neighborhood (HELLO messages) or broadcast to all routers in the network (TC messages).

For example, a control message in this protocol is always a point-to-multipoint transmission. It is therefore important that the authentication mechanism employed permits that any receiving router can validate the authenticity of a message. As an analogy, given a block of text, signed by a PGP private key, then anyone with the corresponding public key can verify the authenticity of the text.

19.3. Interaction with External Routing Domains

This protocol does, through the use of TC messages, provide a basic mechanism for injecting external routing information to this protocol's domain. Routing information can be extracted from the protocol's Information Bases, in particular the Routing Set, of this protocol and, potentially, injected into an external domain, if the routing protocol governing that domain permits this.

When operating routers connecting a MANET using this protocol to an external routing domain, care MUST be taken not to allow potentially

insecure and untrustworthy information to be injected from this domain to external routing domains. Care MUST also be taken to validate the correctness of information prior to it being injected as to avoid polluting routing tables with invalid information.

A recommended way of extending connectivity from an existing routing domain to a MANET routed using this protocol is to assign an IP prefix (under the authority of the routers/gateways connecting the MANET with the exiting routing domain) exclusively to that MANET area, and to statically configure the gateways to advertise routes for that IP sequence to routers in the existing routing domain.

20. IANA Considerations

This specification defines one Message Type, which must be allocated from the "Message Types" repository of [RFC5444], two Message TLV Types, which must be allocated from the "Message TLV Types" repository of [RFC5444], and three Address Block TLV Types, which must be allocated from the "Address Block TLV Types" repository of [RFC5444].

20.1. 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].

20.2. Message Types

This specification defines one Message Type, to be allocated from the 0-223 range of the "Message Types" namespace defined in [RFC5444], as specified in Table 6.

Name	Type	Description
TC	TBD1	Topology Control (MANET-wide signaling)

Table 6: Message Type assignment

20.3. Message-Type-specific TLV Type Registries

IANA is requested to create a registry for Message-Type-specific Message TLVs for TC messages, in accordance with Section 6.2.1 of [RFC5444], and with initial assignments and allocation policies as specified in Table 7.

Type	Description	Allocation Policy
128-223	Unassigned	Expert Review

Table 7: TC Message-Type-specific Message TLV Types

IANA is requested to create a registry for Message-Type-specific Address Block TLVs for TC messages, in accordance with Section 6.2.1 of [RFC5444], and with initial assignments and allocation policies as specified in Table 8.

Type	Description	Allocation Policy
128-223	Unassigned	Expert Review

Table 8: TC Message-Type-specific Address Block TLV Types

20.4. Message TLV Types

This specification defines two Message TLV Types, which must be allocated from the "Message TLV Types" namespace defined in [RFC5444]. IANA are requested to make allocations in the 8-127 range for these types. This will create two new Type Extension registries with assignments as specified in Table 9 and Table 10. Specifications of these TLVs are in Section 8.1.1 and Section 8.2.1, respectively. Each of these TLVs MUST NOT be included more than once in a Message TLV Block.

Name	Type	Type Extension	Description
MPR_WILLING	TBD2	0	Specifies the originating router's willingness to act as a relay and to partake in network formation
Unassigned	TBD2	1-255	Expert Review

Table 9: Message TLV Type assignment: MPR_WILLING

Name	Type	Type Extension	Description
CONT_SEQ_NUM	TBD3	0 (COMPLETE)	Specifies a content sequence number for this complete message
CONT_SEQ_NUM	TBD3	1 (INCOMPLETE)	Specifies a content sequence number for this incomplete message
Unassigned	TBD3	2-255	Expert Review

Table 10: Message TLV Type assignment: CONT_SEQ_NUM

Type extensions indicated as Expert Review SHOULD be allocated as described in [RFC5444], based on Expert Review as defined in [RFC5226].

20.5. Address Block TLV Types

This specification defines three Address Block TLV Types, which must be allocated from the "Address Block TLV Types" namespace defined in [RFC5444]. IANA are requested to make allocations in the 8-127 range for these types. This will create three new Type Extension registries with assignments as specified in Table 11, Table 12 and Table 13, respectively. Specifications of these TLVs are in Section 8.1.2 and Section 8.2.2.

Name	Type	Type Extension	Description
MPR	TBD4	0	Specifies that a given address is of a router selected as an MPR
Unassigned	TBD4	1-255	Expert Review

Table 11: Address Block TLV Type assignment: MPR

Name	Type	Type Extension	Description
NBR_ADDR_TYPE	TBD5	0	Specifies that a given address is of a neighbor reached via the originating router
Unassigned	TBD5	1-255	Expert Review

Table 12: Address Block TLV Type assignment: NBR_ADDR_TYPE

The Values which the NBR_ADDR_TYPE Address Block TLV can use are the following:

- ORIGINATOR := 1;
- ROUTABLE := 2;
- ROUTABLE_ORIG := 3.

Name	Type	Type extension	Description
GATEWAY	TBD6	0	Specifies that a given address is reached via a gateway on the originating router
Unassigned	TBD6	1-255	Expert Review

Table 13: Address Block TLV Type assignment: GATEWAY

Type extensions indicated as Expert Review SHOULD be allocated as described in [RFC5444], based on Expert Review as defined in [RFC5226].

21. Contributors

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

- Cedric Adjih, INRIA, France, <Cedric.Adjih@inria.fr>
- Emmanuel Baccelli, INRIA , France, <Emmanuel.Baccelli@inria.fr>
- Thomas Heide Clausen, LIX, France, <T.Clausen@computer.org>

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

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

o Satoh Hiroki, Hitachi SDL, Japan, <hiroki.satoh.yj@hitachi.com>

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

o Monden Kazuya, Hitachi SDL, Japan, <kazuya.monden.vw@hitachi.com>

o Kenichi Mase, Niigata University, Japan, <mase@ie.niigata-u.ac.jp>

o Ryuji Wakikawa, Toyota, Japan, <ryuji@sfc.wide.ad.jp>

22. Acknowledgments

The authors would like to acknowledge the team behind OLSRv1, specified in RFC3626, including Anis Laouiti (INT, Paris), Pascale Minet (INRIA, France), Laurent Viennot (INRIA, France), and Amir Qayyum (M.A. Jinnah University, Islamabad) 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): Khaldoun Al Agha (LRI), Teco Boot (Infinity Networks), Song-Yean Cho (LIX), Alan Cullen (BAE Systems), Louise Lamont (CRC), Li Li (CRC), Joe Macker (NRL), Richard Ogier (SRI), Charles E. Perkins (WiChorus), Henning Rogge (FGAN), and the entire IETF MANET working group.

23. References

23.1. Normative References

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

[RFC5148] Clausen, T., Dearlove, C., and B. Adamson, "Jitter considerations in MANETs", RFC 5148, February 2008.

[RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", RFC 5226, BCP 26, May 2008.

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

[RFC5497] Clausen, T. and C. Dearlove, "Representing multi-value time in MANETs", RFC 5497, March 2009.

[RFC5498] Chakeres, I., "IANA Allocations for MANET Protocols", RFC 5498, March 2009.

[NHDP] Clausen, T., Dean, J., and C. Dearlove, "MANET Neighborhood Discovery Protocol (NHDP)", work in progress draft-ietf-manet-nhdp-10.txt, July 2009.

23.2. Informative References

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

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

[RFC4880] Callas, J., Donnerhacke, L., Finney, H., and R. Thayer, "OpenPGP message format", RFC 4880, November 2007.

[HIPERLAN] ETSI, "ETSI STC-RES10 Committee. Radio equipment and systems: HIPERLAN type 1, functional specifications ETS 300-652", June 1996.

[HIPERLAN2] Jacquet, P., Minet, P., Muhlethaler, P., and N. Rivierre, "Increasing reliability in cable free radio LANs: Low level forwarding in HIPERLAN.", 1996.

[MPR] Qayyum, A., Viennot, L., and A. Laouiti, "Multipoint relaying: An efficient technique for flooding in mobile wireless networks.", 2001.

[FSR] Pei, G., Gerla, M., and T. Chen, "Fisheye state routing in mobile ad hoc networks", 2000.

[FSLS] Santivanez, C., Ramanathan, R., and I. Stavrakakis, "Making link-state routing scale for ad hoc networks", 2000.

Appendix A. Example Algorithm for Calculating MPRs

The following specifies an algorithm which MAY be used to select MPRs. MPRs are calculated per OLSRv2 interface, but then a single set of MPRs is formed from the union of the MPRs for all OLSRv2 interfaces. (As noted in Section 14 a router MAY improve on this, by coordination between OLSRv2 interfaces.) A router's MPRs are

recorded using the element N_mpr in Neighbor Tuples.

If using this example algorithm then the following steps MUST be executed in order for a router to select its MPRs:

1. Set N_mpr := false in all Neighbor Tuples;

2. For each Neighbor Tuple with N_symmetric = true and N_willingness = WILL_ALWAYS, set N_mpr := true;

3. For each OLSRv2 interface of the router, use the algorithm in Appendix A.2. Note that this sets N_mpr := true for some Neighbor Tuples, these routers are already selected as MPRs when using the algorithm for following OLSRv2 interfaces.

4. OPTIONALLY, consider each selected MPR in turn, and if the set of selected MPRs without that router still satisfies the necessary conditions, for all OLSRv2 interfaces, then that router MAY be removed from the set of MPRs. This process MAY be repeated until no MPRs are removed. Routers MAY be considered in order of increasing N_willingness.

Note that only symmetric strict 2-hop neighbors are considered, thus:

- Symmetric 1-hop neighbor routers with N_willingness = WILL_NEVER MUST NOT be selected as MPRs, and MUST be ignored in the following algorithm (and hence also ignore any 2-Hop Tuples whose N2_neighbor_iface_addr_list is included in the N_neighbor_addr_list of any such Neighbor Tuple).

- Symmetric 2-hop neighbor routers which are also symmetric 1-hop neighbor routers MUST be ignored in the following algorithm (i.e. ignore any 2-Hop Tuples whose N2_2hop_addr is in the N_neighbor_addr_list of any Neighbor Tuple).

A.1. Terminology

The following terminology will be used when selecting MPRs for the OLSRv2 interface I:

N(I) - The set of symmetric 1-hop neighbors which have a symmetric link to I.

N2(I) - The set of addresses of interfaces of a router with a symmetric link to a router in N(I); this MAY be restricted to considering only information received over I (in which case N2(I) is the set of N2_2hop_addr in 2-Hop Tuples in the 2-Hop Set for OLSRv2 interface I).

Connected to I via Y - An address A in N2(I) is connected to I via a router Y in N(I) if A is an address of an interface of a symmetric 1-hop neighbor of Y (i.e. A is the N2_2hop_addr in a 2-Hop Tuple in the 2-Hop Set for OLSRv2 interface I, and whose N2_neighbor_iface_addr_list is contained in the set of interface addresses of Y).

D(Y, I) - For a router Y in N(I), the number of addresses in N2(I) which are connected to I via Y.

R(Y, I): - For a router Y in N(I), the number of addresses in N2(I) which are connected to I via Y, but are not connected to I via any router which has already been selected as an MPR.

A.2. MPR Selection Algorithm for each OLSRv2 Interface

When selecting MPRs for the OLSRv2 interface I:

1. For each address A in N2(I) for which there is only one router Y in N(I) such that A is connected to I via Y, select that router Y as an MPR (i.e. set N_mpr := true in the Neighbor Tuple corresponding to Y).

2. While there exists any router Y in N(I) with R(Y, I) > 0:

   1. Select a router Y in N(I) with R(Y, I) > 0 in the following order of priority:

      + greatest N_willingness in the Neighbor Tuple corresponding to Y, THEN;

      + greatest R(Y, I), THEN;

      + greatest D(Y, I), THEN;

      + N_mpr_selector is equal to true, if possible, THEN;

      + any choice.

   2. Select Y as an MPR (i.e. set N_mpr := true in the Neighbor Tuple corresponding to Y).

Appendix B. Example Algorithm for Calculating the Routing Set

The following procedure is given as an example for calculating the Routing Set using a variation of Dijkstra's algorithm. First all Routing Tuples are removed, and then, using the selections and definitions in Appendix B.1, the procedures in the following sections

(each considered a "stage" of the processing) are applied in turn.

B.1. Local Interfaces and Neighbors

The following selections and definitions are made:

1. For each Local Interface Tuple, select an address from its I_local_iface_addr_list, this is defined as the selected address for this Local Interface Tuple.

2. For each Link Tuple, the selected address of its corresponding Local Interface Tuple is defined as the selected local address for this Local Interface Tuple.

3. For each Neighbor Tuple with N_symmetric = true, the selected local address is defined as the selected local address of the selected Link Tuple for that Neighbor Tuple.

4. For each address (N_orig_addr or in N_neighbor_addr_list, the "neighbor address") from a Neighbor Tuple with N_symmetric = true, select a Link Tuple with L_status = SYMMETRIC whose corresponding Neighbor Tuple is this Neighbor Tuple and where, if possible, L_neighbor_iface_addr_list contains the neighbor address. This is defined as the selected Link Tuple for that neighbor address.

5. For each address (N_orig_addr or in N_neighbor_addr_list, the "neighbor address") from a Neighbor Tuple with N_symmetric = true, a selected address from the L_neighbor_iface_addr_list of the selected Link Tuple for the neighbor address, if possible equal to the neighbor address, is defined as the selected link address for that neighbor address.

6. Routing Tuple preference is decided by preference for corresponding Neighbor Tuples in this order:

   * For greater N_willingness.

   * For N_mpr_selector = true over N_mpr_selector = false.

B.2. Add Neighbor Routers

The following procedure is executed once.

1. For each Neighbor Tuple with N_symmetric = true, add a Routing Tuple with:

* R_dest_addr := N_orig_addr;

* R_next_iface_addr := selected link address;

* R_local_iface_addr := selected local address;

* R_dist := 1.

B.3. Add Remote Routers

The following procedure is executed for each value of h, starting with h := 1 and incrementing by 1 for each iteration. The execution MUST stop if no new Routing Tuples are added in an iteration.

1. For each Router Topology Tuple, if:

   * TR_to_orig_addr is not equal to the R_dest_addr of any Routing Tuple added in an earlier stage, AND;

   * TR_from_orig_addr is equal to the R_dest_addr of a Routing Tuple with R_dist = h (the "previous Routing Tuple"),

   then add a new Routing Tuple, with:

   * R_dest_addr := TR_to_orig_addr;

   * R_next_iface_addr := R_next_iface_addr of the previous Routing Tuple;

   * R_local_iface_addr := R_local_iface_addr of the previous Routing Tuple;

   * R_dist := h+1.

   There may be more than one possible Routing Tuple that may be added for an R_dest_addr in this stage. If so, then, for each such R_dest_addr, a Routing Tuple which is preferred SHOULD be added.

B.4. Add Neighbor Addresses

The following procedure is executed once.

1. For each Neighbor Tuple with N_symmetric = true:

   1. For each address (the "current address") in N_neighbor_addr_list, if the current address is not equal to the R_dest_addr of any Routing Tuple, then add a new Routing

Tuple, with:

+ R_dest_addr := current address;

+ R_next_iface_addr := selected link address;

+ R_local_iface_addr := selected local address;

+ R_dist := 1.

B.5. Add Remote Routable Addresses

The following procedure is executed once.

1. For each Routable Address Topology Tuple, if:

   * TA_dest_addr is not equal to the R_dest_addr of any Routing Tuple added in an earlier stage, AND;

   * TR_from_orig_addr is equal to the R_dest_addr of a Routing Tuple (the "previous Routing Tuple"),

   then add a new Routing Tuple, with:

   * R_dest_addr := TA_dest_addr;

   * R_next_iface_addr := R_next_iface_addr of the previous Routing Tuple;

   * R_local_iface_addr := R_local_iface_addr of the previous Routing Tuple;

   * R_dist := R_dist of the previous Routing Tuple + 1.

   There may be more than one possible Routing Tuple that may be added for an R_dest_addr in this stage. If so, then, for each such R_dest_addr, a Routing Tuple which is preferred SHOULD be added.

B.6. Add Attached Networks

The following procedure is executed once.

1. For each Attached Network Tuple, if:

   * AN_orig_addr is not equal to the R_dest_addr of any Routing Tuple added in an earlier stage, AND;

* AN_orig_addr is equal to the R_dest_addr of a Routing Tuple (the "previous Routing Tuple),

then add a new Routing Tuple, with:

* R_dest_addr := AN_net_addr;

* R_next_iface_addr := R_next_iface_addr of the previous Routing Tuple;

* R_local_iface_addr := R_local_iface_addr of the previous Routing Tuple;

* R_dist := R_dist of the previous Routing Tuple + AN_dist.

There may be more than one possible Routing Tuple that may be added for an R_dest_addr in this stage. If so, then, for each such R_dest_addr, a Routing Tuple with minimum R_dist MUST be selected, otherwise a Routing Tuple which is preferred SHOULD be added.

B.7. Add 2-Hop Neighbors

The following procedure is executed once.

1. For each 2-Hop Tuple, if:

   * N2_2hop_addr is a routable address, AND;

   * N2_2hop_addr is not equal to the R_dest_addr of any Routing Tuple added in an earlier stage,

   then define the "previous Routing Tuple" as that with R_dest_addr = N_orig_addr of the corresponding Neighbor Tuple, and add a new Routing Tuple, with:

   * R_dest_addr := N2_2hop_addr;

   * R_next_iface_addr := R_next_iface_addr of the previous Routing Tuple;

   * R_local_iface_addr := R_local_iface_addr of the previous Routing Tuple;

   * R_dist := 2.

   There may be more than one possible Routing Tuple that may be added for an R_dest_addr in this stage. If so, then, for each

such R_dest_addr, a Routing Tuple which is preferred SHOULD be added.

Appendix C. Example Message Layout

An example TC message is as follows. The message has full Message Header (four bit Flags field value is 15). Its four bit Message Address Length field has value 3 and hence addresses in the message have length four octets, here being IPv4 addresses. The overall message length is 57 octets.

The message has a Message TLV Block with content length 13 octets containing three TLVs. The first two TLVs are interval and validity times for the message. The third TLV is the content sequence number TLV used to carry the 2 octet ANSN, and (with default type extension zero, i.e. COMPLETE) indicating that the TC message is complete. Each TLV uses a TLV with Flags octet value 16, indicating that it has a Value, but no type extension or start and stop indexes. The first two TLVs have a Value Length of 1 octet, the last has a Value Length of 2 octets.

The message has two Address Blocks. (This is not necessary, the information could be conveyed using a single Address Block, the use of two Address Blocks, which is also allowed, is illustrative only.) The first Address Block contains 3 addresses, with Flags octet value 128, hence with a Head section, (with length 2 octets) but no Tail section, and hence Mid sections with length two octets. The following TLV Block (content length 6 octets) contains a single NBR_ADDR_TYPE TLV (Flags octet value 16, includes a Value but no indexes) indicating that these addresses are associated with the Value (with Value Length 1 octet) ROUTABLE_ORIG, i.e. they are originator addresses of advertised neighbors that are also routable addresses.

The second Address Block contains 1 address, with Flags octet 176 indicating that there is a Head section (with length 2 octets), that the Tail section (length 2 octets) consists of zero valued octets (not included), and that there is a single prefix length, which is 16. The network address is thus Head.0.0/16. The following TLV Block (content length 8 octets) includes one TLV that indicates that the originating router is a gateway to this network, at a given number of hops distance (Value Length 1 octet). The TLV Flags octet value of 16 again indicates that a Value, but no indexes are needed.

```
     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
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |      TC       |1 1 1 1 0 0 1 1|0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 1|
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                      Originator Address                       |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |   Hop Limit   |   Hop Count   |    Message Sequence Number    |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 1| INTERVAL_TIME |0 0 0 1 0 0 0 0|
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |0 0 0 0 0 0 0 1|     Value     | VALIDITY_TIME |0 0 0 1 0 0 0 0|
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |0 0 0 0 0 0 0 1|     Value     | CONT_SEQ_NUM  |0 0 0 1 0 0 0 0|
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |0 0 0 0 0 0 1 0|         Value (ANSN)          |0 0 0 0 0 0 1 1|
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |1 0 0 0 0 0 0 0|0 0 0 0 0 0 1 0|             Head              |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |              Mid              |              Mid              |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |              Mid              |0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0|
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    | NBR_ADDR_TYPE |0 0 0 1 0 0 0 0|0 0 0 0 0 0 0 1| ROUTABLE_ORIG |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |0 0 0 0 0 0 0 1|1 0 1 1 0 0 0 0|0 0 0 0 0 0 1 0|     Head      |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |  Head (cont)  |0 0 0 0 0 0 1 0|0 0 0 1 0 0 0 0|0 0 0 0 0 0 0 0|
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |0 0 0 0 0 1 0 0|    GATEWAY    |0 0 0 1 0 0 0 0|0 0 0 0 0 0 0 1|
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |  Number Hops  |
    +-+-+-+-+-+-+-+-+
```

Appendix D. Constraints

Any process which updates the Local Information Base, the Neighborhood Information Base or the Topology Information Base MUST ensure that all constraints specified in this appendix are maintained, as well as those specified in [NHDP].

In each Originator Tuple:

- O_orig_addr MUST NOT equal any other O_orig_addr.

- O_orig_addr MUST NOT equal this router's originator address.

In each Local Attached Network Tuple:

- o AL_net_addr MUST NOT equal any other AL_net_addr.

- o AL_net_addr MUST NOT be in the I_local_iface_addr_list of any Local Interface Tuple or equal the IR_local_iface_addr of any Removed Interface Address Tuple.

- o AL_net_addr MUST not equal this router's originator address, or equal the O_orig_addr in any Originator Tuple.

- o AL_dist MUST NOT be less than zero.

In each Link Tuple:

- o L_neighbor_iface_addr_list MUST NOT contain the AL_net_addr of any Local Attached Network Tuple.

In each Neighbor Tuple:

- o N_orig_addr MUST NOT be changed to unknown.

- o N_orig_addr MUST NOT equal this router's originator address, or equal O_orig_addr in any Originator Tuple.

- o N_orig_addr MUST NOT equal the AL_net_addr in any Local Attached Network Tuple.

- o N_neighbor_addr_list MUST NOT contain this router's originator address, the O_orig_addr in any Originator Tuple, or the AL_net_addr in any Local Attached Network Tuple.

- o If N_orig_addr = unknown, then N_willingness = WILL_NEVER, N_mpr = false, N_mpr_selector = false, and N_advertised = false.

- o If N_willingness MUST be in the range from WILL_NEVER to WILL_ALWAYS, inclusive.

- o If N_mpr = true, then N_symmetric MUST be true and N_willingness MUST NOT equal WILL_NEVER.

- o If N_symmetric = true and N_mpr = false, then N_willingness MUST NOT equal WILL_ALWAYS.

- o If N_mpr_selector = true, then N_symmetric MUST be true and N_advertised MUST be true.

- o If N_advertised = true, then N_symmetric MUST be true.

In each Lost Neighbor Tuple:

o NL_neighbor_addr MUST NOT equal this router's originator address, equal the O_orig_addr in any Originator Tuple, or equal the AL_net_addr in any Local Attached Network Tuple.

In each 2-Hop Tuple:

o N2_2hop_addr MUST NOT equal this router's originator address, equal the O_orig_addr in any Originator Tuple, or equal the AL_net_addr in any Local Attached Network Tuple.

In each Advertising Remote Router Tuple:

o AR_orig_addr MUST NOT be in the I_local_iface_addr_list in any Local Interface Tuple or equal the IR_local_iface_addr in any Removed Interface Address Tuple.

o AR_orig_addr MUST NOT equal this router's originator address or equal the O_orig_addr in any Originator Tuple.

o AR_orig_addr MUST NOT equal the AL_net_addr in any Local Attached Network Tuple.

o AR_orig_addr MUST NOT equal the AR_orig_addr in any other Advertising Remote Router Tuple.

In each Router Topology Tuple:

o There MUST be an Advertising Remote Router Tuple with AR_orig_addr = TR_from_orig_addr.

o TR_to_orig_addr MUST NOT be in the I_local_iface_addr_list in any Local Interface Tuple or equal the IR_local_iface_addr in any Removed Interface Address Tuple.

o TR_to_orig_addr MUST NOT equal this router's originator address or equal the O_orig_addr in any Originator Tuple.

o TR_to_orig_addr MUST NOT equal the AL_net_addr in any Local Attached Network Tuple.

o The ordered pair (TR_from_orig_addr, TR_to_orig_addr) MUST NOT equal the corresponding pair for any other Router Topology Tuple.

o TR_seq_number MUST NOT be greater than AR_seq_number in the Advertising Remote Router Tuple with AR_orig_addr = TR_from_orig_addr.

In each Routable Address Topology Tuple:

- o There MUST be an Advertising Remote Router Tuple with AR_orig_addr = TA_from_orig_addr.

- o TA_dest_addr MUST be routable.

- o TA_dest_addr MUST NOT be in the I_local_iface_addr_list in any Local Interface Tuple or equal the IR_local_iface_addr in any Removed Interface Address Tuple.

- o TA_dest_addr MUST NOT equal this router's originator address or equal the O_orig_addr in any Originator Tuple.

- o TA_dest_addr MUST NOT equal the AL_net_addr in any Local Attached Network Tuple.

- o The ordered pair (TA_from_orig_addr, TA_dest_addr) MUST NOT equal the corresponding pair for any other Attached Network Tuple.

- o TA_seq_number MUST NOT be greater than AR_seq_number in the Advertising Remote Router Tuple with AR_orig_addr = TA_from_orig_addr.

In each Attached Network Tuple:

- o There MUST be an Advertising Remote Router Tuple with AR_orig_addr = AN_orig_addr.

- o AN_net_addr MUST NOT be in the I_local_iface_addr_list in any Local Interface Tuple or equal the IR_local_iface_addr in any Removed Interface Address Tuple.

- o AN_net_addr MUST NOT equal this router's originator address or equal the O_orig_addr in any Originator Tuple.

- o AN_net_addr MUST NOT equal the AL_net_addr in any Local Attached Network Tuple.

- o The ordered pair (AN_orig_addr, AN_net_addr) MUST NOT equal the corresponding pair for any other Attached Network Tuple.

- o AN_seq_number MUST NOT be greater than AR_seq_number in the Advertising Remote Router Tuple with AR_orig_addr = AN_orig_addr.

- o AN_dist MUST NOT be less than zero.

In each Received Tuple:

- o RX_orig_addr MUST NOT equal this router's originator address or the O_orig_addr in any Originator Tuple.

- o Each ordered triple (RX_type, RX_orig_addr, RX_seq_number) MUST NOT equal the corresponding triple for any other Received Tuple in the same Received Set.

In each Processed Tuple:

- o P_orig_addr MUST NOT equal this router's originator address or equal the O_orig_addr in any Originator Tuple.

- o Each ordered triple (P_type, P_orig_addr, P_seq_number) MUST NOT equal the corresponding triple for any other Processed Tuple.

In each Forwarded Tuple:

- o F_orig_addr MUST NOT equal this router's originator address or equal the O_orig_addr in any Originator Tuple.

- o Each ordered triple (F_type, F_orig_addr, F_seq_number) MUST NOT equal the corresponding triple for any other Forwarded Tuple.

Appendix E. Flow and Congestion Control

Due to its proactive nature, this protocol has a natural control over the flow of its control traffic. Routers transmit control messages at predetermined rates specified and bounded by message intervals.

This protocol employs [NHDP] for local signaling, embedding MPR selection advertisement through a simple Address Block TLV, and router willingness advertisement (if any) as a single Message TLV. Local signaling, therefore, shares the characteristics and constraints of [NHDP].

Furthermore, the use of MPRs can greatly reduce the signaling overhead from link state information dissemination in two ways, attaining both flooding reduction and topology reduction. First, using MPR flooding, the cost of distributing link state information throughout the network is reduced, as compared to when using classic flooding, since only MPRs need to forward link state declaration messages. Second, the amount of link state information for a router to declare is reduced to need only contain that router's MPR selectors. This reduces the size of a link state declaration as compared to declaring full link state information. In particular some routers may not need to declare any such information. In dense networks, the reduction of control traffic can be of several orders of magnitude compared to routing protocols using classical flooding

[MPR]. This feature naturally provides more bandwidth for useful data traffic and pushes further the frontier of congestion.

Since the control traffic is continuous and periodic, it keeps the quality of the links used in routing more stable. However, using some options, some control messages (HELLO messages or TC messages) may be intentionally sent in advance of their deadline in order to increase the responsiveness of the protocol to topology changes. This may cause a small, temporary, and local increase of control traffic, however this is at all times bounded by the use of minimum message intervals.

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/

Philippe Jacquet
Project Hipercom, INRIA

Phone: +33 1 3963 5263
EMail: philippe.jacquet@inria.fr

The OLSRv2 Design Team
MANET Working Group