

Mobile Ad hoc Networking (MANET)  
Internet-Draft  
Updates: 7182 (if approved)  
Intended status: Standards Track  
Expires: December 28, 2014

C. Dearlove  
BAE Systems ATC  
June 26, 2014

Identity-Based Signatures for MANET Routing Protocols  
draft-dearlove-manet-ibs-00

Abstract

This document extends [RFC7182], which specifies a framework for, and specific examples of, integrity check values (ICVs) for packets and messages using the generalized packet/message format specified in [RFC5444]. It does so by defining an additional cryptographic function that allows the creation of an ICV that is an identity-based signature, defined according to the ECCSI (Elliptic Curve-Based Certificateless Signatures for Identity-Based Encryption) algorithm specified in [RFC6507].

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on December 28, 2014.

Copyright Notice

Copyright (c) 2014 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect

to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

1. Introduction . . . . .	3
2. Terminology . . . . .	5
3. Applicability Statement . . . . .	5
4. Specification . . . . .	5
4.1. Cryptographic Function . . . . .	5
4.2. ECCSI parameters . . . . .	6
4.3. Identity . . . . .	7
5. IANA Considerations . . . . .	7
6. Security Considerations . . . . .	8
7. Acknowledgments . . . . .	8
8. References . . . . .	9
8.1. Normative References . . . . .	9
8.2. Informative References . . . . .	9
Appendix A. Example . . . . .	9
Author's Address . . . . .	9

## 1. Introduction

[RFC7182] defines ICV (integrity check value) TLVs for use in packets and messages that use the generalized MANET packet/message format defined in [RFC5444]. This specification extends the TLV definitions therein by defining two new cryptographic function code points that allow the use of an identity-based signature (IBS) as an ICV. An IBS has an additional property that is not shared by any of the previously specified ICVs, it not only indicates that the protected packet or message is valid, but also verifies the originator of the packet/message.

This specification assumes that each router (protocol participant) has an identity that may be tied to the packet or message. The router may have more than one identity, but will only use one for each ICV TLV. The cryptographic strength of the IBS is not dependent on the choice of identity.

Two options for the choice of identity are supported (the two code points allocated). In the first the identity can be any octet sequence (up to 255 octets) included in the ICV TLV. In the second, the octet sequence is preceded by an address, either the IP source address for a packet TLV, or the message originator address for a message or address block TLV. In particular, the second option allows just the address to be used as an identity.

Identity-based signatures, compared to the shared secret key ICVs specified in [RFC7182], allow identifying the originator of information in a packet or message. They thus allow additional security functions, such as revocation of an identity, and removing all information with a specific originator, if this is recorded - as it is for OLSRv2 [RFC7181], an expected user of this specification. When applied to messages (rather than packets) this can significantly reduce the damage that a compromised router can inflict on the network.

Identity-based signatures are based on forms of asymmetric (public key) cryptography - identity-based encryption (IBE). In terms of their use, IBE and IBS methods have a major advantage, and a major disadvantage, compared to more widely used public key cryptography solutions, such as RSA.

The advantage referred to is that each router can be configured once (for its key lifetime) by a trusted authority, independently of all other routers. Thus router A can connect to the authority (typically in a secure environment) to receive a private key, or can have a private key delivered securely (out of band) from the authority. During normal operation of the MANET, there is no need for the

trusted authority to be connected to the MANET, or even to still exist. Additional routers can be authorized, with no reference to previously authorized routers (the trusted authority must still exist in this case). A router's public key is its identity, which when tied to a packet or message (as is the case when using an address as, or as part of, the identity) means that there is no need for public key certificates or a certificate authority.

The disadvantage referred to is that the trusted authority has complete authority, even more so than a conventional certificate authority. Routers cannot generate their own private keys, only the trusted authority can do that. Through the master secret held by the trusted authority, it could impersonate any router (existing or not). When used for identity-based encryption (not part of this specification) the trusted authority can decrypt anything. However, note that the shared secret key options described in [RFC7182] also have this limitation.

There are alternative mathematical realizations of identity-based signatures. This specification uses one that has been previously published as [RFC6507], known as ECCSI (Elliptic Curve-Based Certificateless Signatures for Identity-Based Encryption). In common with other identity-based encryption/signature approaches, it is based on the use of elliptic curves. Unlike some, it does not use "pairings" (bilinear maps from a product of two elliptic curve groups to another group). It thus may be easier to implement, and more efficient, than some alternatives, although with a greater signature size than some. This specification allows the use of any elliptic curve that may be used by [RFC6507].

The computational load imposed by ECCSI (and, perhaps more so, other IBS methods) is not trivial, though depending significantly on the quality of implementation of the required elliptic curve and other mathematical functions. For a security level of 128 bits, the ICV data length is 129 octets, which is longer than for alternative ICVs specified in [RFC7182] (e.g., 32 octets for the similar strength HMAC-SHA-256). The signature format used could have been slightly shortened (to 97 octets) by using a compressed representation of an elliptic curve point, however at the expense of some additional work when verifying a signature, and loss of direct compatibility with [RFC6507], and implementations thereof.

The trusted authority is referred to in [RFC6507] as the KMS (Key Management Service). That term will be used in the rest of this specification.

## 2. Terminology

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

Additionally, this document uses the terminology of [RFC5444], [RFC6507], and [RFC7182].

## 3. Applicability Statement

This specification adds an additional option to the framework specified in [RFC7182] for use by [RFC5444] formatted packets and messages. It is applicable as described in [RFC7182], and subject to the additional comments in Section 6.

Specific examples of protocols for which this specification is suitable are NHDP [RFC6130] and OLSRv2 [RFC7181].

## 4. Specification

### 4.1. Cryptographic Function

This specification defines a cryptographic function named ECCSI that is implemented as specified as the "sign" function in Section 5.2.1 of [RFC6507]. To use that specification:

- o The ICV is not calculated as `cryptographic-function(hash-function(content))` as defined in [RFC7182], but (like the HMAC ICVs defined there) uses the hash function within the cryptographic function. The option "none" is not permitted for hash-function, and the hash function must have a known fixed length of N octets, as specified in Section 4.2.
- o M in [RFC6507] is "content" as specified in in [RFC7182].
- o ID, used in [RFC6507], is as specified in Section 4.3.
- o KPAK, SSK and PVT, used in [RFC6507], are as specified in Sections 4.2 and 5.1.1 of [RFC6507], provided by the KMS.

The length of the signature is  $4N+1$  octets, as specified in [RFC6507], whose affine coordinate format (including an octet valued 0x04 to identify this) is used unchanged.

Verification of the ICV is not implemented by the receiver recalculating the ICV and comparing with the received ICV, as it is necessarily incapable of doing so. Instead the receiver evaluates the "verify" function described in Section 5.2.2 of [RFC6507], which may pass or fail.

To use that function M, KPAK, SSK and PVT are as specified above, while ID is deduced from the received packet or message, as specified in Section 4.3, using the <key-id> element in the <ICV-value>. This element need not match that used by the receiver, and thus when using this cryptographic function, multiple ICV TLVs differing only in their <key-id>, or in the choice of cryptographic function from the two defined in this specification, SHOULD NOT be used unless routers are administratively configured to recognize which to verify.

Routers MAY be administratively configured to reject a packet or message ICV TLV using ECCSI based on part or all of <key-id>; for example if this encodes a time after which this identity is no longer valid.

#### 4.2. ECCSI parameters

Section 4.1 of [RFC6507] specifies parameters  $n$ ,  $N$ ,  $p$ ,  $E$ ,  $B$ ,  $G$ , and  $q$ . The first of these,  $n$ , is specified as "A security parameter; the size in bits of the prime  $p$  over which elliptic curve cryptography is to be performed." For typical security levels (e.g., 128, 192 and 256 bits),  $n$  must be at least twice the required bits of security, see Section 5.6.1 of [NIST-SP-800-57].

Selection of an elliptic curve, and all related parameters, MUST be by administrative means, and known to all routers. This specification follows [RFC6507] with a RECOMMENDED selection to follow Appendix D.1.2 of [NIST-FIPS-186-4]. (Note that  $n$  in that document is  $q$  in [RFC6507].)

The parameter that is required by this specification is  $N$ , which is defined as  $\text{Ceiling}(n/8)$ . The hash function used must create an output of size  $N$  octets. In particular for 128 bit security, and hence  $n = 256$ ,  $N = 32$ , and the RECOMMENDED hash function is SHA-256. The signature (i.e. <ICV-data>) length is  $4N + 1$  octets, i.e., 129 octets for  $N = 32$ .

Note: [RFC6507] actually refers to the predecessor to [NIST-FIPS-186-4], but the latest version is specified here; there are no significant differences in this regard.

### 4.3. Identity

There are two options for the identity ID used by [RFC6507], which are indicated by there being two code points allocated for this cryptographic function, see Section 5.

- o For the cryptographic function ECCSI ID is the element <key-id> defined in Section 12.1 of [RFC7182]. This MUST NOT be empty.
- o For the cryptographic function ECCSI-ADDR, ID is the concatenation of an address (in network byte order) and the element <key-id> defined in Section 12.1 of [RFC7182], where the latter MAY be empty. For a packet TLV this address is the IP source address of the IP datagram in which this packet is included. For a message TLV or an address block TLV this address is the message originator address (the element <msg-orig-addr> defined in [RFC5444]) if that address is present, if not present and the message is known to have travelled only one hop, then the IP source address of the IP datagram in which this message is included is used, otherwise no address is defined and the message MUST be rejected. (Note that HELLO messages specified in NHDP [RFC6130] and used in OLSRv2 [RFC7181] always only travel one hop, and hence their IP source address SHOULD be used if no originator address is present.)

Note that this identity is formatted by [RFC6507], and thus does not need a length field incorporated into it by this specification.

## 5. IANA Considerations

IANA has, in accordance with [RFC7182], defined a registry for the cryptographic functions. IANA is requested to modify this allocation as indicated.

Value	Algorithm	Description	Reference
7	ECCSI	ECCSI [RFC6507]	This specification
8	ECCSI-ADDR	ECCSI [RFC6507] with an address (source or originator) joined to identity	This specification
9-251		Unassigned; Expert Review	

Table 1: Cryptographic Function Registry

## 6. Security Considerations

This specification extends the security framework for MANET routing protocols specified in [RFC7182] by the addition of an additional cryptographic function, in two forms according to how identity is specified.

This cryptographic function implements a form of identity-based signature (IBS), a stronger form of integrity check value (ICV) that verifies not just that the received packet or message is valid but that the packet or message originated at a router that was assigned a private key for the specified identity.

For a message the identity is, and for a packet it is recommended that it is, either the originator address of the router (i.e., an address unique to that router), or the originator address with additional information appended. The use of that additional information is outside the scope of this specification, a typical use may be to indicate an expiry time for signatures created using that identity.

In common with other forms of IBS, a feature of the form of IBS (known as ECCSI) used in this specification is that it requires a trusted authority (KMS) that issues all private keys, and has complete cryptographic information about all possible private keys. However to set against that, the solution is scalable, as all routers can be independently keyed, and does not need the KMS in the network. If no future keys will be required, then the KMS's master secret can be destroyed. As routers are individually keyed, key revocation (by blacklist and time expiry of keys) is possible, but is beyond the scope of this specification.

ECCSI is based on elliptic curve mathematics. This specification follows [RFC6507] in its recommendation of elliptic curves, but any suitable (prime power) elliptic curve may be used; this must be administratively specified. Implementation of this specification will require an available implementation of suitable mathematical functions. Unlike some other forms of IBS, ECCSI requires only basic elliptic curve operations, it does not require "pairings" (bilinear functions of a product of two elliptic curve groups). This increases the available range of suitable mathematical libraries.

## 7. Acknowledgments

The author would like to thank his colleagues who have been involved in identity-based security for ad hoc networks, including (in alphabetical order) Alan Cullen, Peter Smith and Bill Williams.



## 8. References

### 8.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC5444] Clausen, T., Dearlove, C., Dean, J., and C. Adjih, "Generalized Mobile Ad Hoc Network (MANET) Packet/Message Format", RFC 5444, February 2009.
- [RFC6507] Groves, M., "Elliptic Curve-Based Certificateless Signatures for Identity-Based Encryption (ECCSI)", RFC 6507, February 2012.
- [RFC7182] Herberg, U., Clausen, T., and C. Dearlove, "Integrity Check Value and Timestamp TLV Definitions for Mobile Ad Hoc Networks (MANETs)", RFC 7182, April 2014.

### 8.2. Informative References

- [NIST-FIPS-186-4] National Institute of Standards and Technology, "Digital Signature Standard (DSS)", FIPS 186-4, July 2013.
- [NIST-SP-800-57] National Institute of Standards and Technology, "Recommendation for Key Management - Part 1: General (Revision 3)", SP 800-57, Part 1, Revision 3, July 2012.
- [RFC6130] Clausen, T., Dearlove, C., and J. Dean, "Mobile Ad Hoc Network (MANET) Neighborhood Discovery Protocol (NHDP)", RFC 6130, April 2011.
- [RFC7181] Clausen, T., Dearlove, C., Jacquet, P., and U. Herberg, "The Optimized Link State Routing Protocol Version 2", RFC 7181, April 2014.

## Appendix A. Example

TBD.

Author's Address

Christopher Dearlove  
BAE Systems Advanced Technology Centre  
West Hanningfield Road  
Great Baddow, Chelmsford  
United Kingdom

Phone: +44 1245 242194  
Email: [chris.dearlove@baesystems.com](mailto:chris.dearlove@baesystems.com)  
URI: <http://www.baesystems.com/>



Mobile Ad hoc Networking (MANET)  
Internet-Draft  
Updates: RFC 6130, RFC 7181  
(if approved)  
Intended status: Standards Track  
Expires: December 22, 2014

C. Dearlove  
BAE Systems ATC  
T. Clausen  
LIX, Ecole Polytechnique  
June 20, 2014

An Optimization for the MANET Neighborhood Discovery Protocol (NHDP)  
draft-dearlove-manet-nhdp-optimization-01

Abstract

The link quality mechanism of the MANET Neighborhood Discovery Protocol (NHDP) enables "ignoring" some 1-hop neighbors if the measured link quality from that 1-hop neighbor is below an acceptable threshold, while still retaining the corresponding link information as acquired from HELLO message exchange. This allows immediate reinstatement of the 1-hop neighbor if the link quality later improves sufficiently.

NHDP also collects information about symmetric 2-hop neighbors. However it specifies that if a link from a symmetric 1-hop neighbor ceases being symmetric, including while "ignored" as described above, then corresponding symmetric 2-hop neighbors are removed. This may lead to symmetric 2-hop neighborhood information being permanently removed (until further HELLO messages are received) if the link quality of a symmetric 1-hop neighbor drops below the acceptable threshold, even if only for a moment.

This specification updates NHDP, and the Optimized Link State Routing Protocol version 2 (OLSRv2) to permit retaining, but ignoring, symmetric 2-hop information when the link quality from the corresponding 1-hop neighbor drops below the acceptable threshold. This allows immediate reinstatement of the symmetric 2-hop neighbor if the link quality later improves sufficiently, thus making the symmetric 2-hop neighborhood more "robust".

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on December 22, 2014.

#### Copyright Notice

Copyright (c) 2014 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

1. Introduction . . . . .	4
2. Terminology . . . . .	5
3. Applicability Statement . . . . .	5
4. Changes to NHDP . . . . .	5
4.1. Interface Information Bases . . . . .	5
4.2. HELLO Message Processing . . . . .	6
4.3. Information Base Changes . . . . .	6
4.4. Constraints . . . . .	7
5. Changes to OLSRv2 . . . . .	7
6. MIB Considerations . . . . .	8
6.1. Updates to the State Group . . . . .	9
6.2. Updates to the Notification Group . . . . .	10
7. IANA Considerations . . . . .	11
8. Security Considerations . . . . .	11
9. Acknowledgments . . . . .	11
10. Normative References . . . . .	11
Authors' Addresses . . . . .	12

## 1. Introduction

The MANET Neighborhood Discovery Protocol (NHDP) [RFC6130], Section 14, contains a link admission mechanism known as "link quality" that allows a router using that protocol to "take considerations other than message exchange into account for determining when a link is and is not a candidate for being considered as HEARD or SYMMETRIC". Specifically, [RFC6130] permits a router to disallow consideration of some of its 1-hop neighbors, for as long as the quality of the link from that 1-hop neighbor is below an acceptable link quality threshold.

A feature of this mechanism is that while the link quality remains too low, the link information, established by the exchange of HELLO messages, is retained. Thus if the link quality later goes above the required threshold (note that a hysteresis mechanism means that two thresholds are used) then the link is immediately established and will be immediately available for use.

[RFC6130] collects not just 1-hop neighbor information, but also information about symmetric 2-hop neighbors. However [RFC6130] specifies that if a 1-hop neighbor was, but no longer is, considered symmetric, then the corresponding 2-Hop Tuples that may have been recorded for that 2-hop neighbor, are to be removed, without a retention mechanism for a (possibly temporary) loss due to link quality.

This means that if there is a short period in which link quality is too low, then when the link quality is reestablished, all 1-hop neighbor information is immediately available for use again. However, the corresponding symmetric 2-hop neighbor information has been removed, and is not available for use until restored by receipt of the next corresponding HELLO message.

This specification describes how [RFC6130] can be modified to avoid this situation, by retaining (but not using) 2-hop information, similar to what is done with 1-hop information. This modification is strictly optional, and routers that do and do not implement it can interwork entirely successfully (as they also can with different link quality specifications). In addition, by a suitable interpretation (that ignored 2-Hop Tuples are not externally advertised), this change can be invisible to any other protocols using [RFC6130], in particular [RFC7181]. However the impact on [RFC7181] when 2-Hop Tuples are not so handled is also described, in particular owing to the existence of implementations of that protocol that are not modularly separated from [RFC6130].

This specification therefore updates [RFC6130] and [RFC7181].

## 2. Terminology

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

Additionally, this document uses the terminology of [RFC6130] and [RFC7181].

## 3. Applicability Statement

This specification updates [RFC6130]. The optimization presented in this specification is simply permissive, as it allows retaining information which otherwise would have been removed, but does not use that information except when it could have been used by [RFC6130].

This can, in some cases, ensure that the symmetric 2-hop neighborhood is more robust against temporary link quality changes, and consequently yield a more stable network. The only other consequence of this optimization is that state for some otherwise expired 2-Hop Tuples may be maintained for longer.

This specification also updates [RFC7181]. This could be avoided by simply noting that this specification describes how the updates to [RFC6130] may be handled so as to be invisible to any other protocol using it. However as it is known that some implementations of [RFC7181] are not independent of the implementation of [RFC6130] that they use, it is useful to indicate the direct impact on [RFC7181].

A router that implements the optimization described in this specification will interoperate successfully with routers that implement [RFC6130], but do not implement this optimization.

## 4. Changes to NHDP

The following changes are made to [RFC6130] if using this specification. Note that while this specification is OPTIONAL, if any of these changes are made then all of these changes MUST be made.

### 4.1. Interface Information Bases

The 2-Hop Set is modified by adding this additional element to each 2-Hop Tuple:



N2\_lost is a boolean flag, which indicates the state of the corresponding Link Tuple. If L\_status = SYMMETRIC (and thus L\_lost = false), then N2\_lost = false. If L\_SYM\_time has not expired, and L\_lost = false (and hence L\_status = LOST), then N2\_lost = true.

In all other cases, including other cases with L\_status = LOST, there will be no such 2-Hop Tuples.

#### 4.2. HELLO Message Processing

In Section 12.6 of [RFC6130] make the following changes:

- o In point 2, change "L\_status = SYMMETRIC" to "L\_SYM\_time not expired".
- o When creating a 2-Hop Tuple, set N2\_lost := L\_lost.

#### 4.3. Information Base Changes

In Section 13, replace the second bullet point by:

- o A Link Tuple's L\_status changes from SYMMETRIC, L\_SYM\_time expires, or the Link Tuple is removed. In this case, the actions specified in Section 13.2 are performed.

and replace the paragraph after the bullet points by:

If a Link Tuple is removed, or if L\_HEARD\_time expires and either L\_status changes from SYMMETRIC or L\_SYM\_time expires, then the actions specified in Section 13.2 MUST be performed before the actions specified in Section 13.3 are performed for that Link Tuple.

In Section 13.2 of [RFC6130], add the following, before all other text:

For each Link Tuple that has L\_SYM\_time not expired:

1. If L\_SYM\_time then expires, or if the Link Tuple is removed:
  1. Remove each 2-Hop Tuple for the same MANET interface with:
    - + N2\_neighbor\_iface\_addr\_list contains one or more network addresses in L\_neighbor\_iface\_addr\_list.
2. If L\_status then changes from SYMMETRIC to LOST because L\_lost is set to true:

1. For each 2-Hop Tuple for the same MANET interface with:
  - + N2\_neighbor\_iface\_addr\_list contains one or more network addresses in L\_neighbor\_iface\_addr\_list;

set N2\_lost := true.

Also in Section 13.2 of [RFC6130], remove point 2, renumbering point 2 as point 1.

#### 4.4. Constraints

In Appendix B, under "In each 2-Hop Tuple:" change the first bullet point to:

- o There MUST be a Link Tuple associated with the same MANET interface with:
  - \* L\_neighbor\_iface\_addr\_list = N2\_neighbor\_iface\_addr\_list; AND
  - \* L\_SYM\_time not expired; AND
  - \* L\_lost = N2\_lost.

#### 5. Changes to OLSRv2

If the implementation of [RFC6130] conceals from any protocol using it the existence of all 2-Hop Tuples with N2\_lost = true, then no changes are required to any protocol using [RFC6130], in particular no changes are required to [RFC7181].

However if instead the implementation of [RFC6130] makes all 2-Hop Tuples visible, including those with N2\_lost = true, then protocols using [RFC6130] MUST ignore such 2-Hop Tuples.

For [RFC7181], given that this protocol uses 2-hop information for MPR Set and Routing Set calculation, but not includes that information in control traffic, this means that an implementation must be (i) behaving as if a 2-Hop Tuple only exists if N2\_lost=false, and (ii) as if a change of N2\_lost (from false to true, or true to false) corresponds to a 2-Hop Tuple appearing or being removed. Specifically, this means behaving as if all of the following changes were to be made to [RFC7181]:

- o In Section 17.6 of [RFC7181], point 1, replace the final two bullet points with:

- \* A 2-Hop Tuple with N2\_out\_metric != UNKNOWN\_METRIC and N2\_lost = false is added or removed, OR;
  - \* A 2-Hop Tuple with N2\_out\_metric != UNKNOWN\_METRIC has N2\_lost changed, OR;
  - \* The N2\_out\_metric of any 2-Hop Tuple with N2\_lost = false changes, and either the flooding MPR selection process uses metric values (see Section 18.4) or the change is to or from UNKNOWN\_METRIC.
- o In Section 17.6 of [RFC7181], point 3, replace the final two bullet points with:
    - \* A 2-Hop Tuple with N2\_in\_metric != UNKNOWN\_METRIC and N2\_lost = false is added or removed, OR;
    - \* A 2-Hop Tuple with N2\_in\_metric != UNKNOWN\_METRIC has N2\_lost changed, OR;
    - \* The N2\_in\_metric of any 2-Hop Tuple with N2\_lost = false changes.
  - o In Section 17.7 of [RFC7181], in the fifth bullet point, add "and N2\_lost = false" after "N2\_out\_metric != UNKNOWN\_METRIC".
  - o In Section 18.4 of [RFC7181], in the third bullet point, add ", N2\_lost = false" after "N2\_out\_metric != UNKNOWN\_METRIC".
  - o In Section 18.5 of [RFC7181], in the third bullet point, add ", N2\_lost = false" after "N2\_in\_metric != UNKNOWN\_METRIC".
  - o In Section 19.1 of [RFC7181], in the final main bullet point (marked as "(OPTIONAL)"), add "and N2\_lost = false" after "N2\_out\_metric != UNKNOWN\_METRIC".
  - o In Appendix C.7 of [RFC7181], in point 1, add "and N2\_lost = false" after "N2\_out\_metric != UNKNOWN\_METRIC".

## 6. MIB Considerations

This update to [RFC6130] does not change the definition of a symmetric 2-hop neighbor. It adds new information and states for each symmetric 2-hop neighbor, recorded in the Neighbor Information Base of a router and to be reflected in the appropriate tables of the corresponding NHDP-MIB module [RFC6779].

### 6.1. Updates to the State Group

This update introduces, to the state of each 2-Hop Tuple, the boolean flag `N2_lost`. In order to reflect this, the updates in this section are to be made to the State Group (`nhdpStateObjGrp`) of the NHDP-MIB module [RFC6779].

The DESCRIPTION of `nhdpIib2HopSetEntry` Object Type is to be updated, so as to read as follows:

```
nhdpIib2HopSetEntry  OBJECT-TYPE
    SYNTAX             Nhdpiib2HopSetEntry
    MAX-ACCESS          not-accessible
    STATUS              current
    DESCRIPTION
```

```
    "nhdpIib2HopSetTable consists of 2-Hop Tuples, each
     representing a single network address of a symmetric
     2-hop neighbor and a single MANET interface of a
     symmetric 1-hop neighbor.
```

```
    (N2_neighbor_iface_addr_list,
     N2_2hop_addr, N2_lost, N2_time).
```

```
    The entries include the 2-hop neighbor addresses, which
    act as the table index, and associated symmetric 1-hop
    neighbor address set, designated through nhdpDiscIfIndex,
    an expiration time, and a flag indicating if the 1-hop
    neighbor, through which this 2-hop neighbor is reachable,
    is considered lost due to link quality, or not.
```

```
    The nhdpIfIndex in the INDEX is the interface index of
    the local interface through which these 2-hop addresses
    are accessible. The nhdpDiscIfIndex in the INDEX
    represents the 1-hop neighbor interface through which
    these 2-hop neighbor addresses are reachable."
```

#### REFERENCE

```
    "RFC 6130 - Mobile Ad Hoc Network (MANET) Neighborhood
     Discovery Protocol (NHDP), Clausen, T., Dearlove,
     C., and J. Dean, April 2011"
```

```
INDEX { nhdpIfIndex,
        nhdpDiscIfIndex,
        nhdpIib2HopSetIpAddressType,
        nhdpIib2HopSetIpAddress
      }
 ::= { nhdpIib2HopSetTable 1 }
```

The SEQUENCE of `Nhdpiib2HopSetEntry` is to be updated, so as to read as follows:

```

NhdpIib2HopSetEntry ::=
  SEQUENCE {
    nhdpIib2HopSetIpAddressType
      InetAddressType,
    nhdpIib2HopSetIpAddress
      InetAddress,
    nhdpIib2HopSetIpAddrPrefixLen
      InetAddressPrefixLength,
    nhdpIib2HopSet1HopIfIndex
      NeighborIfIndex,
    nhdpIib2HopSetN2Time
      TimeStamp,
    nhdpIib2HopSetN2Lost
      TruthValue
  }

```

The nhdpIib2HopSetN2Lost OBJECT-TYPE is to be defined as follows:

```

nhdpIib2HopSetN2Lost OBJECT-TYPE
  SYNTAX      TruthValue
  MAX-ACCESS  read-only
  STATUS      current
  DESCRIPTION
    "nhdpIib2HopSetN2Lost corresponds to N2_lost of NHDP and
     is a boolean flag, describing if for a 2-Hop Tuple, the
     corresponding Link Tuple currently is considered lost
     due to link quality."
  REFERENCE
    "draft-dearlove-manet-nhdp-optimization-01"
  ::= { nhdpIib2HopSetEntry 5}

```

## 6.2. Updates to the Notification Group

This update introduces an additional state for each 2-Hop Tuple. Whereas [RFC6130] has two states for 2-Hop Tuples, 'up' (a 2-Hop Tuple exists) and 'down' (a 2-Hop Tuple expires), this update introduces a third state for a 2-Hop Tuple: it exists, but (due to the link quality of the link to the corresponding 1-Hop neighbor) is not currently considered.

To reflect this, the SYNTAX and DESCRIPTION of nhdp2HopNbrState OBJECT-TYPE are to be updated, so as to read as follows:

```
nhdp2HopNbrState OBJECT-TYPE
    SYNTAX  INTEGER {
                down(0),
                up(1),
                notconsidered(2)
            }
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "NHDP 2-hop neighbor states. In NHDP, it is not necessary
        to remove Protocol Tuples from Protocol Sets at the
        exact time indicated, only to behave as if the Protocol
        Tuples were removed at that time. This case is indicated
        here as 'down(0)'; otherwise, it is either 'up(1)', if
        N2_lost for the 2-Hop Tuple is equal to false, or
        'notconsidered(2)' otherwise."
    ::= { nhdpNotificationsStates 2 }
```

## 7. IANA Considerations

This document has no actions for IANA.

[This section may be removed by the RFC Editor.]

## 8. Security Considerations

The update to [RFC6130] enables the retention and reuse of some information collected by that protocol, for only the duration that it could have been used in any case. As such, this protocol introduces no new security considerations to an implementation of [RFC6130] or of any other protocol that uses it, such as [RFC7181].

## 9. Acknowledgments

The authors would like to thank Liz Cullen (BAE Systems) for first illustrating the issue addressed in this specification.

## 10. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC6130] Clausen, T., Dean, J., and C. Dearlove, "Mobile Ad Hoc Network (MANET) Neighborhood Discovery Protocol (NHDP)",

RFC 6130, April 2011.

[RFC6779] Herberg, U., Cole, R., and I. Chakeres, "Definition of Managed Objects for the Neighborhood Discovery Protocol", RFC 6779, October 2012.

[RFC7181] Clausen, T., Dearlove, C., Jacquet, P., and U. Herberg, "The Optimized Link State Routing Protocol version 2", RFC 7181, April 2014.

#### Authors' Addresses

Christopher Dearlove  
BAE Systems Advanced Technology Centre  
West Hanningfield Road  
Great Baddow, Chelmsford  
United Kingdom

Phone: +44 1245 242194  
Email: [chris.dearlove@baesystems.com](mailto:chris.dearlove@baesystems.com)  
URI: <http://www.baesystems.com/>

Thomas Heide Clausen  
LIX, Ecole Polytechnique

Phone: +33 6 6058 9349  
Email: [T.Clausen@computer.org](mailto:T.Clausen@computer.org)  
URI: <http://www.ThomasClausen.org/>





Network Working Group  
Internet-Draft  
Intended status: Experimental  
Expires: August 29, 2015

Y. Yi  
S. Lee  
University of California, Los Angeles  
W. Su  
The Boeing Company  
M. Gerla  
A. Colin de Verdiere  
University of California, Los Angeles  
February 25, 2015

On-Demand Multicast Routing Protocol (ODMRP) for Ad Hoc Networks  
draft-gerla-manet-odmrp-05

Abstract

The On-Demand Multicast Routing Protocol (ODMRP) is a multicast routing protocol designed for ad hoc networks with mobile hosts. ODMRP is a mesh-based, rather than a conventional tree-based, multicast scheme and uses a forwarding group concept (only a subset of nodes forwards the multicast packets via scoped flooding). It applies on-demand procedures to dynamically build routes and maintain multicast group membership, without relying on pre-existing unicast routing protocols. ODMRP is well suited for ad hoc wireless networks with mobile hosts where bandwidth is limited, topology changes frequently and rapidly, and power is constrained.

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on August 29, 2015.

Copyright Notice

Copyright (c) 2015 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

1. Introduction . . . . .	4
1.1. Motivation and Experiments . . . . .	4
2. Terminology and Notation . . . . .	5
2.1. Notation . . . . .	5
2.2. Terminology . . . . .	5
3. Applicability Statement . . . . .	7
4. Protocol Overview and Functioning . . . . .	7
4.1. Routers and Interfaces . . . . .	8
4.2. Information Base Overview . . . . .	8
4.3. Signaling Overview . . . . .	9
4.4. Overview . . . . .	9
5. Parameters and Constants . . . . .	11
5.1. Router Parameters . . . . .	11
5.2. Interface Parameters . . . . .	11
6. Sequence Numbers . . . . .	12
7. Packets and Messages . . . . .	12
7.1. Join Query Format . . . . .	12
7.2. Join Reply Format . . . . .	13
8. RFC5444 Encoding . . . . .	13
8.1. Join Query Encoding . . . . .	14
8.2. Join Reply Encoding . . . . .	14
9. Information Bases . . . . .	15
9.1. Local Interface Set . . . . .	15
9.2. Neighbor Interface Set . . . . .	15
9.3. Multicast Routing Set . . . . .	16
9.4. Forwarding Table . . . . .	16
9.5. Pending Acknowledgements . . . . .	17
9.6. Pre-acknowledgements . . . . .	18
9.7. Blacklist . . . . .	18
9.8. Sent JQ set . . . . .	19
10. Protocol Details . . . . .	19
10.1. Join Query . . . . .	20

10.1.1. Invalid Join Queries . . . . .	20
10.1.2. Join Query Generation . . . . .	20
10.1.3. Join Query Processing . . . . .	21
10.1.4. Join Query Forwarding . . . . .	22
10.2. Join Reply . . . . .	22
10.2.1. Invalid Join Replies . . . . .	23
10.2.2. Join Reply Generation . . . . .	23
10.2.3. Join Reply Processing . . . . .	23
10.2.4. Join Reply Forwarding . . . . .	25
10.2.5. Join Reply Transmission . . . . .	26
10.3. Forwarding Group Maintenance . . . . .	27
10.4. Message Transmission . . . . .	27
11. Unidirectional Links Handling . . . . .	27
12. SMF considerations . . . . .	29
13. IGMP and MLD considerations . . . . .	29
14. Multicast Packet Forwarding . . . . .	29
15. Security Considerations . . . . .	30
15.1. Confidentiality . . . . .	30
15.2. Integrity . . . . .	30
15.3. Channel Overload . . . . .	31
16. IANA Considerations . . . . .	31
16.1. Join Query Registries . . . . .	31
16.2. Join Reply Registries . . . . .	32
17. Acknowledgements . . . . .	33
18. References . . . . .	33
18.1. Normative References . . . . .	33
18.2. Informative References . . . . .	34
Appendix A. Illustrations . . . . .	35
A.1. Join Query Message . . . . .	35
A.2. Join Reply Message . . . . .	36
Authors' Addresses . . . . .	37

## 1. Introduction

This document describes the On-Demand Multicast Routing Protocol (ODMRP) [ODMRP-Journal]. ODMRP applies "on-demand" routing techniques to avoid channel overhead and improve scalability. It uses the concept of "forwarding group" [FGMP], a set of nodes responsible for forwarding multicast data, to build a forwarding mesh for each multicast group. By maintaining and using a mesh instead of a tree, the drawbacks of multicast trees in mobile wireless networks (e.g., intermittent connectivity, traffic concentration, frequent tree reconfiguration, non-shortest path in a shared tree, etc.) are avoided. A soft-state approach is taken to maintain multicast group members, meaning that no explicit control message is required to leave the group. ODMRP does not rely on any unicast routing protocol: in particular, it can operate in conjunction with both reactive and proactive unicast routing protocols.

### 1.1. Motivation and Experiments

The main rationale for ODMRP is its potential to reduce control and traffic overhead in certain MANET deployments, typically where multicast traffic is relatively sparse. While this protocol has been extensively studied in simulations, it does not yet benefit from sufficient operational experience in order to be considered for Standards Track. In addition to general operational experience such as interoperability testing, this specification is intended to collect data on the following points:

- o As a multicast routing protocol for MANET, ODMRP can be compared with [RFC6621], but can also be used in conjunction, taking advantage of its Duplicate Packet Detection and optimized flooding mechanisms. The rationale behind ODMRP is that, with sparser traffic, and in particular less sources, ODMRP should reduce the control overhead and number of data packets transmitted by making use of Forwarding Groups. This hypothesis should be validated, and experiments and operational deployments demonstrating the scenarios in which ODMRP performs better, or worse, than [RFC6621] should be performed.
- o The potential scope of deployment of ODMRP should be assessed, particularly in comparison to other MANET protocols.
- o Default values and guidelines for the parameters described in Section 5 should be provided, based on operational experience gathered from implementing and deploying this specification.
- o The feasibility of implementing ODMRP in common MANET situations should be examined. In particular, it should be determined if a

linux user space implementation is possible.

## 2. Terminology and Notation

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].

This document also makes use of the terminology defined in [RFC5444]. Additionally, it uses the notation defined in Section 2.1, and the terminology defined in Section 2.2.

### 2.1. Notation

ODMRP Routers generate and process messages, each of which has a number of distinct fields. For describing the protocol operations, specifically the generation and processing of such messages, the following notation is employed:

MsgType.field

where:

MsgType - is the type of message (e.g., JQ or JR);

field - is the field in the message (e.g., SourceAddress).

Furthermore, the following notational conventions are used:

a := b an assignment operator, whereby the left side (a) is assigned the value of the right side (b)

c = d a comparison operator, returning TRUE if and only if the value of the left side (c) is equal to the value of the right side (d)

[x] a list containing x as its only element

The different messages, their fields and their meaning are described in Section 7.

### 2.2. Terminology

ODMRP Router - A router that implements this protocol. An ODMRP Router is equipped with at least one, and possibly more, ODMRP Interfaces.

ODMRP Interface - An ODMRP Router's attachment to a communication medium, over which it receives and generates control messages, according to this specification. An ODMRP Interface is assigned one or more addresses.

Neighbor (ODMRP) Router - An ODMRP Router A is a neighbor of another ODMRP Router B if B can receive control messages from A according to this specification. This relationship is not necessarily symmetrical.

Neighbor (ODMRP) Interface - An interface X of an ODMRP Router A is a neighbor relative to interface Y of ODMRP Router B if B can receive control messages sent by A over the link X - Y. The link, and this relationship, are not necessarily symmetrical.

Multicast session - The entity defined by a (multicast group, source) pair, representing the group to which a source sends multicast packets.

Forwarding group - A group of ODMRP Routers participating in multicast packet forwarding for a given Multicast Session. In particular, the Forwarding Group is constituted of the Multicast Source, the Multicast Receivers and the Intermediate Routers.

Multicast Receiver - An ODMRP Router is a Multicast Receiver, relative to a given Multicast Session, if it subscribes to the Multicast Session in order to receive data packets sent by the Multicast Source.

Intermediate Router - An Intermediate Router is an ODMRP Router that is a member of a Forwarding Group without being a Multicast Receiver. In other words, it joined the Forwarding Group to transmit control and data traffic between the Multicast Source and the Multicast Receivers.

Join Query - The control message sent by Multicast Sources to establish and update group memberships and routes.

Join Reply - The control message sent by Multicast Receivers and forwarded by Intermediate Routers to build the Forwarding Group according to group membership information.

Upstream - An ODMRP Router (A) is said to be "upstream" compared to another ODMRP Router (B), relatively to a given Multicast Session, if A is on the path, which is discovered by a Join Query-Join Reply exchange and used by data packets, between B and the Multicast Source. In other words, any data packet sent within the Multicast Session has to transit through A before reaching B.

Downstream - An ODMRP Router (A) is said to be "downstream" compared to another ODMRP Router (B), relatively to a given Multicast Session, if B is on the path which is discovered by a Join Query-Join Reply exchange and used by data packets, between A and the Multicast Source. In other words, A is "downstream" from B if B is "upstream" from A.

### 3. Applicability Statement

This protocol is a multicast routing protocol, intended for use in Mobile Ad Hoc Networks (MANETs). MANETs generally have constrained resources (processing power, battery, etc.) and very dynamic topologies. With ODMRP, routing state is installed and maintained in an on-demand fashion, which avoids the issue of frequent tree reconfiguration seen with more classic multicast routing protocols.

ODMRP does not rely on the use of any unicast routing protocol, whether reactive or proactive, but MAY be used coinjointly with such protocols, such as [RFC7181] or [RFC3561]. Additionally, ODMRP can run in conjunction with [RFC6621], and take advantage of any optimized flooding mechanism used in the network, such as those offered by SMF, to disseminate Join Query messages as described in Section 12.

### 4. Protocol Overview and Functioning

The objective of this protocol is to allow each ODMRP Router to:

- o Build a Forwarding Group only when it has data traffic to send to a Multicast group.
- o Maintain the Forwarding Group for as long as necessary, until there is no more data to be sent to the Multicast group.
- o Join any Forwarding Group, in order to receive multicast data packets from the corresponding multicast source. The decision to join a given Forwarding Group is triggered by Multicast membership information relative to the corresponding Multicast session. Such information can be received from other protocols, such as IGMP

[RFC3376] and MLD [RFC3810].

#### 4.1. Routers and Interfaces

Each ODMRP Router MUST be provisioned with at least one ODMRP Interface, and keep a list of all these interfaces, as described in Section 9. The management of these interfaces (addition, deletion, re-addressing of any interface) is out of scope for this document.

#### 4.2. Information Base Overview

Protocol state is recorded in eight distinct information sets: the Local Interface Set, the Neighbor Interface Set, the Multicast Routing Set, the Forwarding Table, the Pending Acknowledgement Set, the Pre-acknowledgement Set, the Blacklist and the Sent JQ Set. With the exception of the Local Interface Set, all these information sets are both used and updated by this protocol.

The Local Interface Set records a list matching each ODMRP interface of this router to the addresses in use for this interface. This set is used, but not updated by this protocol.

The Neighbor Interface Set records all the known addresses of neighbor ODMRP interfaces, by way of recording data from received JQ messages. This set can also be updated by other protocols with knowledge of neighbor interfaces, such as [RFC6130].

The Multicast Routing Set contains tuples, each representing the address of a multicast group, the address of a source sending data to this multicast group, and the next hop towards the multicast source.

The Forwarding Table contains tuples, each representing a given Multicast session for which the ODMRP Router forwards packets.

The Pending Acknowledgement Set contains tuples, each corresponding to a Join Reply message which has been sent by this Router and is waiting for an acknowledgement from a chosen upstream Router.

The Pre-acknowledgement Set contains tuples, representing overheard Join Reply messages, that are not destined to this Router but may pre-acknowledge a future Join Reply from this Router.

The Blacklist contains tuples, corresponding to neighbor ODMRP Routers, with which connectivity has been detected to be unidirectional.

The Sent JQ Set matches interfaces of this Router with the address carried by the last JQ message to have transited through that



interface.

#### 4.3. Signaling Overview

This protocol generates and processes the following routing messages:

Join Query - Generated by an ODMRP Router while it has data packets to send to a multicast group, and flooded periodically to maintain the Forwarding Group necessary to deliver these data packets. A Join Query message hence advertises a Multicast Session, and contains:

- \* The multicast group address
- \* The source address
- \* A sequence number
- \* The last address used by this interface to send a JQ message

Join Reply - Generated by an ODMRP Router belonging to a multicast session, in reply to a Join Query message advertising this multicast session (corresponding Join Query), then forwarded by ODMRP Routers belonging to the same multicast session along the reverse path to the multicast source. A Join Reply message contains:

- \* The multicast group address and source address, identifying the multicast session
- \* The sequence number carried by the corresponding Join Query
- \* The address of the next hop on the path towards the multicast session source

#### 4.4. Overview

The objectives of this protocol are achieved, for each ODMRP Router, by the following operations:

- o When having data to send to a multicast group, for which no Forwarding Group is already established, an ODMRP Router generates a Join Query and transmits it over all of its ODMRP Interfaces. It then periodically repeat this process, until it has no more data to send to the multicast group.
- o Upon receiving a Join Query, an ODMRP Router installs or refreshes a tuple in the Multicast Routing Set indicating the reverse path

towards the source of the Join Query, then considers it for forwarding, according to the forwarding mechanism specified for the network.

- o If this Router belongs to, i.e., has attached hosts which have subscribed to, the multicast session that the Join Query advertises, it originates a Join Reply and transmits it over all of its ODMRP Interfaces.
- o Upon receiving a Join Reply, an ODMRP Router inspects the next hop address carried by this packet:
  - \* If it corresponds to the address of an interface of this Router, and if this Router has a tuple in its Multicast Routing Set, corresponding to the advertised source, the ODMRP Router belongs to the Forwarding Group for the Multicast Session. Consequently, it installs or refreshes the corresponding entry in its Forwarding Table. It then considers the Join Reply for forwarding, according to the forwarding mechanism specified for the network.
  - \* Otherwise, it verifies if any Pending Acknowledgement tuple corresponds to this Join Reply and marks each such tuple as acknowledged. It silently discards the Join Reply.
- o After sending a Join Reply, addressed to an upstream router A, an ODMRP Router looks in its Pre-acknowledgement Set for a corresponding Overheard tuple.
  - \* If such a tuple exists, the Overheard tuple is discarded and no further action is taken.
  - \* Otherwise, i.e., if the Pre-acknowledgement Set does not contain any corresponding Overheard tuple, it creates a Pending Acknowledgement tuple in the Pending Acknowledgement Set. If this tuple expires without being acknowledged, the link with router A is considered unidirectional: it is blacklisted, and the current router MAY try other means of joining the Forwarding Group.
- o While it has data to send to the multicast group, an ODMRP Router periodically originates a Join Query and transmits it to all of its neighbors, in order to maintain the Forwarding Group.

## 5. Parameters and Constants

This specification uses both Router parameters, described in Section 5.1, and per-interface parameters, described in Section 5.2.

### 5.1. Router Parameters

This specification uses the following Router parameters:

ROUTE\_REFRESH\_INTERVAL - is the interval between two periodic Join Queries sent by a Multicast Source

FG\_TIMEOUT - is the minimum time a Forwarding Tuple SHOULD be kept in the Forwarding Table after it was last refreshed

### 5.2. Interface Parameters

This specification uses the following interface parameters:

ROUTE\_TIMEOUT - is the minimum time a Routing Tuple SHOULD be kept in the Routing Set after it was last refreshed

JR\_RETRIES - is the number of times an ODMRP Router SHOULD attempt to retransmit a given Join Reply before declaring the link with the upstream neighbor interface unidirectional

ACK\_TIMEOUT - is the time after which a Pending Tuple expires and MUST be considered invalid, as well as trigger the appropriate action according to Section 11

PRE\_ACK\_TIMEOUT - is the time after which an Overheard Tuple expires and MUST be considered invalid

LOCAL\_ADDRESS\_TIMEOUT - is the time after which a sent JQ tuple expires and MUST be considered invalid. This parameter SHOULD be less than the time an interface address is expected to be in use for the corresponding communication medium

NEIGHBOR\_ADDRESS\_TIMEOUT - is the time after which an address tuple of a neighbor interface tuple expires and MUST be considered invalid. This parameter SHOULD be less than the time an interface address is expected to be in use for the corresponding communication medium

## 6. Sequence Numbers

Each ODMRP Router maintains a single sequence number, which must be included in each Join Query message it generates. Each ODMRP Router MUST make sure that no two Join Query messages are generated with the same sequence number, and MUST generate sequence numbers such that these are monotonically increasing. This sequence number is used as freshness information for when comparing routes to the ODMRP Router having generated the message.

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 the protocol, the following MUST be observed. The term `MAX_SEQ_NUM` designates in the following the largest possible value for a sequence number. The sequence number `S1` is said to be "greater than" (denoted '>') the sequence number `S2` if:

$$S2 < S1 \text{ AND } S1 - S2 \leq \text{MAX\_SEQ\_NUM}/2 \text{ OR}$$
$$S1 < S2 \text{ AND } S2 - S1 > \text{MAX\_SEQ\_NUM}/2$$

## 7. Packets and Messages

This section describes the protocol messages generated and processed by ODMRP, according to the notations defined in Section 2. The objective of this section is to specify the content and meaning of each message. The specifics of the encoding of these messages, including the exact type and length of each field, in accordance with [RFC5444], are described in Section 8.

### 7.1. Join Query Format

A Join Query (JQ) message has the following fields:

`JQ.AddressLength` encodes the length of the addresses carried by this message as follows:

$$\text{JQ.AddressLength} := \text{the length of an address in octets} - 1$$

`JQ.MulticastGroupAddress` encodes the address of the Multicast Group, to which this Join Query is addressed

JQ.SourceAddress encodes an address of the source of this Join Query

JQ.SequenceNumber encodes the sequence number (see Section 6) of the ODMRP Router, generating the Join Query message

JQ.LastAddress encodes the address set as the source address of the last IP datagram sent through the same interface and containing a JQ message, or of the IP datagram carrying this JQ message if no such datagram is known

## 7.2. Join Reply Format

A Join Reply (JR) message has the following fields:

JR.AddressLength encodes the length of the addresses carried by this message as follows:

JR.AddressLength := the length of an address in octets - 1

JR.MulticastGroupAddress encodes the address of the Multicast Group, to which this Join Reply is addressed

JR.AckRequired is a boolean flag. When set ('1'), it specifies that the recipient of the Join Reply MUST acknowledge its reception by a sending Join Reply message. If cleared ('0'), the recipient of this message MAY suppress its Join Reply transmission, according to Section 10

JR.SourceAddress encodes the address of the Source of the Multicast Session

JR.SequenceNumber encodes the sequence number (see Section 6) of the corresponding Join Query message

JR.NextHopAddress encodes the the address of the next hop on the path towards the source of the multicast session

## 8. RFC5444 Encoding

This section describes the encoding of ODMRP messages using [RFC5444].

### 8.1. Join Query Encoding

This protocol defines the Join Query message type. Hence, according to [RFC5444], all Join Query messages are generated, processed and transmitted following this specification. Table 1 shows the mapping between the Join Query elements described in Section 7.1 and their encoding. All elements described in Table 1 MUST be included in every Join Query message, with the exception of the JQ.LastAddress element. JQ.LastAddress MAY be omitted in a given JQ message if it corresponds to the source address of the IP datagram containing this message.

JQ Element	RFC5444 Element
JQ.AddressLength	<msg-addr-length>
JQ.SourceAddress	<msg-orig-addr>
JQ.MulticastGroupAddress	Address in address block + TLV
JQ.SequenceNumber	<msg-seq-num>
JQ.LastAddress	Address in address block + TLV

Table 1: Join Query Message Elements

### 8.2. Join Reply Encoding

This protocol defines the Join Reply message type. Hence, according to [RFC5444], all Join Reply messages are generated, processed and transmitted following this specification. Table 2 shows the mapping between the Join Reply elements described in Section 7.2 and their encoding. With the exception of the ACKREQUIRED TLV, all elements described in Table 2 MUST be included in every Join Reply message.

JR Element	RFC5444 Element
JR.AddressLength	<msg-addr-length>
JR.SourceAddress	<msg-orig-addr>
JR.MulticastGroupAddress	Address in address block + TLV
JR.SequenceNumber	<msg-seq-num>
JR.NextHopAddress	Address in address block + TLV
JR.AckRequired	ACKREQUIRED TLV

Table 2: Join Reply Message Elements

## 9. Information Bases

Each router maintains an Information Base, containing a Local Interface Set, a Neighbor Interface Set, a Multicast Routing Set, a Forwarding Table, a Pending Acknowledgement Set, a Pre-Acknowledgement Set, a Blacklist and a Sent JQ Set, as described in the following sections. These information sets are given so as to facilitate description of message generation, forwarding and processing rules. In particular, an implementation may choose any representation or structure for when maintaining this information.

### 9.1. Local Interface Set

The Local Interface Set records a list of all the interfaces of the ODMRP Router, which participate in the operations of this protocol; that is, over which ODMRP control messages are exchanged, according to this specification. Each tuple of the Interface Set, or Interface Tuple, is as follows:

(I\_interface, I\_interface\_address\_list)

Where:

I\_interface - The local ODMRP Interface

I\_interface\_address\_list - The list of addresses used by the local interface in the operations of this protocol.

### 9.2. Neighbor Interface Set

The Neighbor Interface Set records the known addresses of Neighbor ODMRP Interfaces. It is used by this protocol, and can be updated by this protocol or by any other suitable protocol in operation that provides the necessary information, such as NHDP [RFC6130]. Each neighbor interface tuple is as follows:

(N\_interface\_address\_list)

Where:

N\_interface\_address\_list - Is an unordered list of at least one address tuple (i\_addr, i\_addr\_exp\_time), where:

i\_addr - is a known address of the Neighbor Interface, i.e., an address that was set as the sender of an IP datagram sent through this interface

i\_addr\_exp\_time - is the time at which the address tuple MUST be considered expired and thus MUST NOT be taken in considerations for the operations of this protocol

A neighbor interface tuple that contains no valid (i.e., non-expired) address tuple MUST be considered expired and MUST NOT be taken in considerations for the operations of this protocol

### 9.3. Multicast Routing Set

The Multicast Routing Set contains Routing Tuples, indicating the path towards Multicast Sources, and containing the following fields:

(R\_source, R\_next\_hop, R\_local\_interface,  
R\_seq\_num, R\_exp\_time)

Where:

R\_source - is the address of the Multicast Source.

R\_next\_hop - is an address of the next hop along the path to the Multicast Source, i.e., an address of one of the interfaces of the neighbor ODMRP Router, from which the last valid Join Query message from this source was received, as recorded by the packet containing this Join Query.

R\_local\_interface - is the local interface, through which the next hop can be reached.

R\_seq\_num - corresponds to the JQ.SequenceNumber of the last valid Join Query originated by the Multicast Source and received by this ODMRP Router.

R\_exp\_time - is the time at which the tuple MUST be considered expired and thus MUST NOT be taken into consideration by the operations of this protocol.

### 9.4. Forwarding Table

The Forwarding Table contains Forwarding Tuples, representing Multicast Sessions for which the ODMRP Router forwards messages, i.e., the ODMRP Router is part of these Multicast Sessions' Forwarding Groups. These tuples are as follows:

(F\_multicast\_group, F\_multicast\_source,  
F\_seq\_num, F\_exp\_time)

Where:



F\_multicast\_group - is the address of the Multicast Group of the Multicast Session, for which the ODMRP Router forwards messages.

F\_source - is the address of the Multicast Source of the Multicast Session, for which the ODMRP Router forwards messages.

F\_seq\_num - is the sequence number, corresponding to the last Join Query sent by the multicast source for the multicast session.

F\_exp\_time - is the time at which the tuple MUST be considered expired and thus MUST NOT be taken into consideration by the operations of this protocol.

#### 9.5. Pending Acknowledgements

The Pending Acknowledgements Set contains Pending Acknowledgement tuples, representing Join Reply messages that are waiting to be acknowledged by the selected upstream Forwarding Group member. These tuples are as follows:

(P\_multicast\_group, P\_multicast\_source, P\_seq\_num,  
P\_local\_interface, P\_next\_hop, P\_nth\_time, P\_exp\_time)

Where:

P\_multicast\_group - is the JR.MulticastGroupAddress carried in the Join Reply awaiting acknowledgement (henceforth corresponding Join Reply).

P\_multicast\_source - is the JR.SourceAddress field carried in the corresponding Join Reply.

P\_seq\_num - is the JR.SequenceNumber field of the corresponding Join Reply.

P\_next\_hop - is the JR.NextHopAddress field of the corresponding Join Reply.

P\_local\_interface - is the local interface, through which the Join Reply was sent.

P\_nth\_time - corresponds to the number of times this Join Reply has been previously sent without being acknowledged.

P\_exp\_time - is the time at which this tuple MUST be considered expired.

P\_acknowledged - is a boolean indicating whether the corresponding Join Reply has been acknowledged.

#### 9.6. Pre-acknowledgements

The Pre-acknowledgements Set contains Overheard Tuples, corresponding to Join Reply messages, which have been sent by neighbors of this ODMRP Router but do not contain an address of this Router and do not acknowledge any tuple in the Pending Acknowledgement Set. The Overheard Tuples are as follows:

(O\_multicast\_group, O\_multicast\_source, O\_seq\_num,  
O\_originator, O\_exp\_time)

Where:

O\_multicast\_group - is the JR.MulticastGroupAddress carried in the overheard Join Reply.

O\_multicast\_source - is the JR.SourceAddress field carried in the corresponding Join Reply.

O\_seq\_num - is the JR.SequenceNumber field of the corresponding Join Reply.

O\_originator - is the address of the ODMRP Router's interface which has sent the Join Reply.

O\_exp\_time - is the time at which this tuple expires MUST be considered invalid.

#### 9.7. Blacklist

The Blacklist contains Blacklisted Tuples, corresponding to neighbor ODMRP Router interfaces, with which connectivity has been detected to be unidirectional, e.g., which have not acknowledged Join Replies from this Router, as specified in Section 10. In other words, a Blacklisted Tuple corresponds to a link between one local interface and one neighbor interface which has been detected to be unidirectional or broken. The Blacklist Tuples are as follows:

(B\_neighbor\_interface, B\_local\_interface, B\_exp\_time)

Where:

B\_neighbor\_interface\_address\_list - is a list of addresses belonging to the blacklisted interface.

B\_local\_interface - is the interface of this ODMRP router over which packets from the blacklisted interface were received.

B\_exp\_time - is the time at which this tuple expires and MUST be considered invalid.

#### 9.8. Sent JQ set

The Sent JQ Set contains tuples matching transmitted (generated or relayed) Join Queries with interfaces addresses. Each of its tuples contains the source address, multicast group address and sequence number uniquely identifying a JQ message, as well as an interface and the address of that interface that was advertised when transmitting the packet containing the Join Query. More precisely, given a transmitted Join Query and an interface over which it was transmitted, a tuple of this set, or Sent JQ Tuple, is as follows:

Where:

S\_interface - is the local interface, through which the JQ message was sent

S\_interface\_address - is the address of the ODMRP interface that was set as the packet source

S\_exp\_time - is the time at which this tuple expires and MUST be considered invalid.

#### 10. Protocol Details

This protocol generates and processes Join Query and Join Reply messages, according to the operations described in the following sections. This section uses the additional notation and variables:

previous-hop-address - refers to the address of the neighbor ODMRP interface recorded by the source address field of the IP datagram carrying the message currently being processed (Join Query or Join Reply)

this Router - refers to the ODMRP Router generating, processing or forwarding the message (Join Query or Join Reply)

Receiving Interface (receiving-interface) - refers to the local ODMRP Interface, over which the message currently being processed was received

#### 10.1. Join Query

A Join Query is generated by an ODMRP Router, which has data to send to a multicast group, for which no multicast session has been initialized. Join Queries are then periodically originated by the ODMRP Router while it has data to send to the multicast group.

##### 10.1.1. Invalid Join Queries

A Join Query, received by an ODMRP Router, is invalid and MUST be discarded without processing (and in particular, MUST NOT be considered for forwarding) if any of these conditions applies:

- o The address length carried by the Join Query (see Section 7) differs from the length of the addresses of this Router
- o The Multicast Routing Set of this Router contains a Multicast Routing tuple, for which:
  - \* R\_multicast\_source = JQ.SourceAddress, and
  - \* R\_seqnum > JQ.SequenceNumber or R\_seqnum = JQ.SequenceNumber
- o JQ.SourceAddress is an address of an interface of this Router
- o The Blacklist contains a Blacklisted Tuple, for which
  - \* previous-hop-address is contained in B\_neighbor\_interface\_address\_list
  - \* B\_local\_interface = receiving-interface

##### 10.1.2. Join Query Generation

A Join Query is generated according to Section 7 with the following content:

- o JQ.AddressLength set to the length of the addresses of this Router minus 1, as specified in Section 7

- o JQ.MulticastGroupAddress set to the address of the multicast group, to which this Router is sending data
- o JQ.SourceAddress set to an address of this ODMRP Router
- o JQ.SequenceNumber set to the current sequence number of this Router, as specified in Section 6

#### 10.1.3. Join Query Processing

Upon receiving a valid Join Query message, an ODMRP Router proceeds as follows:

1. Find the neighbor interface tuple such as  
N\_interface\_address\_list contains an address tuple with i\_addr = previous-hop-address, and update the address tuple such as  
i\_addr\_exp\_time := current-time + NEIGHBOR\_ADDRESS\_TIMEOUT
2. If no such tuple exists, create one with:
  - \* N\_interface\_address\_list := [(previous-hop-address, current-time + NEIGHBOR\_ADDRESS\_TIMEOUT)]
3. Find the Routing Tuple which satisfies: R\_source = JQ.SourceAddress
4. If no such tuple exists, create a Routing Tuple with the following fields:
  - \* R\_source := JQ.SourceAddress
  - \* R\_next\_hop := previous-hop
  - \* R\_local\_interface := receiving-interface
  - \* R\_seq\_num := JQ.SequenceNumber
  - \* R\_exp\_time := current-time + ROUTE\_TIMEOUTand insert this tuple in the Routing Set
5. Else, i.e., if such a tuple exist, update it as follows:
  - \* R\_next\_hop := previous-hop
  - \* R\_seq\_num := JQ.SequenceNumber

- \* R\_local\_interface := receiving-interface
  - \* R\_exp\_time := current-time + ROUTE\_TIMEOUT
6. Consider the Join Query for forwarding, according to Section 10.1.4
  7. If this Router is a member of the Multicast Group, addressed by JQ.MulticastGroupAddress, create a new Join Reply according to Section 10.2 and transmit it to all of this Router's neighbors

#### 10.1.4. Join Query Forwarding

This section defines the following additional variables:

this-interface - is the ODMRP interface being considered

packet-source-address - is the source address of the outbound IP datagram carrying the JQ message being transmitted

For each ODMRP interface over which a JQ message is to be transmitted, a router MUST proceed as follows:

- o Find the corresponding sent JQ tuple in the sent JQ set, such as S\_interface = this-interface
- o If no such tuple exists, create one with:
  - \* S\_interface := this-interface
  - \* S\_interface\_address := packet-source-address
  - \* S\_exp\_time := current-time + LOCAL\_ADDRESS\_TIMEOUT
- o Set JQ.LastAddress to S\_interface\_address
- o Update the corresponding sent JQ tuple such as:
  - \* S\_interface\_address = packet-source-address
  - \* S\_exp\_time = current-time + LOCAL\_ADDRESS\_TIMEOUT

#### 10.2. Join Reply

A Join Reply is generated by an ODMRP Router when it receives a Join Query such that at least one host attached to the ODMRP Router is a member of the Multicast Session advertised by the Join Query. This section makes use of the variable "new-jr", which is a boolean flag

set to TRUE if the Join Reply being processed contains more recent data than in the current information base. It has an initial value of FALSE.

#### 10.2.1. Invalid Join Replies

A Join Reply, received by an ODMRP Router, is invalid and MUST be discarded without processing (and in particular, MUST NOT be considered for forwarding) if:

- o The address length carried by the Join Reply (see Section 7) differs from the length of the address of the ODMRP Router
- o There exists a Forwarding Tuple in this Router's Forwarding Group table, such as:
  - \* F\_source = JR.MulticastSourceAddress
  - \* F\_seq\_num > JR.SequenceNumber

#### 10.2.2. Join Reply Generation

An ODMRP Router MUST generate a Join Reply in response to a received Join Query (henceforth "corresponding Join Query"), if at least one host attached to this Router is a member of the Multicast Session, advertised by the Join Query. A Join Reply is generated according to Section 7 with the following content:

- o JR.AddressLength is set to the length of the address of this router minus 1, as specified in Section 7
- o JR.MulticastGroupAddress is set to JQ.MulticastGroupAddress for the corresponding Join Query
- o JR.SourceAddress is set to JQ.SourceAddress for the corresponding Join Query
- o JR.SequenceNumber is set to JQ.SequenceNumber for the corresponding Join Query
- o JR.NextHopAddress is set to the source address of the IP datagram containing the Join Query message

#### 10.2.3. Join Reply Processing

Upon receiving a valid Join Reply, an ODMRP Router proceeds as follows:

1. If JR.NextHopAddress corresponds to an address recorded in the Local Interface Set of this ODMRP Router:
  1. Find the Forwarding Tuple (henceforth Matching Forwarding Tuple) such that:
    - + F\_multicast\_group = JR.MulticastGroupAddress
    - + F\_multicast\_source = JR.MulticastSourceAddress
  2. If no such tuple exists, insert in the Forwarding Table a new Forwarding Tuple such that:
    - + F\_multicast\_group = JR.MulticastGroupAddress
    - + F\_multicast\_source = JR.MulticastSourceAddress
    - + F\_seq\_num = JR.SequenceNumber
    - + F\_exp\_time = current-time + FG\_TIMEOUTAnd set new-jr to TRUE
  3. Otherwise, the variable "new-jr" is set to TRUE if JR.SequenceNumber > F\_seq\_num, and to FALSE otherwise. Then, the pre-existing Matching Forwarding Tuple is updated as follows:
    - + F\_seq\_num := JR.SequenceNumber
    - + F\_exp\_time := current-time + FG\_TIMEOUT
  4. If new-jr = TRUE or if JR.AckRequired is set the Join Reply is considered for forwarding. Otherwise, it is not processed further; in particular, it MUST NOT be considered for forwarding.
2. Otherwise, find the Multicast Routing Tuple in the Routing Set (henceforth "Matching Multicast Routing Tuple"), such as:
  - \* R\_source = JR.SourceAddress
  - \* R\_seq\_num <= JR.SequenceNumberIf previous-hop-address = R\_next\_hop, then:
  3. If the Pending Acknowledgement Set contains a Pending Tuple (henceforth "Matching Pending Tuple") such as:



- + P\_multicast\_group = JR.MulticastAddress
- + P\_multicast\_source = JR.SourceAddress
- + P\_seq\_num = JR.SequenceNumber
- + P\_next\_hop = previous-hop-address

The Matching Pending Tuple MUST be updated as follows:

- + P\_acknowledged = TRUE
- + P\_exp\_time = EXPIRED

The Join Reply is not processed further, and in particular MUST NOT be considered for forwarding

4. Otherwise, if the Pre-Acknowledgement Set does not contain any Overheard Tuple such as:

- + O\_multicast\_group = JR.MulticastGroupAddress
- + O\_multicast\_source = JR.SourceAddress
- + O\_seq\_num = JR.SequenceNumber
- + O\_originator = previous-hop-address

Insert a tuple with these fields, and O\_exp\_time = current-time + PRE\_ACK\_TIMEOUT in the Pre-Acknowledgement Set. The Join Reply is not processed further, and in particular MUST NOT be considered for forwarding

3. Otherwise, the Join Reply is silently discarded without further processing

#### 10.2.4. Join Reply Forwarding

A Join Reply, considered for forwarding, MUST be updated as follows:

- o Find the Matching Routing Tuple, such that:

- \* R\_source = JR.MulticastSourceAddress
- \* R\_seq\_num <= JR.SequenceNumber

- o If no such tuple exists, then the Join Reply is not processed further, and in particular MUST NOT be forwarded
- o Otherwise, set JR.NextHop to R\_next\_hop

The Join Reply is then transmitted according to Section 10.2.5

#### 10.2.5. Join Reply Transmission

A Join Reply is transmitted to all of an ODMRP Router's neighbors, in order to achieve two objectives:

- o Set up or refresh the corresponding Forwarding Tuple for the upstream ODMRP neighbor
- o If the Join Reply was not originated by this router, acknowledge its reception to the previous hop

Before transmitting the Join Reply, the Information Base is updated as follows:

1. If the Pre-acknowledgement Set contains a tuple, such that:

- \* O\_multicast\_group = JR.MulticastGroupAddress
- \* O\_multicast\_source = JR.SourceAddress
- \* O\_seq\_num = JR.SequenceNumber
- \* O\_originator = JR.NextHopAddress

Then clear the JR.AckRequired flag, and set O\_exp\_time to EXPIRED

2. Otherwise, if the Pending Acknowledgement Set contains a Pending Tuple such as:

- \* P\_multicast\_group = JR.MulticastGroupAddress
- \* P\_multicast\_source = JR.SourceAddress
- \* P\_seq\_num = JR.SequenceNumber
- \* P\_next\_hop = JR.NextHopAddress

Then set JR.AckRequired, and increase P\_nth\_time by 1

3. Finally, if neither the Pre-acknowledgement Set nor the Pending Acknowledgement Set contain a corresponding tuple:

1. Insert a Pending Tuple in the Pending Acknowledgement Set, such as:

- + P\_multicast\_group = JR.MulticastGroupAddress
- + P\_multicast\_source = JR.SourceAddress
- + P\_seq\_num = JR.SequenceNumber
- + P\_next\_hop = JR.NextHopAddress
- + P\_nth\_time = 1
- + P\_acknowledged = FALSE
- + P\_expiration\_time = current-time + ACK\_TIMEOUT

2. Clear the JR.AckRequired flag

### 10.3. Forwarding Group Maintenance

While an ODMRP Router has data to send to a Multicast Group (on behalf of the Multicast Source), it MUST maintain the Forwarding Group generated by the initial Join Query. To this end, it MUST periodically generate JQ messages, according to Section 10.1.2. The interval between two Join Queries SHOULD be no less than ROUTE\_REFRESH\_INTERVAL. Note should be taken that, if the Multicast Session has no member other than the source, the Forwarding Group may contain only the designated ODMRP Router for the Multicast Source. That Router still needs to periodically flood Join Queries in order to rebuild a Forwarding Group if necessary.

### 10.4. Message Transmission

When using physical media subject to collisions and packet loss, both Join Query and Join Reply messages SHOULD be jittered to minimize the effect of collisions, as described in [RFC5148]

## 11. Unidirectional Links Handling

After sending a Join Reply, an ODMRP Router MUST verify that the upstream neighbor has joined the Forwarding Group. To this end, the following three mechanisms are used after transmitting a given Join Reply:

- o If the ODMRP Router overhears a corresponding Join Reply from the upstream neighbor (see Section 10.2.3), this verifies that the

link is bidirectional and that the upstream neighbor has joined the Forwarding Group (passive acknowledgement)

- o If the ODMRP Router has already overheard a corresponding Join Reply from the upstream neighbor prior to transmitting its own Join Reply, this means that the upstream neighbor has already joined the Forwarding Group (see Section 10.2.3) (pre-acknowledgement)
- o Otherwise, i.e., if neither the pre-acknowledgement nor the passive acknowledgement have verified that the upstream neighbor joined the Forwarding Group (i.e., if the corresponding Pending Tuple expires with P\_acknowledged set to False), then the ODMRP Router MUST proceed as follows:
  1. If the corresponding Pending tuple verifies `P_nth_time < JR_RETRIES`, then the ODMRP Router MUST retransmit the Join Reply with the `JR.AckRequired` flag set
  2. Otherwise, the link between the local interface and the interface of the upstream ODMRP Router identified by `JR.NextHopAddress` is considered unidirectional. In that case, the ODMRP Router SHOULD proceed as follows:
    - + Find the neighbor interface tuple such that `N_address_list` contains an address tuple with `i_addr = JR.NextHopAddress`, and set the variable `blacklisted-addresses` to the list of addresses contained in `N_address_list`
    - + Otherwise, if no such tuple exists, set the variable `blacklisted-addresses` to `[JR.NextHopAddress]`
    - + Add a tuple in the Blacklist such as:
      - `B_neighbor_interface_address_list := blacklisted-addresses`
      - `B_local_interface := P_local_interface`
      - `B_exp_time = current-time + BLACKLIST_TIMEOUT`

An ODMRP Router MAY attempt to use other mechanisms, such as [I-D.gerla-manet-odmrp-asym], to resume the Forwarding Group building process, instead of or in addition to using the Blacklist

## 12. SMF considerations

This protocol MAY be run in conjunction with SMF [RFC6621], and benefit from some of its features. In particular, if SMF is in use, it is RECOMMENDED that its duplicate packet detection feature described in Section 6 be used for multicast packet forwarding. Additionally, optimized flooding mechanisms, such as E-CDS or S-MPR, as specified in Appendices A through C of [RFC6621], MAY be used to flood Join Query messages throughout the network.

## 13. IGMP and MLD considerations

In order to determine whether or not it needs to reply to a Join Query message with a Join Reply message (as specified in Section 10.1.3), an ODMRP Router needs Multicast Group membership information. Such information can be provided by protocols such as IGMP [RFC3376] and/or MLD [RFC3810]. In particular, an ODMRP Router MUST reply with a Join Reply message to a valid Join Query messages advertising a Multicast Session if any of those conditions apply:

- o This Router is subscribed to the corresponding Multicast Group.
- o A host attached to this Router has signaled, for example using IGMP, that it has subscribed to the corresponding Multicast Group.

## 14. Multicast Packet Forwarding

ODMRP Routers originating and forwarding multicast packets MUST implement a duplicate packet detection (DPD) mechanism. If using IPv4 or IPV6 addresses, the use of SMF [RFC6621] is RECOMMENDED, as described in Section 12.

An ODMRP Router, receiving a non-duplicate multicast data packet, transmits it over all of its interfaces if it is a member of the forwarding group for this data packet, i.e., there exists a tuple in the Forwarding Group Table such as:

F\_multicast\_group correspond to the multicast address of this packet

F\_multicast\_source corresponds to the source of this packet

## 15. Security Considerations

As a multicast routing protocol, this protocol is potentially vulnerable to a number of attacks. This section attempts to describe the envisioned threats to the protocol, as well as some guidelines as to how to ensure confidentiality and integrity of the operations of ODMRP, and to mitigate threats of network overload.

This protocol relies on the use of a Duplicate Packet Detection (DPD) mechanism, such as one described in [RFC6621] (SMF), and suggests the use of optimized flooding to disseminate JQ messages. Some deployments of ODMRP are thus expected to function on top of [RFC6621] by taking advantage of the DPD and optimized flooding mechanisms provided by SMF. Such deployments are thus subject to the same security threats as SMF, such as those described in [I-D.ietf-manet-smf-threats].

### 15.1. Confidentiality

ODMRP routers which forward packets for multicast data source have to periodically transmit JQ messages throughout the network. In an unsecured network, an attacker could then eavesdrop on those messages and learn part or all of the network topology, depending on the traffic pattern.

### 15.2. Integrity

ODMRP relies on routers, in particular intermediate routers, to correctly transmit JQ and JR messages. An ODMRP router could, by malice or malfunction, originate JQ messages on behalf of a target multicast source with high enough sequence numbers to replace routing information in other routers. Such behavior would prevent the interested multicast receivers from receiving data packets sent by the target multicast source. An ODMRP router could also forward JQ messages with altered sequence numbers, thus preventing future routing updates. Both behaviors can be mitigated by end-to-end authentication of routing messages.

If NHDP [RFC6130] is not in use to update the Neighbor Interface Set, ODMRP relies on routers correctly informing their neighbors of the addresses they use via the JQ.LastAddress field. Upon transmission of a JQ message, an ODMRP router could, by malice or malfunction, set JQ.LastAddress to a network address that does not belong to this router (address spoofing). This could force neighbor ODMRP routers to blacklist this address in case the malicious router simulate unidirectional links by withholding JR messages. This behavior would break or slow down protocol convergence, potentially triggering data packet loss for multicast receivers. If NHDP is in use, the

deployment is subject to its own security vulnerabilities, such as those described in [RFC7186].

### 15.3. Channel Overload

ODMRP's main construct, the forwarding group, is built and maintained by having the source ODMRP router periodically flood JQ messages, which can be a costly operation in terms of bandwidth, processing and battery life, if applicable. A malicious router could flood JQ messages at a very high rate to overload the network. It is thus RECOMMENDED that ODMRP routers in a given deployment implement a rate-limit mechanism to prevent such behavior.

The efficiency (in terms of number of multicast data packets transmitted) of forwarding groups depends on routers actually sending JR messages only when necessary, in order to build a graph as sparse as possible and avoid redundant transmissions. Thus an ODMRP router which replies to JQ messages by transmitting one JR message for each of its known neighbors and with JR.NextHopAddress set to an address of this neighbor, would severely harm the efficiency of ODMRP by forcing the routers to build a forwarding group with unnecessary redundancy. Such behavior could also result in routing loops.

## 16. IANA Considerations

This specification defines two new Message Types, Join Query and Join Reply, which must be allocated from the "Message Type" repository of [RFC5444].

### 16.1. Join Query Registries

IANA is requested to create a registry for Message-Type-specific Message TLV Types for Join Query messages, with initial assignments according to Table 3.

Type	Description	Allocation Policy
128-223	Unassigned	Expert Review

Table 3: Join Query Message-Type-specific Message TLV Types

IANA is requested to create a registry for Message-Type-specific Address Block TLV Types for Join Query messages, with initial assignments according to Table 4.

Type	Description	Allocation Policy
128	ADDR-TYPE	
129-223	Unassigned	Expert Review

Table 4: Join Query Message-Type-specific Address Block TLV Types

Allocation of the ADDR-TYPE TLV from the Join Query specific Address Block TLV Types will create a new Type Extension Registry with initial assignments as specified in Table 5.

Name	Type	Type Ext.	Description	Al. Policy
ADDR-TYPE	128	0	MULTICAST-GROUP-ADDRESS	
ADDR-TYPE	128	1	LAST-ADDRESS	
ADDR-TYPE	128	2-255	Unassigned	Expert Review

Table 5: Address Block TLV Type assignment for ADDR-TYPE

## 16.2. Join Reply Registries

IANA is requested to create a registry for Message-Type-Specific Message TLV Types for Join Reply messages, with initial assignments according to Table 6.

Type	Description	Allocation Policy
128	ACKREQUIRED	
129-223	Unassigned	Expert Review

Table 6: Join Reply Message-Type-specific Message TLV Types

IANA is requested to create a registry for Message-Type-specific Address Block TLV Types for Join Reply messages, with initial assignments according to Table 7.



Type	Description	Allocation Policy
128	ADDR-TYPE	
129-223	Unassigned	Expert Review

Table 7: Join Reply Message-Type-specific Address Block TLV Types

Allocation of the ADDR-TYPE TLV from the Join Reply specific Address Block TLV Types will create a new Type Extension Registry with initial assignments as specified in Table 8.

Name	Type	Type Ext.	Description	Al. Policy
ADDR-TYPE	128	0	MULTICAST-GROUP-ADDRESS	
ADDR-TYPE	128	1	NEXT-HOP-ADDRESS	Expert Review
ADDR-TYPE	128	2-255	Unassigned	Expert Review

Table 8: Address Block TLV Type assignment for ADDR-TYPE

## 17. Acknowledgements

The authors would like to thank Thomas Clausen and Justin Dean for their insightful reviews and comments.

## 18. References

### 18.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", RFC 2119, BCP 14, March 1997.
- [RFC3376] Cain, B., Deering, S., Kouvelas, I., Fenner, B., and A. Thyagarajan, "Internet Group Management Protocol, Version 3", RFC 3376, October 2002.
- [RFC3810] Vida, R. and L. Costa, "Multicast Listener Discovery Version 2 (MLDv2) for IPv6", RFC 3810, June 2004.
- [RFC5148] Clausen, T., Dearlove, C., and B. Adamson, "Jitter

Considerations in Mobile Ad Hoc Networks (MANETs)",  
RFC 5148, February 2008.

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

[RFC6621] Macker, J., "Simplified Multicast Forwarding", RFC 6621,  
May 2012.

## 18.2. Informative References

[FGMP] Chiang, C., Gerla, M., and L. Zhang, "Forwarding Group  
Multicast Protocol (FGMP) for Multihop, Mobile Wireless  
Networks", Avril 1998.

[I-D.gerla-manet-odmrp-asym]  
Gerla, M., Oh, S., and A. Colin de Verdiere, "ODMRP\_ASYM",  
draft-gerla-manet-odmrp-asym-00 (work in progress).

[I-D.ietf-manet-smf-threats]  
Yi, J., Clausen, T., and U. Herberg, "Security Threats for  
Simplified Multicast Forwarding (SMF)",  
draft-ietf-manet-smf-threats-00 (work in progress),  
August 2014.

[ODMRP-Journal]  
Lee, S., Su, W., and M. Gerla, "On-Demand Multicast  
Routing Protocol in Multihop Wireless Networks",  
Journal of Mobile Networks and Applications, Volume 7  
Issue 6, Pages 441 - 453, December 2002.

[RFC3561] Perkins, C., Belding-Royer, E., and S. Das, "Ad hoc On-  
Demand Distance Vector (AODV) Routing", RFC 3561,  
July 2003.

[RFC6130] Clausen, T., Dearlove, C., and J. Dean, "Mobile Ad Hoc  
Network (MANET) Neighborhood Discovery Protocol (NHDP)",  
RFC 6130, April 2011.

[RFC7181] Clausen, T., Dearlove, C., Jacquet, P., and U. Herberg,  
"The Optimized Link State Routing Protocol Version 2",  
RFC 7181, April 2014.

[RFC7186] Yi, J., Herberg, U., and T. Clausen, "Security Threats for  
the Neighborhood Discovery Protocol (NHDP)", RFC 7186,  
April 2014.

## Appendix A. Illustrations

This section shows examples of ODMRP control messages encoded using [RFC5444]. [RFC5444] specifies that a packet is formed by a packet header, an optional TLV block and zero or more messages. This specification does not use or require any packet TLV. Additionally, the minimal packet header required by ODMRP is shown in Figure 1.

```

0
0 1 2 3 4 5 6 7
+---+---+---+---+---+---+
| Ver=0 | PF=0 |
+---+---+---+---+---+---+

```

Figure 1: Packet Header

### A.1. Join Query Message

JQ messages are instances of [RFC5444] messages. This section illustrates an example of one such message.

The JQ message's header's flag octet has a value of 9, meaning that the sequence number and source address fields are present, encoding respectively the sequence number and the address of the multicast source that originated the message. Additionally, the address length field (MAL) is set to 3, corresponding to an address length of 4 octets (i.e., the length of an IPv4 address). The overall message size is 23 octets.

An additional Message-Type specific address block is present, with one address and a flag octet (ABF) having value 0, meaning that the address block has no Head or Tail element. The Mid element encodes the Multicast group address. The associated TLV is of type ADDR-TYPE and value 0, i.e. MULTICAST-GROUP-ADDRESS.

The LastAddress element is omitted, meaning that the last JQ message from this interface was transmitted using the same address as this one.

```

      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
+-----+-----+-----+-----+-----+-----+-----+-----+
|  Join Query   | 1 0 0 1 | MAL=3 |      Message Length = 23      |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     Multicast Source Address      |
+-----+-----+-----+-----+-----+-----+-----+-----+
|      Message Sequence Number      |      TLVs length = 0      |
+-----+-----+-----+-----+-----+-----+-----+-----+
| Num Addrs = 1 |      ABF = 0      |      Multicast Group      ... |
+-----+-----+-----+-----+-----+-----+-----+-----+
| ...   Address                                     | Address TLV Block Length = 3 |
+-----+-----+-----+-----+-----+-----+-----+-----+
|      ADDR-TYPE   | 1 0 0 0 0 0 0 0 |      0      |
+-----+-----+-----+-----+-----+-----+-----+

```

#### A.2. Join Reply Message

JR messages are instances of [RFC5444] messages. This section illustrates an example of one such message.

The JR message's header's flag octet has a value of 9, meaning that the sequence number and source address fields are present, encoding respectively the sequence number and the address of the multicast source that originated the message. Additionally, the address length field (MAL) is set to 3, corresponding to an address length of 4 octets (i.e., the length of an IPv4 address). The overall message size is 34 octets.

Two additional Message-Type specific address blocks are present, both with one address and a flag octet (ABF) having value 0, meaning that the address block has no Head or Tail element. For the first address block, the Mid element encodes the Multicast group address; the associated Message-Type-specific TLV is of type ADDR-TYPE and value 0, i.e. MULTICAST-GROUP-ADDRESS. The second address block's Mid element encodes the Next Hop address; its associated Message-Type-specific TLV is of type ADDR-TYPE and value 1, i.e., NEXT-HOP-ADDRESS.

```

      0               1               2               3
      0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+-----+-----+-----+-----+
|  Join Reply  | 1 0 0 1 | MAL=3 |      Message Length = 34      |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     Multicast Source Address      |
+-----+-----+-----+-----+-----+-----+-----+-----+
|      Message Sequence Number      |      TLVs length = 0      |
+-----+-----+-----+-----+-----+-----+-----+-----+
| Num Addrs = 1 |      ABF = 0      |      Multicast Group      ... |
+-----+-----+-----+-----+-----+-----+-----+-----+
| ... Address                                     Address TLV Block Length = 3 |
+-----+-----+-----+-----+-----+-----+-----+-----+
|      ADDR-TYPE  | 1 0 0 0 0 0 0 0 |      0      | Num Addrs = 1 |
+-----+-----+-----+-----+-----+-----+-----+-----+
|      ABF = 0      |                                     Next Hop      ... |
+-----+-----+-----+-----+-----+-----+-----+-----+
| ... Address      |      Address TLV Block Length = 3 |      ADDR-TYPE      |
+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 0 0 0 0 0 0 0 |      1      |
+-----+-----+-----+-----+-----+-----+-----+

```

## Authors' Addresses

Yunjung Yi  
University of California, Los Angeles

Sung-Ju Lee  
University of California, Los Angeles

William Su  
The Boeing Company

Email: [william.su@boeing.com](mailto:william.su@boeing.com)

Mario Gerla  
University of California, Los Angeles  
3732F Boelter Hall  
Computer Science Department  
University of California  
Los Angeles, CA 90095-1596,  
USA

Phone: +1 310 825-4367  
Email: gerla@cs.ucla.edu

Axel Colin de Verdiere  
University of California, Los Angeles  
  
Email: axel@axelcdv.com



Mobile Ad hoc Networks Working Group  
Internet-Draft  
Intended status: Experimental  
Expires: November 4, 2016

C. Perkins  
Futurewei  
S. Ratliff  
Idirect  
J. Dowdell  
Airbus Defence and Space  
L. Steenbrink  
HAW Hamburg, Dept. Informatik  
V. Mercieca  
Airbus Defence and Space  
May 3, 2016

Ad Hoc On-demand Distance Vector Version 2 (AODVv2) Routing  
draft-ietf-manet-aodvv2-16

Abstract

The Ad Hoc On-demand Distance Vector Version 2 (AODVv2) routing protocol is intended for use by mobile routers in wireless, multihop networks. AODVv2 determines unicast routes among AODVv2 routers within the network in an on-demand fashion.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on November 4, 2016.

Copyright Notice

Copyright (c) 2016 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of



publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

1. Overview . . . . .	4
2. Terminology . . . . .	5
3. Applicability Statement . . . . .	9
4. Purpose of the Experiment . . . . .	11
5. Data Structures . . . . .	12
5.1. InterfaceSet . . . . .	12
5.2. Router Client Set . . . . .	12
5.3. Neighbor Set . . . . .	13
5.4. Sequence Numbers . . . . .	14
5.5. Local Route Set . . . . .	15
5.6. Multicast Route Message Set . . . . .	17
5.7. Route Error (RERR) Set . . . . .	19
6. Metrics . . . . .	19
7. AODVv2 Protocol Operations . . . . .	21
7.1. Initialization . . . . .	21
7.2. Next Hop Monitoring . . . . .	21
7.3. Neighbor Set Update . . . . .	23
7.4. Interaction with the Forwarding Plane . . . . .	24
7.5. Message Transmission . . . . .	26
7.6. Route Discovery, Retries and Buffering . . . . .	27
7.7. Processing Received Route Information . . . . .	28
7.7.1. Evaluating Route Information . . . . .	29
7.7.2. Applying Route Updates . . . . .	30
7.8. Suppressing Redundant Messages Using the Multicast Route Message Set . . . . .	33
7.9. Suppressing Redundant Route Error Messages using the Route Error Set . . . . .	35
7.10. Local Route Set Maintenance . . . . .	35
7.10.1. LocalRoute State Changes . . . . .	35
7.10.2. Reporting Invalid Routes . . . . .	38
8. AODVv2 Protocol Messages . . . . .	38
8.1. Route Request (RREQ) Message . . . . .	38
8.1.1. RREQ Generation . . . . .	40
8.1.2. RREQ Reception . . . . .	41
8.1.3. RREQ Forwarding . . . . .	42
8.2. Route Reply (RREP) Message . . . . .	42
8.2.1. RREP Generation . . . . .	43
8.2.2. RREP Reception . . . . .	45
8.2.3. RREP Forwarding . . . . .	46

8.3.	Route Reply Acknowledgement (RREP_Ack) Message . . . . .	47
8.3.1.	RREP_Ack Request Generation . . . . .	47
8.3.2.	RREP_Ack Reception . . . . .	48
8.3.3.	RREP_Ack Response Generation . . . . .	49
8.4.	Route Error (RERR) Message . . . . .	49
8.4.1.	RERR Generation . . . . .	50
8.4.2.	RERR Reception . . . . .	51
8.4.3.	RERR Regeneration . . . . .	53
9.	RFC 5444 Representation . . . . .	53
9.1.	Route Request Message Representation . . . . .	54
9.1.1.	Message Header . . . . .	55
9.1.2.	Message TLV Block . . . . .	55
9.1.3.	Address Block . . . . .	55
9.1.4.	Address Block TLV Block . . . . .	55
9.2.	Route Reply Message Representation . . . . .	56
9.2.1.	Message Header . . . . .	56
9.2.2.	Message TLV Block . . . . .	56
9.2.3.	Address Block . . . . .	57
9.2.4.	Address Block TLV Block . . . . .	57
9.3.	Route Reply Acknowledgement Message Representation . . . . .	58
9.3.1.	Message Header . . . . .	58
9.3.2.	Message TLV Block . . . . .	58
9.3.3.	Address Block . . . . .	58
9.3.4.	Address Block TLV Block . . . . .	58
9.4.	Route Error Message Representation . . . . .	58
9.4.1.	Message Header . . . . .	58
9.4.2.	Message TLV Block . . . . .	59
9.4.3.	Address Block . . . . .	59
9.4.4.	Address Block TLV Block . . . . .	59
10.	Simple External Network Attachment . . . . .	60
11.	Configuration . . . . .	62
11.1.	Timers . . . . .	62
11.2.	Protocol Constants . . . . .	64
11.3.	Local Settings . . . . .	65
11.4.	Network-Wide Settings . . . . .	65
11.5.	MetricType Allocation . . . . .	66
11.6.	RFC 5444 Message Type Allocation . . . . .	66
11.7.	RFC 5444 Message TLV Types . . . . .	66
11.8.	RFC 5444 Address Block TLV Type Allocation . . . . .	67
11.9.	ADDRESS_TYPE TLV Values . . . . .	67
12.	IANA Considerations . . . . .	68
13.	Security Considerations . . . . .	68
13.1.	Availability . . . . .	68
13.1.1.	Denial of Service . . . . .	68
13.1.2.	Malicious RERR messages . . . . .	69
13.1.3.	False Confirmation of Link Bidirectionality . . . . .	70
13.1.4.	Message Deletion . . . . .	71
13.2.	Confidentiality . . . . .	71

13.3. Integrity . . . . .	71
13.3.1. Message Insertion . . . . .	71
13.3.2. Message Modification - Man in the Middle . . . . .	72
13.3.3. Replay Attacks . . . . .	73
13.4. Protection Mechanisms . . . . .	73
13.4.1. Confidentiality and Authentication . . . . .	73
13.4.2. Integrity and Trust using ICVs . . . . .	73
13.4.3. Replay Protection using Timestamps . . . . .	73
13.4.4. Application to AODVv2 . . . . .	74
13.5. Key Management . . . . .	79
14. Acknowledgments . . . . .	81
15. References . . . . .	81
15.1. Normative References . . . . .	81
15.2. Informative References . . . . .	82
Appendix A. AODVv2 Draft Updates . . . . .	83
Authors' Addresses . . . . .	83

## 1. Overview

The Ad hoc On-Demand Distance Vector Version 2 (AODVv2) protocol enables dynamic, multihop routing between participating mobile nodes wishing to establish and maintain an ad hoc network. The basic operations of the AODVv2 protocol are route discovery and route maintenance. AODVv2 does not require nodes to maintain routes to destinations that are not in active communication. AODVv2 allows mobile nodes to respond to link breakages and changes in network topology in a timely manner. The operation of AODVv2 is loop-free, and by avoiding the Bellman-Ford "counting to infinity" problem offers quick convergence when the ad hoc network topology changes (typically, when a node moves in the network). When links break, AODVv2 causes the affected set of nodes to be notified so that they are able to invalidate the routes using the lost link.

One distinguishing feature of AODVv2 is its use of a destination sequence number for each route entry. The destination sequence number is created by the destination to be included along with any route information it sends to requesting nodes. Using destination sequence numbers ensures loop freedom and is simple to program. Given the choice between two routes to a destination, a requesting node is required to select the one with the greatest sequence number.

Compared to AODV [RFC3561], AODVv2 has moved some features out of the scope of the document, notably intermediate route replies, expanding ring search, route lifetimes and precursor lists. However, the document has been designed to allow their specification in a separate document. Hello messages and local repair have been removed. AODVv2 provides a mechanism for the use of multiple metric types. Message formats have been updated and made compliant with [RFC5444]. AODVv2

control messages are defined as sets of data, which are mapped to message elements using the Generalized MANET Packet/Message Format defined in [RFC5444] and sent using the parameters in [RFC5498]. Verification of link bidirectionality has been substantially improved, and additional refinements made for route timeouts and state management.

The basic protocol mechanisms are as follows. Since AODVv2 is a reactive protocol, route discovery is initiated only when a route to the target is needed (i.e. when a router's client wants to send data). AODVv2 does this with the help of a Route Request (RREQ) and Route Reply (RREP) cycle: an RREQ is distributed across the network until it arrives at the target. When forwarding an RREQ, all routers across the network store the neighbor they've received the RREQ from, memorizing a possible route back to the originator of the RREQ. When the target receives the RREQ, it answers with an RREP, which then travels back to the originator along the path memorized by the intermediate routers. A metric value is included within the messages to record the cost of the route. AODVv2 uses sequence numbers to identify stale routing information, and compares route metric values to determine if advertised routes could form loops.

Route maintenance includes confirming bidirectionality of links to next hop AODVv2 routers and issuing Route Error (RERR) messages informing other routers of broken links. It also includes reacting to received Route Error messages, and extending and enforcing route timeouts.

The on-demand nature of AODVv2 requires signals to be exchanged between AODVv2 and the forwarding plane. These signals indicate when: \* a packet is to be forwarded, in order to initiate route discovery \* packet forwarding fails, in order to initiate route error reporting \* a packet is successfully forwarded, for route maintenance.

Security for authentication of AODVv2 routers and encryption of control messages is accomplished using the TIMESTAMP and ICV TLVs defined in [RFC7182].

## 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]. In addition, this document uses terminology from [RFC5444], and defines the following terms:

AddressList

A list of IP addresses as used in AODVv2 messages.

**AckReq**

Used in a Route Reply Acknowledgement message to indicate that a Route Reply Acknowledgement is expected in return.

**AdvRte**

A route advertised in an incoming route message.

**AODVv2 Router**

An IP addressable device in the ad hoc network that performs the AODVv2 protocol operations specified in this document.

**CurrentTime**

The current time as maintained by the AODVv2 router.

**ENAR (External Network Access Router)**

An AODVv2 router with an interface to an external, non-AODVv2 network.

**InterfaceSet**

The set of all network interfaces supporting AODVv2.

**Invalid route**

A route that cannot be used for forwarding but still contains useful sequence number information.

**LocalRoute**

An entry in the Local Route Set as defined in Section 5.5.

**MANET**

A Mobile Ad Hoc Network as defined in [RFC2501].

**MetricType**

The metric type for a metric value included in a message.

**MetricTypeList**

A list of metric types associated with the addresses in the AddressList of a Route Error message.

**Neighbor**

An AODVv2 router from which an RREQ or RREP message has been received. Neighbors exchange routing information and verify bidirectionality of the link to a neighbor before installing a route via that neighbor into the Local Route Set.

**OrigAddr**

The source IP address of the IP packet triggering route discovery.

**OrigMetric**

The metric value associated with the route to OrigPrefix.

**OrigPrefix**

The prefix configured in the Router Client entry which includes OrigAddr.

**OrigPrefixLen**

The prefix length, in bits, configured in the Router Client entry which includes OrigAddr.

**OrigSeqNum**

The sequence number of the AODVv2 router which originated the Route Request on behalf of OrigAddr.

**PktSource**

The source address of the IP packet that triggered a Route Error message.

**PrefixLengthList**

A list of routing prefix lengths associated with the addresses in the AddressList of a message.

**Reactive**

Performed only in reaction to specific events. In AODVv2, routes are requested only when data packets need to be forwarded. In this document, "reactive" is synonymous with "on-demand".

**RERR (Route Error)**

The AODVv2 message type used to indicate that an AODVv2 router does not have a valid LocalRoute toward one or more particular destinations.

**RERR\_Gen (RERR Generating Router)**

The AODVv2 router generating a Route Error message.

**RerrMsg (RERR Message)**

A Route Error (RERR) message.

**Routable Unicast IP Address**

A routable unicast IP address is a unicast IP address that is scoped sufficiently to be forwarded by a router. Globally-scoped unicast IP addresses and Unique Local Addresses (ULAs) [RFC4193] are examples of routable unicast IP addresses.

**Router Client**

An address or address range configured on an AODVv2 router, on behalf of which that router will initiate and respond to route

discoveries. These addresses may be used by the AODVv2 router itself or by its Router Clients that are reachable without traversing another AODVv2 router.

**RREP (Route Reply)**

The AODVv2 message type used to reply to a Route Request message.

**RREP\_Gen (RREP Generating Router)**

The AODVv2 router that generates the Route Reply message, i.e., the router configured with TargAddr as a Router Client.

**RREQ (Route Request)**

The AODVv2 message type used to discover a route to TargAddr and distribute information about a route to OrigPrefix.

**RREQ\_Gen (RREQ Generating Router)**

The AODVv2 router that generates the Route Request message, i.e., the router configured with OrigAddr as a Router Client.

**RteMsg (Route Message)**

A Route Request (RREQ) or Route Reply (RREP) message.

**SeqNum**

The sequence number maintained by an AODVv2 router to indicate freshness of route information.

**SeqNumList**

A list of sequence numbers associated with the addresses in the AddressList of a message.

**TargAddr**

The target address of a route request, i.e., the destination address of the IP packet triggering route discovery.

**TargMetric**

The metric value associated with the route to TargPrefix.

**TargPrefix**

The prefix configured in the Router Client entry which includes TargAddr.

**TargPrefixLen**

The prefix length, in bits, configured in the Router Client entry which includes TargAddr.

**TargSeqNum**

The sequence number of the AODVv2 router which originated the Route Reply on behalf of TargAddr.

**Unreachable Address**

An address reported in a Route Error message, as described in Section 8.4.1.

**Upstream**

In the direction from destination to source (from TargAddr to OrigAddr).

**Valid route**

A route that can be used for forwarding, as described in Section 8.4.1.

This document uses the notational conventions in Table 1 to simplify the text.

Notation	Meaning
Route[Address]	A route toward Address
Route[Address].Field	A field in a route toward Address
RteMsg.Field	A field in either RREQ or RREP

Table 1: Notational Conventions

### 3. Applicability Statement

The AODVv2 routing protocol is a reactive routing protocol intended for use in mobile ad hoc wireless networks. A reactive protocol only sends messages to discover a route when there is data to send on that route. Therefore, a reactive routing protocol requires certain interactions with the forwarding plane (for example, to indicate when a packet is to be forwarded, in order to initiate route discovery). The set of signals exchanged between AODVv2 and the forwarding plane are discussed in Section 7.4.

AODVv2 is designed for stub or disconnected mobile ad hoc networks, i.e., non-transit networks or those not connected to the internet. AODVv2 can, however, be configured to perform gateway functions when attached to external networks, as discussed in Section 10.

AODVv2 handles a wide variety of mobility and traffic patterns by determining routes on-demand. In networks with a large number of routers, AODVv2 is best suited for relatively sparse traffic scenarios where each router forwards IP packets to a small percentage of other AODVv2 routers in the network. In this case fewer routes are needed, and therefore less control traffic is produced. In large networks with very frequent or bursty traffic, AODVv2 control



messages may cause a broadcast storm, overwhelming the network with control messages and preventing routes from being established. This especially applies to networks with point-to-point or point-to-multipoint traffic. In this case, the transmission priorities described in Section 7.5 prioritize route maintenance traffic over route discovery traffic.

Data packets may be buffered until a route to their destination is available, as described in Section 7.6.

AODVv2 provides for message integrity and security against replay attacks by using integrity check values, timestamps and sequence numbers, as described in Section 13. If security associations can be established, encryption can be used for AODVv2 messages to ensure that only trusted routers participate in routing operations.

Since the route discovery process aims for a route to be established in both directions along the same path, uni-directional links are not suitable. AODVv2 will detect and exclude those links from route discovery. The route discovered is optimised for the requesting router, and the return path may not be the optimal route.

AODVv2 is applicable to memory constrained devices, since only a little routing state is maintained in each AODVv2 router. AODVv2 routes that are not needed for forwarding data do not need to be maintained. On routers unable to store persistent AODVv2 state, recovery can impose a performance penalty (e.g., in case of AODVv2 router reboot), since if a router loses its sequence number, there is a delay before the router can resume full operations. This is described in Section 7.1.

AODVv2 supports routers with multiple interfaces and multiple IP addresses per interface. A router may also use the same IP address on multiple interfaces. AODVv2 requires only that each interface configured for AODVv2 has at least one unicast IP address. Address assignment procedures are out of scope for AODVv2.

AODVv2 supports Router Clients with multiple interfaces, as long as each interface is configured with its own unicast IP address. Multi-homing of a Router Client IP address is not supported by AODVv2, and therefore an IP address SHOULD NOT be configured as a Router Client on more than one AODVv2 router at any one time.

The routing algorithm in AODVv2 MAY be operated at layers other than the network layer, using layer-appropriate addresses.

#### 4. Purpose of the Experiment

AODVv2 is an Experimental protocol. While it is based on AODV [RFC3561], important protocol mechanisms have changed: \*

- \* Bidirectionality is ensured using a new mechanism
- \* Alternate metrics may be used to determine route quality
- \* Support for multiple interfaces has been improved
- \* Support for multi-interface IP addresses has been added
- \* A new security model allowing end to end integrity checks has been added
- \* A new message format ([RFC5444]) is used.

Many of these changes have been made quite recently, after a protocol development hiatus of several years.

Thus, the purpose of the experiment is to gain information on the behavior of these significant changes in real-world deployments, not only to learn about AODVv2 in particular, but also to further the knowledge base of reactive protocols in general.

Suitable future experiments could be:

- o Evaluation of the new features mentioned above with regard to performance and functionality
- o determining default values for configuration parameters such as timeouts, numbers of retries, buffer sizes, control message limits (ensuring the level of multicast traffic does not interfere with data traffic throughput)
- o specification of optimisations / verification of minimum requirements for low-power or low-memory routers
- o developing security strategies for different environments
- o Quantification of effectiveness and performance of precursors
- o Evaluation of different metric types and their suitability for reactive distance vector protocols
- o Evaluation of use of an AODVv2 router as an External Network Attached Router or gateway router, including network topologies including multiple gateways.
- o Achieving implementations
- o multiple and interoperable
- o deployments in different network types

- o Analysis of the effects of buffering traffic while route discovery is in progress
- o Specification of extensions to deal with timed routes, expanding ring multicast, unicast RERR to specific route precursors, accurate bidirectional metric discovery, dealing with and allowing uni-directional links and routes

The final goal of the experiment is to determine if sufficient demand exists for the AODVv2 protocol to prompt an effort to bring the protocol to Standards Track.

## 5. Data Structures

### 5.1. InterfaceSet

The InterfaceSet is a conceptual data structure which contains information about all interfaces configured for use by AODVv2. Any interface with an IP address can be used. Multiple interfaces on a single router can be used. Multiple interfaces on the same router may be configured with the same IP address.

Each element in the InterfaceSet MUST contain the following:

#### Interface.Id

An identifier that is unique in node-local scope and that allows the AODVv2 implementation to identify exactly one local network interface.

If multiple interfaces of the AODVv2 router are configured for use by AODVv2, they MUST be configured in the InterfaceSet.

Implementations for constrained devices using only one interface MAY choose not to use the InterfaceSet.

### 5.2. Router Client Set

An AODVv2 router provides route discovery services for its own local applications and for its Router Clients that are reachable without traversing another AODVv2 router. The addresses used by these devices, and the AODVv2 router itself, are configured in the Router Client Set. An AODVv2 router will only originate Route Request and Route Reply messages on behalf of configured Router Client addresses.

Router Client Set entries MUST contain:

RouterClient.IPAddress

An IP address or the start of an address range that requires route discovery services from the AODVv2 router.

**RouterClient.PrefixLength**

The length, in bits, of the routing prefix associated with the RouterClient.IPAddress. If the prefix length is not equal to the address length of RouterClient.IPAddress, the AODVv2 router MUST participate in route discovery on behalf of all addresses within that prefix.

**RouterClient.Cost**

The cost associated with reaching this address or address range.

A Router Client address MUST NOT be served by more than one AODVv2 router at any one time. To shift responsibility for a Router Client to a different AODVv2 router, correct AODVv2 routing behavior MUST be observed; The AODVv2 router adding the Router Client MUST wait for any existing routing information about this Router Client to be purged from the network, i.e., at least MAX\_SEQNUM\_LIFETIME since the last SeqNum update on the router that is removing this Router Client.

### 5.3. Neighbor Set

A Neighbor Set MUST be maintained with information about neighboring AODVv2 routers. Neighbor Set entries are stored when AODVv2 messages are received. If the Neighbor is chosen as a next hop on an installed route, the link to the Neighbor MUST be tested for bidirectionality and the result stored in this set. A route will only be considered valid when the link is confirmed to be bidirectional.

Neighbor Set entries MUST contain:

**Neighbor.IPAddress**

An IP address of the neighboring router, learned from the source IP address of a received route message.

**Neighbor.State**

Indicates whether the link to the neighbor is bidirectional. There are three possible states: Confirmed, Heard, and Blacklisted. Heard is the initial state. Confirmed indicates that the link to the neighbor has been confirmed as bidirectional. Blacklisted indicates that the link to the neighbor is unidirectional. Section 7.2 discusses how to monitor link bidirectionality.

**Neighbor.Timeout**

Indicates at which time the Neighbor.State should be updated:

- o If the value of Neighbor.State is Blacklisted, this indicates the time at which Neighbor.State will revert to Heard. By default this value is calculated at the time the router is blacklisted and is equal to CurrentTime + MAX\_BLACKLIST\_TIME.
- o If Neighbor.State is Heard, and an RREP\_Ack has been requested from the neighbor, it indicates the time at which Neighbor.State will be set to Blacklisted, if an RREP\_Ack has not been received.
- o If the value of Neighbor.State is Heard and no RREP\_Ack has been requested, or if Neighbor.State is Confirmed, this time is set to INFINITY\_TIME.

Neighbor.Interface

The interface on which the link to the neighbor was established.

Neighbor.AckSeqNum

The next sequence number to use for the TIMESTAMP value in an RREP\_Ack request, in order to detect replay of an RREP\_Ack response. Initially set to a random value.

Neighbor.HeardRERRSeqNum

The last heard sequence number used as the TIMESTAMP value in a RERR received from this neighbor, saved in order to detect replay of a RERR message. Initially set to zero.

See Section 13.4.4.3 and Section 13.4.4.4 for more information on how Neighbor.AckSeqNum and Neighbor.HeardRERRSeqNum are used.

#### 5.4. Sequence Numbers

Sequence Numbers enable AODVv2 routers to determine the temporal order of route discovery messages, identifying stale routing information so that it can be discarded. The sequence number fulfills the same roles as the "Destination Sequence Number" of DSDV [Perkins94], and the AODV Sequence Number in [RFC3561].

Each AODVv2 router in the network MUST maintain its own sequence number. All RREQ and RREP messages created by an AODVv2 router include the router's sequence number, reported as a 16-bit unsigned integer. Each AODVv2 router MUST ensure that its sequence number is strictly increasing, and that it is incremented by one (1) whenever an RREQ or RREP is created, except when the sequence number is 65,535 (the maximum value of a 16-bit unsigned integer), in which case it MUST be reset to one (1) to achieve wrap around. The value zero (0) is reserved to indicate that the sequence number is unknown.

An AODVv2 router **MUST** only attach its own sequence number to information about a route to one of its configured Router Clients, all route messages forwarded by other routers retain the originator's sequence number.

To determine if newly received information is stale and therefore redundant, the sequence number attached to the information is compared to the sequence number of existing information about the same route. The comparison is carried out by subtracting the existing sequence number from the newly received sequence number, using unsigned arithmetic. The result of the subtraction is to be interpreted as a signed 16-bit integer.

- o If the result is negative, the newly received information is considered older than the existing information and is considered stale and redundant and **MUST** therefore be discarded.
- o If the result is positive, the newly received information is considered newer than the existing information and is not considered stale or redundant and **MUST** therefore be processed.
- o If the result is zero, the newly received information is not considered stale, and therefore **MUST** be processed further to determine if it is redundant. For example, it is considered redundant if the metric attached to the newly received information is higher than the metric of existing information about the same route (see Section 7.7.1 and Section 7.8).

This, along with the processes in Section 7.7.1, ensures loop freedom.

An AODVv2 router **SHOULD** maintain its sequence number in persistent storage. If the sequence number is lost, the router **MUST** follow the procedure in Section 7.1 to safely resume routing operations with a new sequence number.

## 5.5. Local Route Set

All AODVv2 routers **MUST** maintain a Local Route Set, containing information about routes learned from AODVv2 route messages. The Local Route Set is stored separately from the forwarding plane's routing table (referred to as Routing Information Base (RIB)), which may be updated by other routing protocols operating on the AODVv2 router as well. The Routing Information Base is updated using information from the Local Route Set. Alternatively, implementations **MAY** choose to modify the Routing Information Base directly.

Routes learned from AODVv2 route messages are referred to in this document as LocalRoutes, and MUST contain the following information:

LocalRoute.Address

An address, which, when combined with LocalRoute.PrefixLength, describes the set of destination addresses this route includes.

LocalRoute.PrefixLength

The prefix length, in bits, associated with LocalRoute.Address.

LocalRoute.SeqNum

The sequence number associated with LocalRoute.Address, obtained from the last route message that successfully updated this entry.

LocalRoute.NextHop

The source IP address of the IP packet containing the AODVv2 message advertising the route to LocalRoute.Address, i.e. an IP address of the AODVv2 router used for the next hop on the path toward LocalRoute.Address.

LocalRoute.NextHopInterface

The interface used to send IP packets toward LocalRoute.Address.

LocalRoute.LastUsed

If this route is installed in the Routing Information Base, the time it was last used to forward an IP packet.

LocalRoute.LastSeqNumUpdate

The time LocalRoute.SeqNum was last updated.

LocalRoute.MetricType

The type of metric associated with this route.

LocalRoute.Metric

The cost of the route toward LocalRoute.Address expressed in units consistent with LocalRoute.MetricType.

LocalRoute.State

The last known state (Unconfirmed, Idle, Active, or Invalid) of the route.

There are four possible states for a LocalRoute:

Unconfirmed

A route learned from a Route Request message, which has not yet been confirmed as bidirectional. It MUST NOT be used for forwarding IP packets, and therefore it is not referred to as a

valid route. This state only applies to routes learned through RREQ messages.

#### Idle

A route which has been learned from a route message, and has also been confirmed, but has not been used in the last ACTIVE\_INTERVAL. It is able to be used for forwarding IP packets, and therefore it is referred to as a valid route.

#### Active

A route which has been learned from a route message, and has also been confirmed, and has been used in the last ACTIVE\_INTERVAL. It is able to be used for forwarding IP packets, and therefore it is referred to as a valid route.

#### Invalid

A route which has expired or been lost. It MUST NOT be used for forwarding IP packets, and therefore it is not referred to as a valid route. Invalid routes contain sequence number information which allows incoming information to be assessed for freshness.

When the Local Route Set is stored separately from the Routing Information Base, routes are added to the Routing Information Base when LocalRoute.State is valid (set to Active or Idle), and removed from the Routing Information Base when LocalRoute.State becomes Invalid.

Changes to LocalRoute state are detailed in Section 7.10.1.

Multiple valid routes for the same address and prefix length but for different metric types may exist in the Local Route Set, but the decision of which of these routes to install in the Routing Information Base to use for forwarding is outside the scope of AODVv2.

### 5.6. Multicast Route Message Set

Route Request (RREQ) messages are multicast by default and forwarded multiple times. This set stores recently received RREQs in order that received RREQs can be tested for redundancy to avoid unnecessary processing and forwarding.

The Multicast Route Message Set is a conceptual set which contains information about previously received multicast route messages, so that incoming route messages can be compared with previously received messages to determine if the incoming information is redundant or stale, and the router can avoid sending redundant control traffic.



Multicast Route Message Set entries MUST contain the following information:

RteMsg.OrigPrefix

The prefix associated with OrigAddr, the source address of the IP packet triggering the route request.

RteMsg.OrigPrefixLen

The prefix length associated with RteMsg.OrigPrefix, originally from the Router Client entry on RREQ\_Gen which includes OrigAddr.

RteMsg.TargPrefix

The prefix associated with TargAddr, the destination address of the IP packet triggering the route request. In an RREQ this MUST be set to TargAddr.

RteMsg.OrigSeqNum

The sequence number associated with the route to OrigPrefix, if RteMsg is an RREQ.

RteMsg.TargSeqNum

The sequence number associated with the route to TargPrefix.

RteMsg.MetricType

The metric type of the route requested.

RteMsg.Metric

The metric value received in the RteMsg.

RteMsg.Timestamp

The last time this Multicast Route Message Set entry was updated.

RteMsg.RemoveTime

The time at which this entry MUST be removed from the Multicast Route Message Set. This is set to CurrentTime + MAX\_SEQNUM\_LIFETIME, whenever the RteMsg.OrigSeqNum of this entry is updated.

RteMsg.Interface

The interface on which the message was received.

The Multicast Route Message Set is maintained so that no two entries have the same OrigPrefix, OrigPrefixLen, TargPrefix, and MetricType. See Section 7.8 for details about updating this set.

### 5.7. Route Error (RERR) Set

Each RERR message sent because no route exists for packet forwarding SHOULD be recorded in a conceptual set called the Route Error (RERR) Set. Each entry contains the following information:

RerrMsg.Timeout

The time after which the entry SHOULD be deleted.

RerrMsg.UnreachableAddress

The UnreachableAddress reported in the AddressList of the RERR.

RerrMsg.PktSource:

The PktSource of the RERR.

See section Section 7.9 for instructions on how to update the set.

## 6. Metrics

Metrics measure a cost or quality associated with a route or a link, e.g., latency, delay, financial cost, energy, etc. Metric values are reported in Route Request and Route Reply messages.

In Route Request messages, the metric describes the cost of the route from OrigPrefix to the router sending the Route Request. For RREQ\_Gen, this is the cost associated with the Router Client entry which includes OrigAddr. For routers which forward the RREQ, this is the cost from OrigPrefix to the forwarding router, combining the metric value from the received RREQ message with knowledge of the link cost from the sender to the receiver, i.e., the incoming link cost. This updated route cost is included when forwarding the Route Request message, and used to install a route to OrigPrefix.

Similarly, in Route Reply messages, the metric reflects the cost of the route from TargPrefix to the router sending the Route Reply. For RREP\_Gen, this is the cost associated with the Router Client entry which includes TargAddr. For routers which forward the RREP, this is the cost from TargPrefix to the forwarding router, combining the metric value from the received RREP message with knowledge of the link cost from the sender to the receiver, i.e., the incoming link cost. This updated route cost is included when forwarding the Route Reply message, and used to install a route to TargPrefix.

Assuming link metrics are symmetric, the cost of the routes installed in the Local Route Set at each router will be correct. While this assumption is not always correct, calculating incoming/outgoing metric data is outside of scope of this document. The route

discovered is optimised for the requesting router, and the return path may not be the optimal route.

AODVv2 enables the use of multiple metric types. Each route discovery attempt indicates the metric type which is requested for the route. Only one metric type MUST be used in each route discovery attempt.

For each MetricType, AODVv2 requires:

- o A MetricType number, to indicate the metric type of a route. MetricType numbers allocated are detailed in Section 11.5.
- o A maximum value, denoted MAX\_METRIC[MetricType]. This MUST always be the maximum expressible metric value of type MetricType. Field lengths associated with metric values are found in Section 11.5. If the cost of a route exceeds MAX\_METRIC[MetricType], the route is ignored.
- o A function for incoming link cost, denoted Cost(L). Using incoming link costs means that the route learned has a path optimized for the direction from OrigAddr to TargAddr.
- o A function for route cost, denoted Cost(R).
- o A function to analyze routes for potential loops based on metric information, denoted LoopFree(R1, R2). LoopFree verifies that a route R2 is not a sub-section of another route R1. An AODVv2 router invokes LoopFree() as part of the process in Section 7.7.1, when an advertised route (R1) and an existing LocalRoute (R2) have the same destination address, metric type, and sequence number. LoopFree returns FALSE to indicate that an advertised route is not to be used to update a stored LocalRoute, as it may cause a routing loop. In the case where the existing LocalRoute is Invalid, it is possible that the advertised route includes the existing LocalRoute and came from a router which did not yet receive notification of the route becoming Invalid, so the advertised route should not be used to update the Local Route Set, in case it forms a loop to a broken route.

AODVv2 currently supports cost metrics where Cost(R) is strictly increasing, by defining:

- o  $\text{Cost}(R) := \text{Sum of Cost}(L) \text{ of each link in the route}$
- o  $\text{LoopFree}(R1, R2) := ( \text{Cost}(R1) <= \text{Cost}(R2) )$

Implementers MAY consider other metric types, but the definitions of Cost and LoopFree functions for such types are undefined, and interoperability issues need to be considered.

## 7. AODVv2 Protocol Operations

The AODVv2 protocol's operations include managing sequence numbers, monitoring next hop AODVv2 routers on discovered routes and updating the Neighbor Set, performing route discovery and dealing with requests from other routers, processing incoming route information and updating the Local Route Set, updating the Multicast Route Message Set and suppressing redundant messages, and reporting broken routes. These processes are discussed in detail in the following sections.

### 7.1. Initialization

During initialization where an AODVv2 router does not have information about its previous sequence number, or if its sequence number is lost at any point, the router resets its sequence number to one (1). However, other AODVv2 routers may still hold sequence number information that this router previously issued. Since sequence number information is removed if there has been no update to the sequence number in MAX\_SEQNUM\_LIFETIME, the initializing router MUST wait for MAX\_SEQNUM\_LIFETIME before it creates any messages containing its new sequence number. It can then be sure that the information it sends will not be considered stale.

During this wait period, the router is permitted to do the following:

- o Process information in a received RREQ or RREP message to learn a route to the originator or target of that route discovery
- o Forward a received RREQ or RREP
- o Send an RREP\_Ack
- o Maintain valid routes in the Local Route Set
- o Create, process and forward RERR messages

### 7.2. Next Hop Monitoring

To ensure AODVv2 routers do not establish routes over unidirectional links, AODVv2 routers MUST verify that the link to the next hop router is bidirectional before marking a route as valid in the Local Route Set.

AODVv2 provides a mechanism for testing bidirectional connectivity during route discovery, and blacklisting routers where bidirectional connectivity is not available. If a route discovery is retried by RREQ\_Gen, the blacklisted routers can be excluded from the process, and a different route can be discovered. Further, a route is not to be used for forwarding until the bidirectionality of the link to the next hop is confirmed. AODVv2 routers do not need to monitor bidirectionality for links to neighboring routers which are not used as next hops on routes in the Local Route Set.

- o Bidirectional connectivity to upstream routers is tested by requesting acknowledgement of RREP messages by also sending an RREP\_Ack, including an AckReq element to indicate that an acknowledgement is requested. This MUST be answered by sending an RREP\_Ack in response. Receipt of an RREP\_Ack within RREP\_Ack\_SENT\_TIMEOUT proves that bidirectional connectivity exists. Otherwise, a link is determined to be unidirectional. All AODVv2 routers MUST support this process, which is explained in Section 8.2 and Section 8.3.
- o For the downstream router, receipt of an RREP message containing the route to TargAddr is confirmation of bidirectionality, since an RREP message is a reply to a RREQ message which previously crossed the link in the opposite direction.

To assist with next hop monitoring, a Neighbor Set (Section 5.3) is maintained. When an RREQ or RREP is received, search for an entry in the Neighbor Set where all of the following conditions are met:

- o Neighbor.IPAddress == IP address from which the RREQ or RREP was received
- o Neighbor.Interface == Interface on which the RREQ or RREP was received.

If such an entry does not exist, a new entry is created as described in Section 7.3. While the value of Neighbor.State is Heard, acknowledgement of RREP messages sent to that neighbor MUST be requested. If an acknowledgement is not received within the timeout period, the neighbor MUST have Neighbor.State set to Blacklisted. If an acknowledgement is received within the timeout period, Neighbor.State is set to Confirmed. While the value of Neighbor.State is Confirmed, the request for an acknowledgement of any other RREP message is unnecessary.

When routers perform other operations such as those from the list below, these MAY be used as additional indications of connectivity:

- o NHDP HELLO Messages [RFC6130]
- o Route timeout
- o Lower layer triggers, e.g. message reception or link status notifications
- o TCP timeouts
- o Promiscuous listening
- o Other monitoring mechanisms or heuristics

If such an external process signals that the link to a neighbor is bidirectional, the AODVv2 router MAY update the matching Neighbor Set entry by changing the value of Neighbor.State to Confirmed, e.g. receipt of a Neighborhood Discovery Protocol HELLO message with the receiving router listed as a neighbor. If an external process signals that a link is not bidirectional, the the value of Neighbor.State MAY be changed to Blacklisted, e.g. notification of a TCP timeout.

### 7.3. Neighbor Set Update

On receipt of an RREQ or RREP message, the Neighbor Set MUST be checked for an entry with Neighbor.IPAddress which matches the source IP address of a packet containing the AODVv2 message. If no matching entry is found, a new entry is created.

A new Neighbor Set entry is created as follows:

- o Neighbor.IPAddress := Source IP address of the received route message
- o Neighbor.State := Heard
- o Neighbor.Timeout := INFINITY\_TIME
- o Neighbor.Interface := Interface on which the RREQ or RREP was received. MUST equal Interface.Id of one of the entries in the InterfaceSet (see Section 5.1).

When an RREP\_Ack is sent to a neighbor, the Neighbor Set entry is updated as follows:

- o Neighbor.Timeout := CurrentTime + RREP\_Ack\_SENT\_TIMEOUT

When a received message is one of the following:

- o an RREP which answers an RREQ sent within RREQ\_WAIT\_TIME over the same interface as Neighbor.Interface
- o an RREP\_Ack response received from a Neighbor with Neighbor.State set to Heard, where Neighbor.Timeout > CurrentTime

the link to the neighbor is bidirectional and the Neighbor Set entry is updated as follows:

- o Neighbor.State := Confirmed
- o Neighbor.Timeout := INFINITY\_TIME

When the Neighbor.Timeout is reached and Neighbor.State is Heard, then an RREP\_Ack response has not been received from the neighbor within RREP\_Ack\_SENT\_TIMEOUT of sending the RREP\_Ack request. The link is considered to be uni-directional and the Neighbor Set entry is updated as follows:

- o Neighbor.State := Blacklisted
- o Neighbor.Timeout := CurrentTime + MAX\_BLACKLIST\_TIME

When the Neighbor.Timeout is reached and Neighbor.State is Blacklisted, the Neighbor Set entry is updated as follows:

- o Neighbor.State := Heard

If an external mechanism reports a link as broken, the Neighbor Set entry SHOULD be removed.

Route requests from neighbors with Neighbor.State set to Blacklisted are ignored to avoid persistent IP packet loss or protocol failures. Neighbor.Timeout allows the neighbor to again be allowed to participate in route discoveries after MAX\_BLACKLIST\_TIME, in case the link between the routers has become bidirectional.

#### 7.4. Interaction with the Forwarding Plane

The signals described in the following are conceptual signals, and can be implemented in various ways. Conformant implementations of AODVv2 are not mandated to implement the forwarding plane separately from the control plane or data plane; these signals and interactions are identified simply as assistance for implementers who may find them useful.

AODVv2 requires signals from the forwarding plane:

- o A packet cannot be forwarded because a route is unavailable: AODVv2 needs to know the source and destination IP addresses of the packet. If the source of the packet is configured as a Router Client, the router should initiate route discovery to the destination. If it is not a Router Client, the router should create a Route Error message.
- o A packet is to be forwarded: AODVv2 needs to check the state of the route to ensure it is still valid.
- o Packet forwarding succeeds: AODVv2 needs to update the record of when a route was last used to forward a packet.
- o Packet forwarding failure occurs: AODVv2 needs to create a Route Error message.

AODVv2 needs to send signals to the forwarding plane:

- o A route discovery is in progress: buffering might be configured for packets requiring a route, while route discovery is attempted.
- o A route discovery failed: any buffered packets requiring that route should be discarded, and the source of the packet should be notified that the destination is unreachable (using an ICMP Destination Unreachable message). Route discovery fails if an RREQ cannot be generated because the control message generation limit has been reached, or if an RREP is not received within RREQ\_WAIT\_TIME (see Section 7.6).
- o A route discovery is not permitted: any buffered packets requiring that route should be discarded. A route discovery will not be attempted if the source address of the packet needing a route is not configured as a Router Client.
- o A route discovery succeeded: install a corresponding route into the Routing Information Base and begin transmitting any buffered packets.
- o A route has been made invalid: remove the corresponding route from the Routing Information Base.
- o A route has been updated: update the corresponding route in the Routing Information Base.



### 7.5. Message Transmission

AODVv2 sends [RFC5444] formatted messages using the parameters for port number and IP protocol specified in [RFC5498]. Mapping of AODVv2 data to [RFC5444] messages is detailed in Section 9. AODVv2 multicast messages are sent to the link-local multicast address LL-MANET-Routers [RFC5498]. All AODVv2 routers MUST subscribe to LL-MANET-Routers on all AODVv2 interfaces [RFC5498] to receive AODVv2 messages. Note that multicast messages MAY be sent via unicast. For example, this may occur for certain link-types (non-broadcast media), for manually configured router adjacencies, or in order to improve robustness.

When multiple interfaces are available, an AODVv2 router transmitting a multicast message to LL-MANET-Routers MUST send the message on all interfaces that have been configured for AODVv2 operation, as given in the InterfaceSet (Section 5.1).

To avoid congestion, each AODVv2 router's rate of message generation SHOULD be limited (CONTROL\_TRAFFIC\_LIMIT) and administratively configurable. Messages SHOULD NOT be sent more frequently than one message per  $(1 / \text{CONTROL\_TRAFFIC\_LIMIT})$ th of a second. If this threshold is reached, messages MUST be sent based on their priority:

- o Highest priority SHOULD be given to RREP\_Ack messages. This allows links between routers to be confirmed as bidirectional and avoids undesired blacklisting of next hop routers.
- o Second priority SHOULD be given to RERR messages for undeliverable IP packets. This avoids repeated forwarding of packets over broken routes that are still in use by other routers.
- o Third priority SHOULD be given to RREP messages in order that RREQs do not time out.
- o Fourth priority SHOULD be given to RREQ messages.
- o Fifth priority SHOULD be given to RERR messages for newly invalidated routes.
- o Lowest priority SHOULD be given to RERR messages generated in response to RREP messages which cannot be forwarded. In this case the route request will be retried at a later point.

To implement the congestion control, a queue length is set. If the queue is full, in order to queue a new message, a message of lower priority must be removed from the queue. If this is not possible,

the new message MUST be discarded. The queue should be sorted in order of message priority

#### 7.6. Route Discovery, Retries and Buffering

AODVv2's RREQ and RREP messages are used for route discovery. RREQ messages are multicast to solicit an RREP, whereas RREP are unicast. The constants used in this section are defined in Section 11.

When an AODVv2 router needs to forward an IP packet (with source address OrigAddr and destination address TargAddr) from one of its Router Clients, it needs a route to TargAddr in its Routing Information Base. If no route exists, the AODVv2 router generates (RREQ\_Gen) and multicasts a Route Request message (RREQ), on all configured interfaces, containing information about the source and destination. The procedure for this is described in Section 8.1.1. Each generated RREQ results in an increment to the router's sequence number. The AODVv2 router generating an RREQ is referred to as RREQ\_Gen.

Buffering might be configured for IP packets awaiting a route for forwarding by RREQ\_Gen, if sufficient memory is available. Buffering of IP packets might have both positive and negative effects. Real-time traffic, voice, and scheduled delivery may suffer if packets are buffered and subjected to delays, but TCP connection establishment will benefit if packets are queued while route discovery is performed [Koodli01]. Recommendations for appropriate buffer methods are out of scope for this specification. Determining which packets to discard first when the buffer is full is a matter of policy at each AODVv2 router. Note that using different or no buffer methods does not affect interoperability.

RREQ\_Gen awaits reception of a Route Reply message (RREP) containing a route toward TargAddr. This can be achieved by monitoring the entry in the Multicast Route Message Table that corresponds to the generated RREQ. When CurrentTime exceeds RteMsg.Timestamp + RREQ\_WAIT\_TIME and no RREP has been received, RREQ\_Gen will retry the route discovery.

To reduce congestion in a network, repeated attempts at route discovery for a particular target address utilize a binary exponential backoff: for each additional attempt, the time to wait for receipt of the RREP is multiplied by 2. If the requested route is not learned within the wait period, another RREQ is sent, up to a total of DISCOVERY\_ATTEMPTS\_MAX. This is the same technique used in AODV [RFC3561].

Through the use of bidirectional link monitoring and blacklists (see Section 7.2) uni-directional links on initial selected route will be ignored on subsequent route discovery attempts.

Route discovery is considered to have failed after `DISCOVERY_ATTEMPTS_MAX` and the corresponding wait time for an RREP response to the final RREQ. After the attempted route discovery has failed, RREQ\_Gen waits at least `RREQ_HOLDDOWN_TIME` before attempting another route discovery to the same destination, in order to avoid repeatedly generating control traffic that is unlikely to discover a route. Any IP packets buffered for TargAddr are also dropped and a Destination Unreachable ICMP message (Type 3) with a code of 1 (Host Unreachable Error) is delivered to the source of the packet, so that the application knows about the failure.

If RREQ\_Gen does receive a route message containing a route to TargAddr within the timeout, it processes the message according to Section 8. When a valid LocalRoute entry is created in the Local Route Set, the route is also installed in the Routing Information Base, and the router will begin sending the buffered IP packets. Any retry timers for the corresponding RREQ are then cancelled.

During route discovery, all routers on the path learn a route to both OrigPrefix and TargPrefix, so that routes are constructed in both directions. The route is optimized for the forward route.

#### 7.7. Processing Received Route Information

All AODVv2 route messages contain a route. A Route Request (RREQ) contains a route to OrigPrefix, and a Route Reply (RREP) contains a route to TargPrefix. All AODVv2 routers that receive a route message are able to store the route contained within it in their Local Route Set. Incoming information is first checked to verify that it is both safe to use and offers an improvement to existing information, as explained in Section 7.7.1. If these checks pass, the Local Route Set MUST be updated according to Section 7.7.2.

In the processes below, RteMsg is used to denote the route message, AdvRte is used to denote the route contained within it, and LocalRoute denotes an existing entry in the Local Route Set which matches AdvRte on address, prefix length, and metric type.

AdvRte has the following properties:

- o AdvRte.Address := RteMsg.OrigPrefix (in RREQ) or RteMsg.TargPrefix (in RREP)

- o `AdvRte.PrefixLength` := `RteMsg.OrigPrefixLen` (in RREQ) or `RteMsg.TargPrefixLen` (in RREP). If no prefix length was included in `RteMsg`, prefix length is the address length, in bits, of `RteMsg.OrigPrefix` (in RREQ) or `RteMsg.TargPrefix` (in RREP)
- o `AdvRte.SeqNum` := `RteMsg.OrigSeqNum` (in RREQ) or `RteMsg.TargSeqNum` (in RREP)
- o `AdvRte.NextHop` := `RteMsg.IPSourceAddress` (an address of the sending interface of the router from which the `RteMsg` was received)
- o `AdvRte.MetricType` := `RteMsg.MetricType`
- o `AdvRte.Metric` := `RteMsg.Metric`
- o `AdvRte.Cost` := `Cost(R)` using the cost function associated with the route's metric type, i.e.  $\text{Cost}(R) = \text{AdvRte.Metric} + \text{Cost}(L)$ , as described in Section 6, where `L` is the link from the advertising router

#### 7.7.1. Evaluating Route Information

An incoming advertised route (`AdvRte`) is compared to existing `LocalRoutes` to determine whether the advertised route is to be used to update the AODVv2 Local Route Set. The incoming route information MUST be processed as follows:

1. Search for `LocalRoutes` in the Local Route Set matching `AdvRte`'s address, prefix length and metric type
  - \* If no matching `LocalRoute` exists, `AdvRte` MUST be used to update the Local Route Set and no further checks are required.
  - \* If matching `LocalRoutes` are found, continue to Step 2.
2. Compare sequence numbers using the technique described in Section 5.4
  - \* If `AdvRte` is more recent than all matching `LocalRoutes`, `AdvRte` MUST be used to update the Local Route Set and no further checks are required.
  - \* If `AdvRte` is stale, `AdvRte` MUST NOT be used to update the Local Route Set. Ignore `AdvRte` for further processing.
  - \* If the sequence numbers are equal, continue to Step 3.

3. Check that AdvRte is safe against routing loops compared to all matching LocalRoutes (see Section 6)
  - \* If LoopFree(AdvRte, LocalRoute) returns FALSE, ignore AdvRte for further processing. AdvRte MUST NOT be used to update the Local Route Set because using the incoming information might cause a routing loop.
  - \* If LoopFree(AdvRte, LocalRoute) returns TRUE, continue to Step 4.
4. Compare route costs
  - \* If AdvRte is better than all matching LocalRoutes, it MUST be used to update the Local Route Set because it offers improvement.
  - \* If AdvRte is equal in cost and LocalRoute is valid, AdvRte SHOULD NOT be used to update the Local Route Set because it will offer no improvement.
  - \* If AdvRte is worse and LocalRoute is valid, ignore AdvRte for further processing. AdvRte MUST NOT be used to update the Local Route Set because it does not offer any improvement.
  - \* If AdvRte is not better (i.e., it is worse or equal) but LocalRoute is Invalid, AdvRte SHOULD be used to update the Local Route Set because it can safely repair the existing Invalid LocalRoute.

If the advertised route is to be used to update the Local Route Set, the procedure in Section 7.7.2 MUST be followed. If not, non-optimal routes will remain in the Local Route Set.

For information on how to apply these changes to the Routing Information Base, see Section 5.5.

#### 7.7.2. Applying Route Updates

After determining that AdvRte is to be used to update the Local Route Set (as described in Section 7.7.1), the following procedure applies.

If AdvRte is learned from an RREQ message, the link to the next hop neighbor may not be confirmed as bidirectional (see Section 5.3). If there is no existing matching route in the Local Route Set, AdvRte MUST be installed to allow a corresponding RREP to be sent. If a matching entry already exists, AdvRte offers potential improvement, if the link to the neighbor can be confirmed as bidirectional.

The route update is applied as follows:

1. If no existing entry in the Local Route Set matches AdvRte's address, prefix length and metric type, continue to Step 4 and create a new entry in the Local Route Set.
2. If two matching LocalRoutes exist in the Local Route Set, one is a valid route, and one is an Unconfirmed route, AdvRte may offer further improvement to the Unconfirmed route, or may offer an update to the valid route.
  - \* If AdvRte.NextHop's Neighbor.State is Heard, the advertised route may offer improvement to the existing valid route, if the link to the next hop can be confirmed as bidirectional. Continue processing from Step 5 to update the existing Unconfirmed LocalRoute.
  - \* If AdvRte.NextHop's Neighbor.State is Confirmed, the advertised route offers an update or improvement to the existing valid route. Continue processing from Step 5 to update the existing valid LocalRoute.
3. If only one matching LocalRoute exists in the Local Route Set:
  - \* If AdvRte.NextHop's Neighbor.State is Confirmed, continue processing from Step 5 to update the existing LocalRoute.
  - \* If AdvRte.NextHop's Neighbor.State is Heard, AdvRte may offer improvement the existing LocalRoute, if the link to AdvRte.NextHop can be confirmed as bidirectional.
  - \* If LocalRoute.State is Unconfirmed, AdvRte is an improvement to an existing Unconfirmed route. Continue processing from Step 5 to update the existing LocalRoute.
  - \* If LocalRoute.State is Invalid, AdvRte can replace the existing LocalRoute. Continue processing from Step 5 to update the existing LocalRoute.
  - \* If LocalRoute.State is Active or Idle, AdvRte SHOULD be stored as an additional entry in the Local Route Set, with LocalRoute.State set to Unconfirmed. Continue processing from Step 4 to create a new LocalRoute.
4. Create an entry in the Local Route Set and initialize as follows:
  - \* LocalRoute.Address := AdvRte.Address

- \* LocalRoute.PrefixLength := AdvRte.PrefixLength
  - \* LocalRoute.MetricType := AdvRte.MetricType
5. Update the LocalRoute as follows:
- \* LocalRoute.SeqNum := AdvRte.SeqNum
  - \* LocalRoute.NextHop := AdvRte.NextHop
  - \* LocalRoute.NextHopInterface := interface on which RteMsg was received
  - \* LocalRoute.Metric := AdvRte.Cost
  - \* LocalRoute.LastUsed := CurrentTime
  - \* LocalRoute.LastSeqNumUpdate := CurrentTime
6. If a new LocalRoute was created, or if the existing LocalRoute.State is Invalid or Unconfirmed, update LocalRoute as follows:
- \* LocalRoute.State := Unconfirmed (if the next hop's Neighbor.State is Heard)
  - \* LocalRoute.State := Idle (if the next hop's Neighbor.State is Confirmed)
7. If an existing LocalRoute.State changed from Invalid or Unconfirmed to become Idle, any matching Unconfirmed LocalRoute with worse metric value SHOULD be expunged.
8. If an existing LocalRoute was updated with a better metric value, any matching Unconfirmed LocalRoute with worse metric value SHOULD be expunged.
9. If this update results in LocalRoute.State of Active or Idle, which matches a route request which is still in progress, the associated route request retry timers MUST be cancelled.

If this update to the Local Route Set results in two LocalRoutes to the same address, the best LocalRoute will be Unconfirmed. In order to improve the route used for forwarding, the router SHOULD try to determine if the link to the next hop of that LocalRoute is bidirectional, by using that LocalRoute to forward future RREPs and request acknowledgements (see Section 8.2.1 and Section 8.3).

### 7.8. Suppressing Redundant Messages Using the Multicast Route Message Set

When route messages are flooded in a MANET, an AODVv2 router may receive several instances of the same message. Forwarding every one of these gives little additional benefit, and generates unnecessary signaling traffic and might generate unnecessary interference.

Each AODVv2 router stores information about recently received route messages in the AODVv2 Multicast Route Message Set (Section 5.6).

Entries in the Multicast Route Message Set SHOULD be maintained for at least `RteMsg_ENTRY_TIME` after the last Timestamp update in order to account for long-lived RREQs traversing the network. An entry MUST be deleted when the sequence number is no longer valid, i.e., after `MAX_SEQNUM_LIFETIME`. Memory-constrained devices MAY remove the entry before this time.

Received route messages are tested against previously received route messages, and if determined to be redundant, forwarding or response can be avoided.

To determine if a received message is redundant:

1. Search for an entry in the Multicast Route Message Set with the same `OrigPrefix`, `OrigPrefixLen`, `TargPrefix`, `Interface` and `MetricType`
  - \* If there is no entry, the message is not redundant.
  - \* If there is an entry, continue to Step 2.
2. Compare sequence numbers using the technique described in Section 5.4
  - \* Use `OrigSeqNum` of the entry for comparison.
  - \* If the entry has an older sequence number than the received message, the message is not redundant.
  - \* If the entry has a newer sequence number than the received message, the message is redundant.
  - \* If the entry has the same sequence number, continue to Step 3.
3. Compare the metric values



- \* If the entry has a Metric value that is worse than or equal to the metric in the received message, the message is redundant.
- \* If the entry has a Metric value that is better than the metric in the received message, the message is not redundant.

If the message is redundant, update the entry as follows:

- o RteMsg.Timestamp := CurrentTime
- o RteMsg.RemoveTime := CurrentTime + MAX\_SEQNUM\_LIFETIME

since matching route messages are still traversing the network and this entry should be maintained. This message MUST NOT be forwarded or responded to.

If the message is not redundant, create an entry or update the existing entry.

To update a Multicast Route Message Set entry, set:

- o RteMsg.OrigPrefix := OrigPrefix from the message
- o RteMsg.OrigPrefixLen := the prefix length associated with OrigPrefix
- o RteMsg.TargPrefix := TargPrefix from the message
- o RteMsg.OrigSeqNum := the sequence number associated with OrigPrefix, if RteMsg is an RREQ
- o RteMsg.TargSeqNum := the sequence number associated with TargPrefix, if RteMsg is an RREP
- o RteMsg.Metric := the metric value associated with OrigPrefix in a received RREQ
- o RteMsg.MetricType := the metric type associated with RteMsg.Metric
- o RteMsg.Timestamp := CurrentTime
- o RteMsg.RemoveTime := CurrentTime + MAX\_SEQNUM\_LIFETIME

Where the message is determined not redundant before Step 3, it MUST be forwarded or responded to. When a message is determined to be not redundant in Step 3, it MAY be suppressed to avoid extra control traffic. However, since the processing of the message will result in an update to the Local Route Set, the message SHOULD be forwarded or

responded to, to ensure other routers have up-to-date information and the best metrics. If the message is not forwarded, the best route may not be found. Forwarding or response is to be performed using the processes outlined in Section 8.

#### 7.9. Suppressing Redundant Route Error Messages using the Route Error Set

In order to avoid flooding the network with RERR messages when a stream of IP packets to an unreachable address arrives, an AODVv2 router SHOULD avoid creating duplicate messages by determining whether an equivalent RERR has recently been sent. This is achieved with the help of the Route Error Set (see Section 5.7).

To determine if a RERR should be created:

1. Search for an entry in the Route Error Set where:

- \* RerrMsg.UnreachableAddress == UnreachableAddress to be reported
- \* RerrMsg.PktSource == PktSource to be included in the RERR

If a matching entry is found, no further processing is required and the RERR SHOULD NOT be sent.

2. If no matching entry is found, a new entry with the following properties is created, and the RERR is created and sent as described in Section 8.4.1:

- \* RerrMsg.Timeout := CurrentTime + RERR\_TIMEOUT
- \* RerrMsg.UnreachableAddress == UnreachableAddress to be reported
- \* RerrMsg.PktSource == PktSource to be included in the RERR

#### 7.10. Local Route Set Maintenance

Route maintenance involves monitoring LocalRoutes in the Local Route Set, updating LocalRoute.State to handle route timeouts and reporting routes that become Invalid.

##### 7.10.1. LocalRoute State Changes

During normal operation, AODVv2 does not require any explicit timeouts to manage the lifetime of a route. At any time, any LocalRoute MAY be examined and updated according to the rules below.

If timers are not used to prompt updates of LocalRoute.State, the LocalRoute.State MUST be checked before IP packet forwarding and before any operation based on LocalRoute.State.

Route timeout behaviour is as follows:

- o An Unconfirmed route MUST be expunged at MAX\_SEQNUM\_LIFETIME after LocalRoute.LastSeqNumUpdate.
  - o An Idle route MUST become Active when used to forward an IP packet. If the route is not used to forward an IP packet within MAX\_IDLETIME, LocalRoute.State MUST become Invalid.
  - o An Invalid route SHOULD remain in the Local Route Set, since LocalRoute.SeqNum is used to classify future information about LocalRoute.Address as stale or fresh.
  - o In all cases, if the time since LocalRoute.LastSeqNumUpdate exceeds MAX\_SEQNUM\_LIFETIME, LocalRoute.SeqNum must be set to
1. This is required to ensure that any AODVv2 routers following the initialization procedure can safely begin routing functions using a new sequence number. A LocalRoute with LocalRoute.State set to Active or Idle can remain in the Local Route Set after the sequence number has been set to 0, for example if the route is reliably carrying traffic. If LocalRoute.State is Invalid, or later becomes Invalid, the LocalRoute MUST be expunged from the Local Route Set.

LocalRoutes can become Invalid before a timeout occurs:

- o If an external mechanism reports a link as broken, all LocalRoutes using that link for LocalRoute.NextHop MUST immediately have LocalRoute.State set to Invalid.
- o LocalRoute.State MUST immediately be set to Invalid if a Route Error (RERR) message is received where:
  - \* The sender is LocalRoute.NextHop or PktSource is a Router Client address
  - \* There is an Address in AddressList which matches LocalRoute.Address, and:
    - + The prefix length associated with this Address, if any, matches LocalRoute.PrefixLength

- + The sequence number associated with this Address, if any, is newer or equal to LocalRoute.SeqNum (see Section 5.4)
- + The metric type associated with this Address matches LocalRoute.MetricType

LocalRoutes are also updated when Neighbor.State is updated:

- o While the value of Neighbor.State is set to Heard, any routes in the Local Route Set using that neighbor as a next hop MUST have LocalRoute.State set to Unconfirmed.
- o When the value of Neighbor.State is set to Confirmed, the Unconfirmed routes in the Local Route Set using that neighbor as a next hop MUST have LocalRoute.State set to Idle. Any other matching LocalRoutes with metric values worse than LocalRoute.Metric MUST be expunged from the Local Route Set.
- o When the value of Neighbor.State is set to Blacklisted, any valid routes in the Local Route Set using that neighbor for their next hop MUST have LocalRoute.State set to Invalid.
- o When a Neighbor Set entry is removed, all routes in the Local Route Set using that neighbor as next hop MUST have LocalRoute.State set to Invalid.

Memory constrained devices MAY choose to expunge routes from the AODVv2 Local Route Set at other times, but MUST adhere to the following rules:

- o An Active route MUST NOT be expunged, as it is in use. If deleted, IP traffic forwarded to this router will prompt generation of a Route Error message, and it will be necessary for a Route Request to be generated by the originator's router to re-establish the route.
- o An Idle route SHOULD NOT be expunged, as it is still valid for forwarding IP traffic. If deleted, this could result in dropped IP packets and a Route Request could be generated to re-establish the route.
- o Any Invalid route MAY be expunged. Least recently used Invalid routes SHOULD be expunged first, since the sequence number information is less likely to be useful.
- o An Unconfirmed route MUST NOT be expunged if it was installed within the last RREQ\_WAIT\_TIME, because it may correspond to a route discovery in progress. A Route Reply message might be

received which needs to use the LocalRoute.NextHop information. Otherwise, it MAY be expunged.

#### 7.10.2. Reporting Invalid Routes

When LocalRoute.State changes from Active to Invalid as a result of a broken link or a received Route Error (RERR) message, other AODVv2 routers MUST be informed by sending an RERR message containing details of the invalidated route.

An RERR message MUST also be sent when an AODVv2 router receives an RREP message to forward, but the LocalRoute to the OrigPrefix in the RREP has been lost or is marked as Invalid.

An RERR message MUST also be sent when an AODVv2 router receives an RREP message to forward, but the LocalRoute to the OrigAddr in the RREP has been lost or is marked as Invalid.

The packet or message triggering the RERR MUST be discarded.

Generation of an RERR message is described in Section 8.4.1.

### 8. AODVv2 Protocol Messages

AODVv2 defines four message types: Route Request (RREQ), Route Reply (RREP), Route Reply Acknowledgement (RREP\_Ack), and Route Error (RERR).

Each AODVv2 message is defined as a set of data. Rules for the generation, reception and forwarding of each message type are described in the following sections. Section 9 discusses how the data is mapped to [RFC5444] Message TLVs, Address Blocks, and Address TLVs.

#### 8.1. Route Request (RREQ) Message

Route Request messages are used in route discovery operations to request a route to a specified target address. RREQ messages have the following contents:

msg_hop_limit
AddressList
PrefixLengthList (optional)
OrigSeqNum, (optional) TargSeqNum
MetricType
OrigMetric

Figure 1: RREQ message contents

**msg\_hop\_limit**

The remaining number of hops allowed for dissemination of the RREQ message.

**AddressList**

Contains OrigPrefix, from the Router Client entry which includes OrigAddr, the source address of the IP packet for which a route is requested, and TargPrefix, set to TargAddr, the destination address of the IP packet for which a route is requested.

**PrefixLengthList**

Contains OrigPrefixLen, i.e., the length, in bits, of the prefix associated with the Router Client entry which includes OrigAddr. If omitted, the prefix length is equal to OrigAddr's address length in bits.

**OrigSeqNum**

The sequence number associated with OrigPrefix.

**TargSeqNum**

A sequence number associated with an existing Invalid route to TargAddr. This MAY be included if available.

**MetricType**

The metric type associated with OrigMetric.

**OrigMetric**

The metric value associated with the route to OrigPrefix, as seen from the sender of the message.

#### 8.1.1.1. RREQ Generation

An RREQ is generated when an IP packet needs to be forwarded for a Router Client, and no valid route currently exists for the packet's destination in the Routing Information Base.

Before creating an RREQ, the router SHOULD check the Multicast Route Message Set to see if an RREQ has recently been sent for the requested destination. If so, and the wait time for a reply has not yet been reached, the router SHOULD continue to await a response without generating a new RREQ. If the timeout has been reached, a new RREQ MAY be generated. If buffering is configured, incoming IP packets awaiting this route SHOULD be buffered until the route discovery is completed.

If the limit for the rate of AODVv2 control message generation has been reached, no message SHOULD be generated.

To generate the RREQ, the router (referred to as RREQ\_Gen) follows this procedure:

1. Set msg\_hop\_limit := MAX\_HOPCOUNT
2. Set AddressList := {OrigPrefix, TargPrefix}
3. For the PrefixLengthList:
  - \* If OrigAddr is part of an address range configured as a Router Client, set PrefixLengthList := {RouterClient.PrefixLength, null}.
  - \* Otherwise, omit PrefixLengthList.
4. For OrigSeqNum:
  - \* Increment the router Sequence Number as specified in Section 5.4.
  - \* Set OrigSeqNum := router Sequence Number.
5. For TargSeqNum:
  - \* If an Invalid route exists in the Local Route Set matching TargAddr using longest prefix matching and has a valid sequence number, set TargSeqNum := LocalRoute.SeqNum.

- \* If no Invalid route exists in the Local Route Set matching TargAddr, or the route doesn't have a sequence number, omit TargSeqNum.
6. Include MetricType and set the type accordingly
  7. Find the Router Client Set Entry where RouterClient.IPAddress == OrigPrefix:
    - \* Set OrigMetric := RouterClient.Cost

This AODVv2 message is used to create a corresponding [RFC5444] message (see Section 9) which is handed to the RFC5444 multiplexer for further processing. By default, the multiplexer is instructed to multicast the message to LL-MANET- Routers on all interfaces configured for AODVv2 operation. The RREP MUST be sent over LocalRoute[OrigPrefix].NextHopInterface.

#### 8.1.2. RREQ Reception

Upon receiving a Route Request, an AODVv2 router performs the following steps:

1. Check and update the Neighbor Set according to Section 7.3
  - \* If the sender has Neighbor.State set to Blacklisted, ignore this RREQ for further processing.
2. Verify that the message contains the required data: msg\_hop\_limit, OrigPrefix, TargPrefix, OrigSeqNum, and OrigMetric, and that OrigPrefix and TargPrefix are valid addresses
  - \* If not, ignore this RREQ for further processing.
3. Check that the MetricType is supported and configured for use
  - \* If not, ignore this RREQ for further processing.
4. Verify that the cost of the advertised route will not exceed the maximum allowed metric value for the metric type (Metric <= MAX\_METRIC[MetricType] - Cost(L))
  - \* If it will, ignore this RREQ for further processing.
5. Process the route to OrigPrefix as specified in Section 7.7



6. Check if the information in the message is redundant by comparing to entries in the Multicast Route Message Set, following the procedure in Section 7.8
  - \* If redundant, ignore this RREQ for further processing.
  - \* If not redundant, create a new entry in the Multicast Route Message Set and continue processing.
7. Check if the TargPrefix matches an entry in the Router Client Set
  - \* If so, generate an RREP as specified in Section 8.2.1.
  - \* If not, continue to RREQ forwarding.

#### 8.1.3. RREQ Forwarding

By forwarding an RREQ, a router advertises that it will forward IP packets to the OrigPrefix contained in the RREQ according to the information enclosed. The router MAY choose not to forward the RREQ, for example if the router is heavily loaded or low on energy and therefore unwilling to advertise routing capability for more traffic. This could, however, decrease connectivity in the network or result in non-optimal paths.

The RREQ SHOULD NOT be forwarded if the limit for the rate of AODVv2 control message generation has been reached.

The procedure for RREQ forwarding is as follows:

1. Set `msg_hop_limit` := `received msg_hop_limit` - 1
2. If `msg_hop_limit` is now zero, do not continue the forwarding process
3. Set `OrigMetric` := `LocalRoute[OrigPrefix].Metric`

This modified message is handed to the [RFC5444] multiplexer for further processing. By default, the multiplexer is instructed to multicast the message to LL-MANET-Routers on all interfaces configured for AODVv2 operation.

#### 8.2. Route Reply (RREP) Message

When a Route Request message is received, requesting a route to a target address (TargAddr) which is configured as part of a Router Client entry, a Route Reply message is sent in response. The RREP offers a route to TargPrefix.

RREP messages have the following contents:

msg_hop_limit
AddressList
PrefixLengthList (optional)
TargSeqNum
MetricType
TargMetric

Figure 2: RREP message contents

#### msg\_hop\_limit

The remaining number of hops allowed for dissemination of the RREP message.

#### AddressList

Contains OrigPrefix and TargPrefix, the prefixes of the source and destination addresses of the IP packet for which a route is requested.

#### PrefixLengthList

Contains TargPrefixLen, i.e., the length, in bits, of the prefix associated with the Router Client entry which includes TargAddr. If omitted, the prefix length is equal to TargAddr's address length, in bits.

#### TargSeqNum

The sequence number associated with TargPrefix.

#### MetricType

The metric type associated with TargMetric.

#### TargMetric

The metric value associated with the route to TargPrefix, as seen from the sender of the message.

### 8.2.1. RREP Generation

A Route Reply message is generated when a Route Request for a Router Client of the AODVv2 router arrives. This is the case when

RteMsg.TargPrefix matches an entry in the Router Client Set of the AODVv2 router.

Before creating an RREP, the router SHOULD check if CONTROL\_TRAFFIC\_LIMIT has been reached. If so, the RREP SHOULD NOT be created.

The RREP will follow the path of the route to OrigPrefix. If the best route to OrigPrefix in the Local Route Set is Unconfirmed, the link to the next hop neighbor is not yet confirmed as bidirectional (as described in Section 7.2). In this case an RREP\_Ack MUST also be sent as described in Section 8.3, in order to request an acknowledgement message from the next hop router to prove that the link is bidirectional. If the best route to OrigPrefix in the Local Route Set is valid, the link to the next hop neighbor is already confirmed as bidirectional, and no acknowledgement is required.

Implementations MAY allow a number of retries of the RREP if a requested acknowledgement is not received within RREP\_Ack\_SENT\_TIMEOUT, doubling the timeout with each retry, up to a maximum of RREP\_RETRIES, using the same exponential backoff described in Section 7.6 for RREQ retries. The acknowledgement MUST be considered to have failed after the wait time for an RREP\_Ack response to the final RREP.

To generate the RREP, the router (also referred to as RREP\_Gen) follows this procedure:

1. Set msg\_hop\_limit := MAX\_HOPCOUNT - msg\_hop\_limit from the received RREQ message
2. Set Address List := {OrigPrefix, TargPrefix}
3. For the PrefixLengthList:
  - \* If TargAddr is part of an address range configured as a Router Client, set PrefixLengthList := {null, RouterClient.PrefixLength}.
  - \* Otherwise, omit PrefixLengthList.
4. For the TargSeqNum:
  - \* Increment the router Sequence Number as specified in Section 5.4.
  - \* Set TargSeqNum := router Sequence Number.

5. Include MetricType and set the type to match the MetricType in the received RREQ message
6. Set TargMetric := RouterClient.Cost for the Router Client entry which includes TargAddr

This AODVv2 message is used to create a corresponding [RFC5444] message (see Section 9) which is handed to the RFC5444 multiplexer for further processing. The multiplexer is instructed to unicast the RREP to LocalRoute[OrigPrefix].NextHop. The RREP MUST be sent over LocalRoute[OrigPrefix].NextHopInterface.

#### 8.2.2. RREP Reception

Upon receiving a Route Reply, an AODVv2 router performs the following steps:

1. Verify that the message contains the required data: msg\_hop\_limit, OrigPrefix, TargPrefix, TargSeqNum, and TargMetric, and that OrigPrefix and TargPrefix are valid addresses
  - \* If not, ignore this RREP for further processing.
2. Check that the MetricType is supported and configured for use
  - \* If not, ignore this RREP for further processing. <!--
3. If this RREP does not correspond to an RREQ generated or forwarded in the last RREQ\_WAIT\_TIME, ignore for further processing. -->
4. If the Multicast Route Message Set does not contain an entry where:
  - o RteMsg.OrigPrefix == RREP.OrigPrefix
  - o RteMsg.OrigPrefixLen == RREP.OrigPrefixLen
  - o RteMsg.TargAddr exists within RREP.TargPrefix
  - o RteMsg.OrigSeqNum <= RREP.OrigSeqNum
  - o RteMsg.MetricType == RREP.MetricType
  - o RteMsg.Timestamp > CurrentTime - RREQ\_WAIT\_TIME
  - o RteMsg.Interface == The interface on which the RREP was received

ignore this RREP for further processing, since it does not correspond to a previously sent RREQ.

1. Update the Neighbor Set according to Section 7.3
2. Verify that the cost of the advertised route does not exceed the maximum allowed metric value for the metric type ( $\text{Metric} \leq \text{MAX\_METRIC}[\text{MetricType}] - \text{Cost}(L)$ )
  - \* If it does, ignore this RREP for further processing.
3. Process the route to TargPrefix as specified in Section 7.7
4. Check if the message is redundant by comparing to entries in the Multicast Route Message Set (Section 7.8)
  - \* If redundant, ignore this RREP for further processing.
  - \* If not redundant, save the information in the Multicast Route Message Set to identify future redundant RREP messages and continue processing.
5. Check if the OrigPrefix matches an entry in the Router Client Set
  - \* If so, no further processing is necessary.
  - \* If not, continue to Step 10.
6. Check if a valid (Active or Idle) or Unconfirmed LocalRoute exists to OrigPrefix
  - \* If so, continue to RREP forwarding.
  - \* If not, a Route Error message SHOULD be transmitted toward TargPrefix according to Section 8.4.1 and the RREP SHOULD be discarded and not forwarded.

#### 8.2.3. RREP Forwarding

A received Route Reply message is forwarded toward OrigPrefix. By forwarding an RREP, a router advertises that it will forward IP packets to TargPrefix.

The RREP SHOULD NOT be forwarded if CONTROL\_TRAFFIC\_LIMIT has been reached. Otherwise, the router MUST forward the RREP.

The procedure for RREP forwarding is as follows:

1. Set `msg_hop_limit := received msg_hop_limit - 1`
2. If `msg_hop_limit` is now zero, do not continue the forwarding process
3. Set `TargMetric := LocalRoute[TargPrefix].Metric`

This modified message is handed to the [RFC5444] multiplexer for further processing. The multiplexer is instructed to unicast the RREP to LocalRoute[OrigPrefix].NextHop. The RREP MUST be sent over LocalRoute[OrigPrefix].NextHopInterface.

### 8.3. Route Reply Acknowledgement (RREP\_Ack) Message

The Route Reply Acknowledgement is used as both a request and a response message to test bidirectionality of a link over which a Route Reply has also been sent. The router which forwards the RREP MUST send a Route Reply Acknowledgement message to the intended next hop, if the link to the next hop neighbor is not yet confirmed as bidirectional.

The receiving router MUST then reply with a Route Reply Acknowledgement response message.

When the Route Reply Acknowledgement response message is received by the sender of the RREP, it confirms that the link between the two routers is bidirectional (see Section 7.2).

If the Route Reply Acknowledgement is not received within `RREP_Ack_SENT_TIMEOUT`, the link is determined to be unidirectional.

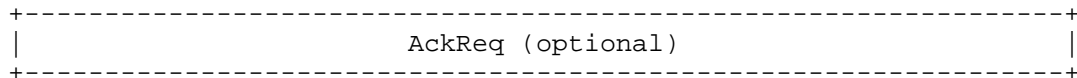


Figure 3: RREP\_Ack message contents

### 8.3.1. RREP\_Ack Request Generation

An RREP\_Ack MUST be generated if a Route Reply is sent over a link which is not known to be bidirectional. It includes an AckReq element to indicate that it is a request for acknowledgement.

The RREP\_Ack SHOULD NOT be generated if the limit for the rate of AODVv2 control message generation has been reached.

The [RFC5444] representation of the RREP\_Ack is discussed in Section 9.

The RREP\_Ack request MUST be sent unicast to the LocalRoute[OrigPrefix].NextHop via LocalRoute[OrigPrefix].NextHopInterface. The multiplexer MAY be instructed to send the RREP\_Ack in the same [RFC5444] packet as the RREP.

The Neighbor Set entry for LocalRoute[OrigPrefix].NextHop MUST also be updated to indicate that an RREP\_Ack is required (see Section 7.3).

#### 8.3.2. RREP\_Ack Reception

Upon receiving an RREP\_Ack, an AODVv2 router performs the following steps:

1. Check if an AckReq element is included:
  - \* If so, create an RREP\_Ack Response as described in Section 8.3.3. No further processing is required.
  - \* If not, continue to step 2.
2. Check if the RREP\_Ack was expected:
  - \* Check if the Neighbor Set contains an entry where:
    - + Neighbor.IPAddress == IP.SourceAddress of the RREP\_Ack message
    - + Neighbor.State == Heard
    - + Neighbor.Timeout < CurrentTime
    - + Neighbor.Interface matches the interface on which the RREP\_Ack was received
  - \* If it does, the router sets Neighbor.Timeout to INFINITY\_TIME, and processing continues to Step 3.
  - \* Otherwise no actions are required and processing ends.
3. Update the Neighbor Set according to Section 7.3, including updating routes using this Neighbor as LocalRoute.NextHop.

### 8.3.3. RREP\_Ack Response Generation

An RREP\_Ack response MUST be generated if a received RREP\_Ack includes an AckReq.

The RREP\_Ack response SHOULD NOT be generated if the limit for the rate of AODVv2 control message generation has been reached.

There is no further data in an RREP\_Ack response. The [RFC5444] representation is discussed in Section 9. In this case, the multiplexer is instructed to unicast the RREP\_Ack to the source IP address of the RREP\_Ack message that requested it, over the same interface on which the RREP\_Ack was received.

#### 8.4. Route Error (RERR) Message

A Route Error message is generated by an AODVv2 router to notify other AODVv2 routers of routes that are no longer available. An RERR message has the following contents:

PktSource (optional)
AddressList
PrefixLengthList (optional)
SeqNumList (optional)
MetricTypeList

Figure 4: RERR message contents

## PktSource

The source address of the IP packet triggering the RERR. If the RERR is triggered by a broken link, PktSource is not required.

## AddressList

The addresses of the routes not available through RERR\_Gen.

## PrefixLengthList

The prefix lengths, in bits, associated with the routes not available through RERR\_Gen. These values indicate whether routes represent a single device or an address range.

## SeqNumList



The sequence numbers of the routes not available through RERR\_Gen (where known).

#### MetricTypeList

The metric types associated with the routes not available through RERR\_Gen.

#### 8.4.1. RERR Generation

A Route Error message is generated when an AODVv2 router (also referred to as RERR\_Gen) needs to report that a destination is not reachable. There are three events that cause this response:

- o When an IP packet that has been forwarded from another router, but cannot be forwarded further because there is no valid route in the Routing Information Base for its destination, the source of the packet needs to be informed that the route to the destination of the packet does not exist. The RERR generated MUST include PktSource set to the source address of the IP packet, and MUST contain only one unreachable address in the AddressList, i.e., the destination address of the IP packet. RERR\_Gen MUST discard the IP packet that triggered generation of the RERR. The prefix length, sequence number and metric type SHOULD be included if known from an existing Invalid LocalRoute to the unreachable address.
- o When an RREP message cannot be forwarded because the LocalRoute to OrigPrefix has been lost or is Invalid, RREP\_Gen needs to be informed that the route to OrigPrefix does not exist. The RERR generated MUST include PktSource set to the TargPrefix of the RREP, and MUST contain only one unreachable address in the AddressList, the OrigPrefix from the RREP. RERR\_Gen MUST discard the RREP message that triggered generation of the RERR. The prefix length, sequence number and metric type SHOULD be included if known from an Invalid LocalRoute to the unreachable address.
- o When a link breaks, multiple LocalRoutes may become Invalid, and the RERR generated MAY contain multiple unreachable addresses. The RERR MUST include MetricTypeList. PktSource is omitted. All previously Active LocalRoutes that used the broken link MUST be reported. The AddressList, PrefixLengthList, SeqNumList, and MetricTypeList will contain entries for each LocalRoute which has become Invalid. An RERR message is only sent if an Active LocalRoute becomes Invalid, though an AODVv2 router can also include Idle LocalRoutes that become Invalid if the configuration parameter ENABLE\_IDLE\_IN\_RERR is set (see Section 11.3).

The RERR SHOULD NOT be generated if CONTROL\_TRAFFIC\_LIMIT has been reached. The RERR also SHOULD NOT be generated if it is a duplicate, as determined by Section 7.9.

Incidentally, if an AODVv2 router receives an ICMP error packet to or from the address of one of its Router Clients, it forwards the ICMP packet in the same way as any other IP packet, and will not generate any RERR message based on the contents of the ICMP packet.

To generate the RERR, the router follows this procedure:

1. If necessary, include PktSource and set the value as given above
2. For each LocalRoute that needs to be reported:
  - \* Insert LocalRoute.Address into the AddressList.
  - \* Insert LocalRoute.PrefixLength into PrefixLengthList, if known and not equal to the address length.
  - \* Insert LocalRoute.SeqNum into SeqNumList, if known.
  - \* Insert LocalRoute.MetricType into MetricTypeList.

The AODVv2 message is used to create a corresponding [RFC5444] message (see Section 9).

If the RERR is sent in response to an undeliverable IP packet or RREP message, i.e., if PktSource is included, the RERR SHOULD be sent unicast to the next hop on the route to PktSource. It MUST be sent over the same interface on which the undeliverable IP packet was received. If there is no route to PktSource, the RERR SHOULD be multicast to LL-MANET-Routers. If the RERR is sent in response to a broken link, i.e., PktSource is not included, the RERR is, by default, multicast to LL-MANET-Routers.

#### 8.4.2. RERR Reception

Upon receiving a Route Error, an AODVv2 router performs the following steps:

1. Verify that the message contains the required data: at least one unreachable address
  - \* If not, ignore this RERR for further processing.
2. For each address in the AddressList, check that:

- \* The address is valid (routable and unicast)
- \* The MetricType is supported and configured for use
- \* There is a LocalRoute with the same MetricType matching the address using longest prefix matching
- \* Either the LocalRoute's next hop is the sender of the RERR and the next hop interface is the interface on which the RERR was received, or PktSource is present in the RERR and is a Router Client address
- \* The unreachable address' sequence number is either unknown, or is greater than the LocalRoute's sequence number

If any of the above are false the address does not match a LocalRoute and MUST NOT be processed or regenerated in a RERR.

If all of the above are true, the LocalRoute which matches the address is no longer valid. If the LocalRoute was previously Active, it MUST be reported in a regenerated RERR. If the LocalRoute was previously Idle, it MAY be reported in a regenerated RERR, if ENABLE\_IDLE\_IN\_RERR is configured. The Local Route Set MUST be updated according to these rules:

- \* If the LocalRoute's prefix length is the same as the unreachable address' prefix length, set LocalRoute.State to Invalid.
  - \* If the LocalRoute's prefix length is longer than the unreachable address' prefix length, the LocalRoute MUST be expunged from the Local Route Set, since it is a sub-route of the route which is reported to be Invalid.
  - \* If the prefix length is different, create a new LocalRoute with the unreachable address, and its prefix length and sequence number, and set LocalRoute.State to Invalid. These Invalid routes are retained to avoid processing stale messages.
  - \* Update the sequence number on the existing LocalRoute, if the reported sequence number is determined to be newer using the comparison technique described in Section 5.4.
3. If there are previously Active LocalRoutes that MUST be reported, as identified in step 2.:
    - \* Regenerate the RERR as detailed in Section 8.4.3.

#### 8.4.3. RERR Regeneration

The Route Error message SHOULD NOT be regenerated if CONTROL\_TRAFFIC\_LIMIT has been reached.

The procedure for RERR regeneration is as follows:

1. If PktSource was included in the original RERR, and PktSource is not a Router Client, copy it into the regenerated RERR
2. For each LocalRoute that needs to be reported as identified in Section 8.4.1:
  - \* Insert LocalRoute.Address into the AddressList.
  - \* Insert LocalRoute.PrefixLength into PrefixLengthList, if known and not equal to the address length.
  - \* Insert LocalRoute.SeqNum into SeqNumList, if known.
  - \* Insert LocalRoute.MetricType into MetricTypeList.

The AODVv2 message is used to create a corresponding [RFC5444] message (see Section 9). If the RERR contains PktSource, the regenerated RERR SHOULD be sent unicast to the next hop on the LocalRoute to PktSource. It MUST be sent over the same interface on which the undeliverable IP packet was received. If there is no route to PktSource, or PktSource is a Router Client, it SHOULD be multicast to LL-MANET-Routers. If the RERR is sent in response to a broken link, the RERR is, by default, multicast to LL-MANET-Routers.

#### 9. RFC 5444 Representation

AODVv2 specifies that all control messages between routers MUST use the Generalized Mobile Ad Hoc Network Packet/Message Format [RFC5444], and therefore AODVv2's route messages comprise data which is mapped to message elements in [RFC5444].

[RFC5444] provides a multiplexed transport for multiple protocols. An [RFC5444] implementation MAY choose to optimize the content of certain elements during message creation to reduce control message overhead.

A brief summary of the [RFC5444] format:

1. A packet contains zero or more messages

2. A message contains a Message Header, one Message TLV Block, zero or more Address Blocks, and one Address Block TLV Block per Address Block
3. The Message TLV Block MAY contain zero or more Message TLVs
4. An Address Block TLV Block MAY include zero or more Address Block TLVs
5. Each TLV value in an Address Block TLV Block can be associated with all of the addresses, or with a contiguous set of addresses, or with a single address in the Address Block

AODVv2 does not require access to the [RFC5444] packet header.

In the message header, AODVv2 uses <msg-type>, <msg-hop-limit> and <msg-addr-length>. The <msg-addr-length> field indicates the length of any addresses in the message, using <msg-addr-length> := (address length in octets - 1), i.e. 3 for IPv4 and 15 for IPv6.

The addresses in an Address Block MAY appear in any order, and values in a TLV in the Address Block TLV Block must be associated with the correct address in the Address Block by the [RFC5444] implementation. To indicate which value is associated with each address, the AODVv2 message representation uses lists where the order of the addresses in the AODVv2 AddressList matches the order of values in other data lists, e.g., the order of SeqNums in the SeqNumList in an RERR. [RFC5444] maps this information to Address Block TLVs associated with the relevant addresses in the Address Block.

Each address included in the Address Block is identified as OrigPrefix, TargPrefix, PktSource, or Unreachable Address by including an ADDRESS\_TYPE TLV in the Address Block TLV Block.

The following sections show how AODVv2 data is represented in [RFC5444] messages. AODVv2 defines (in Section 11.8) a number of new TLVs.

Where the extension type of a TLV is set to zero, this is the default [RFC5444] value and the extension type will not be included in the message.

### 9.1. Route Request Message Representation

## 9.1.1. Message Header

Data	Header Field	Value
None	<msg-type>	RREQ
msg_hop_limit	<msg-hop-limit>	MAX_HOPCOUNT, reduced by number of hops traversed so far by the message.

## 9.1.2. Message TLV Block

AODVv2 does not define any Message TLVs for an RREQ message.

## 9.1.3. Address Block

An RREQ contains OrigPrefix and TargPrefix, and each of these addresses has an associated prefix length. If the prefix length has not been included in the AODVv2 message, it is equal to the address length in bits.

Data	Address Block
OrigPrefix/OrigPrefixLen	<address> + <prefix-length>
TargPrefix/TargPrefixLen	<address> + <prefix-length>

## 9.1.4. Address Block TLV Block

Address Block TLVs are always associated with one or more addresses in the Address Block. The following sections show the TLVs that apply to each address.

## 9.1.4.1. Address Block TLVs for OrigPrefix

Data	TLV Type	Extension Type	Value
None OrigSeqNum	ADDRESS_TYPE SEQ_NUM	0 0	ORIGPREFIX Sequence number of RREQ_Gen, the router which initiated route discovery.
OrigMetric /MetricType	PATH_METRIC	MetricType	Metric value for the route to OrigPrefix, using MetricType.

#### 9.1.4.2. Address Block TLVs for TargPrefix

Data	TLV Type	Extension Type	Value
None TargSeqNum	ADDRESS_TYPE SEQ_NUM	0 0	TARGPREFIX The last known TargSeqNum for TargPrefix.

### 9.2. Route Reply Message Representation

#### 9.2.1. Message Header

Data	Header Field	Value
None msg_hop_limit	<msg-type> <msg-hop-limit>	RREP MAX_HOPCOUNT - msg_hop_limit from the corresponding RREQ, reduced by number of hops traversed so far by the message.

#### 9.2.2. Message TLV Block

AODVv2 does not define any Message TLVs for an RREP message.

### 9.2.3. Address Block

An RREP contains OrigPrefix and TargPrefix, and each of these addresses has an associated prefix length. If the prefix length has not been included in the AODVv2 message, it is equal to the address length in bits.

Data	Address Block
OrigPrefix/OrigPrefixLen	<address> + <prefix-length>
TargPrefix/TargPrefixLen	<address> + <prefix-length>

### 9.2.4. Address Block TLV Block

Address Block TLVs are always associated with one or more addresses in the Address Block. The following sections show the TLVs that apply to each address.

#### 9.2.4.1. Address Block TLVs for OrigPrefix

Data	TLV Type	Extension Type	Value
None	ADDRESS_TYPE	0	ORIGPREFIX

#### 9.2.4.2. Address Block TLVs for TargPrefix

Data	TLV Type	Extension Type	Value
None	ADDRESS_TYPE	0	TARGPREFIX
TargSeqNum	SEQ_NUM	0	Sequence number of RREP_Gen, the router which created the RREP.
TargMetric /MetricType	PATH_METRIC	MetricType	Metric value for the route to TargPrefix, using MetricType.



### 9.3. Route Reply Acknowledgement Message Representation

#### 9.3.1. Message Header

+-----+	+-----+	+-----+	+-----+
Data	Header Field	Value	
+-----+	+-----+	+-----+	+-----+
None	<msg-type>	RREP_Ack	
+-----+	+-----+	+-----+	+-----+

#### 9.3.2. Message TLV Block

AODVv2 defines an AckReq Message TLV, included when an acknowledgement of this message is required, in order to monitor adjacency, as described in Section 7.2.

+-----+	+-----+	+-----+	+-----+
Data	TLV Type	Extension Type	Value
+-----+	+-----+	+-----+	+-----+
AckReq	ACK_REQ	0	None
+-----+	+-----+	+-----+	+-----+

#### 9.3.3. Address Block

AODVv2 does not define an Address Block for an RREP\_Ack message.

#### 9.3.4. Address Block TLV Block

AODVv2 does not define any Address Block TLVs for an RREP\_Ack message.

### 9.4. Route Error Message Representation

Route Error Messages MAY be split into multiple [RFC5444] messages when the desired contents would exceed the MTU. However, all of the resulting messages MUST have the same message header as described below. If PktSource is included in the AODVv2 message, it MUST be included in all of the resulting [RFC5444] messages.

#### 9.4.1. Message Header

+-----+	+-----+	+-----+	+-----+
Data	Header Field	Value	
+-----+	+-----+	+-----+	+-----+
None	<msg-type>	RERR	
+-----+	+-----+	+-----+	+-----+

#### 9.4.2. Message TLV Block

AODVv2 does not define any Message TLVs for an RERR message.

#### 9.4.3. Address Block

The Address Block in an RERR MAY contain PktSource, the source address of the IP packet triggering RERR generation, as detailed in Section 8.4. The prefix length associated with PktSource is equal to the address length in bits.

Address Block always contains one address per route that is no longer valid, and each address has an associated prefix length. If a prefix length has not been included for this address, it is equal to the address length in bits.

Data	Address Block
PktSource	<address> + <prefix-length> for PktSource
AddressList/PrefixLengthList	<address> + <prefix-length> for each unreachable address in AddressList

#### 9.4.4. Address Block TLV Block

Address Block TLVs are always associated with one or more addresses in the Address Block. The following sections show the TLVs that apply to each type of address in the RERR.

##### 9.4.4.1. Address Block TLVs for PktSource

Data	TLV Type	Extension Type	Value
PktSource	ADDRESS_TYPE	0	PKTSOURCE

##### 9.4.4.2. Address Block TLVs for Unreachable Addresses

Data	TLV Type	Extension Type	Value
None	ADDRESS_TYPE	0	UNREACHABLE
SeqNumList	SEQ_NUM	0	Sequence number associated with invalid route to the unreachable address.
MetricTypeList	PATH_METRIC	MetricType	None. Extension Type set to MetricType of the route to the unreachable address.

#### 10. Simple External Network Attachment

Figure 5 shows a stub (i.e., non-transit) network of AODVv2 routers which is attached to an external network via a single External Network Access Router (ENAR). The interface to the external network MUST NOT be configured in the InterfaceSet.

As in any externally-attached network, AODVv2 routers and Router Clients that wish to be reachable from the external network MUST have IP addresses within the ENAR's routable and topologically correct prefix (e.g., 191.0.2.0/24 in Figure 5). This AODVv2 network and networks attached to routers within it will be advertised to the external network using procedures which are out of scope for this specification.

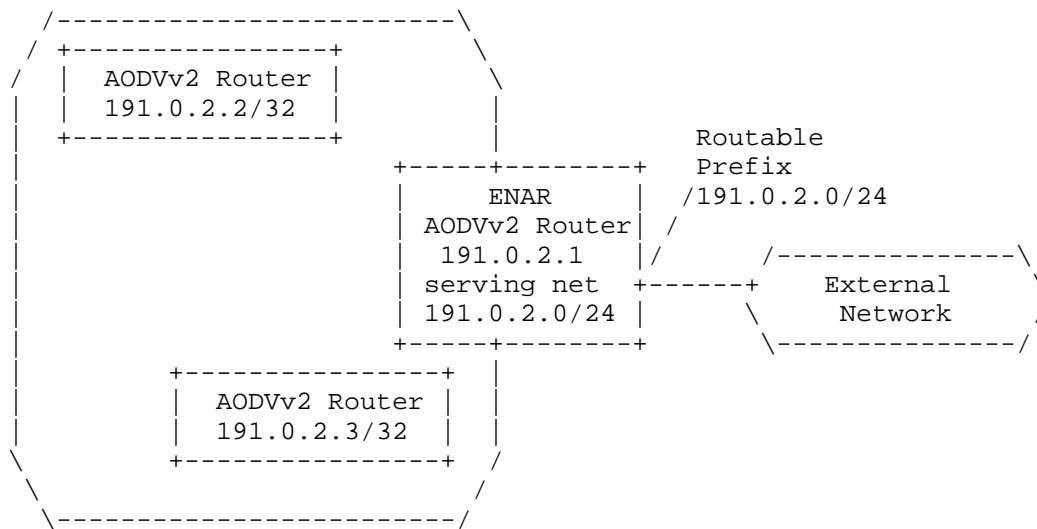


Figure 5: Simple External Network Attachment Example

When an AODVv2 router within the AODVv2 MANET wants to discover a route toward an address on the external network, it uses the normal AODVv2 route discovery for that IP Destination Address. The ENAR MUST respond to RREQ on behalf of all external network destinations, e.g., destinations not on the configured 191.0.2.0/24 network. The ENAR MAY respond with a TargPrefix and TargPrefixLen that represent a prefix including more addresses than just TargAddr, but MUST NOT respond with a TargPrefix and TargPrefixLen which includes any of the networks configured as part of the AODVv2 network. This does result in some inefficiencies in the way external routes are discovered. Sending a Route Request for a gateway is not currently supported.

RREQs for addresses inside the AODVv2 network, e.g. destinations on the configured 191.0.2.0/24 network, are handled using the standard processes described in Section 8. Note that AODVv2 does not support RREQs for prefixes that do not equal address length, but RREPs do advertise the prefix on which TargAddr resides.

When an IP packet from an address on the external network destined for an address in the AODVv2 MANET reaches the ENAR, if the ENAR does not have a route toward that destination in its Routing Information Base, it will perform normal AODVv2 route discovery for that destination.

Configuring the ENAR as a default router is outside the scope of this specification.

## 11. Configuration

AODVv2 uses various parameters which can be grouped into the following categories:

- o Timers
- o Protocol constants
- o Administrative parameters and controls

This section show the parameters along with their definitions and default values (if any).

Note that several fields have limited size (bits or bytes). These sizes and their encoding may place specific limitations on the values that can be set.

### 11.1. Timers

AODVv2 requires certain timing information to be associated with Local Route Set entries and message replies. The default values are as follows:

Name	Default Value
ACTIVE_INTERVAL	5 second
MAX_IDLETIME	200 seconds
MAX_BLACKLIST_TIME	200 seconds
MAX_SEQNUM_LIFETIME	300 seconds
RERR_TIMEOUT	3 seconds
RteMsg_ENTRY_TIME	12 seconds
RREQ_WAIT_TIME	2 seconds
RREP_Ack_SENT_TIMEOUT	1 second
RREQ_HOLDDOWN_TIME	10 seconds

Table 2: Timing Parameter Values

The above timing parameter values have worked well for small and medium well-connected networks with moderate topology changes. The timing parameters SHOULD be administratively configurable. Ideally, for networks with frequent topology changes the AODVv2 parameters SHOULD be adjusted using experimentally determined values or dynamic adaptation. For example, in networks with infrequent topology changes MAX\_IDLETIME MAY be set to a much larger value. If the

values were configured differently, the following consequences may be observed:

- o If `MAX_SEQNUM_LIFETIME` was configured differently across the network, and any of the routers lost their sequence number or rebooted, this could result in their next route messages being classified as stale at any AODVv2 router using a greater value for `MAX_SEQNUM_LIFETIME`. This would delay route discovery from and to the re-initializing router.
- o Routers with lower values for `ACTIVE_INTERVAL + MAX_IDLETIME` will invalidate routes more quickly and free resources used to maintain them. This can affect bursty traffic flows which have quiet periods longer than `ACTIVE_INTERVAL + MAX_IDLETIME`. A route which has timed out due to perceived inactivity is not reported. When the bursty traffic resumes, it would cause a RERR to be generated, and the traffic itself would be dropped. This route would be removed from all upstream routers, even if those upstream routers had larger `ACTIVE_INTERVAL` or `MAX_IDLETIME` values. A new route discovery would be required to re-establish the route, causing extra routing protocol traffic and disturbance to the bursty traffic.
- o Routers with lower values for `MAX_BLACKLIST_TIME` would allow neighboring routers to participate in route discovery sooner than routers with higher values. This could result in failed route discoveries if un-blacklisted links are still uni-directional. Since RREQs are retried, this would not affect success of route discovery unless this value was so small as to un-blacklist the router before the RREQ is retried. This value need not be consistent across the network since it is used for maintaining a 1-hop blacklist. However it MUST be greater than `RREQ_WAIT_TIME`.
- o Routers with lower values for `RERR_TIMEOUT` may create more RERR messages than routers with higher values. This value should be large enough that a RERR will reach all routers using the route reported within it before the timer expires, so that no further data traffic will arrive, and no duplicated RERR messages will be generated.
- o Routers with lower values for `RteMsg_ENTRY_TIME` may not consider received redundant multicast route messages as redundant, and may forward these messages unnecessarily.
- o Routers with lower values for `RREQ_WAIT_TIME` may send more frequent RREQ messages and wrongly determine that a route does not exist, if the delay in receiving an RREP is greater than this interval.

- o Routers with lower values for RREP\_Ack\_SENT\_TIMEOUT may wrongly determine links to neighbors to be unidirectional if an RREP\_Ack is delayed longer than this timeout.
- o Routers with lower values for RREQ\_HOLDDOWN\_TIME will retry failed route discoveries sooner than routers with higher values. This may be an advantage if the network topology is frequently changing, or may unnecessarily cause more routing protocol traffic.

MAX\_SEQNUM\_LIFETIME MUST be configured to have the same values for all AODVv2 routers in the network.

### 11.2. Protocol Constants

AODVv2 protocol constants typically do not require changes. The following table lists these constants, along with their values and a reference to the section describing their use.

Name	Default	Description
DISCOVERY_ATTEMPTS_MAX	3	Section 7.6
RREP_RETRIES	2	Section 8.2.1
MAX_METRIC[MetricType]	[TBD]	Section 6
MAX_METRIC[HopCount]	255	Section 6 and Section 8
MAX_HOPCOUNT	20	Limit to number of hops an RREQ or RREP message can traverse
INFINITY_TIME	[TBD]	Maximum expressible clock time (Section 7.7.2)

Table 3: AODVv2 Constants

MAX\_HOPCOUNT cannot be larger than 255.

MAX\_METRIC[MetricType] MUST always be the maximum expressible metric value of type MetricType. Field lengths associated with metric values are found in Section 11.5.

These protocol constants MUST have the same values for all AODVv2 routers in the ad hoc network. If the values were configured differently, the following consequences may be observed:

- o DISCOVERY\_ATTEMPTS\_MAX: Routers with higher values are likely to be more successful at finding routes, at the cost of additional control traffic.

- o RREP\_RETRIES: Routers with lower values are more likely to blacklist neighbors when there is a temporary fluctuation in link quality.
- o MAX\_METRIC[MetricType]: No interoperability problems due to variations on different routers, but routers with lower values may exhibit overly restrictive behavior during route comparisons.
- o MAX\_HOPCOUNT: Routers with a value too small would not be able to discover routes to distant addresses.
- o INFINITY\_TIME: No interoperability problems due to variations on different routers, but if a lower value is used, route state management may exhibit overly restrictive behavior.

### 11.3. Local Settings

The following table lists AODVv2 parameters which SHOULD be administratively configured for each router:

Name	Default Value	Description
InterfaceSet		Section 5.1
Router Client Set		Section 5.2
BUFFER_SIZE_PACKETS	2	Section 7.6
BUFFER_SIZE_BYTES	MAX_PACKET_SIZE [TBD]	Section 7.6
CONTROL_TRAFFIC_LIMIT	[TBD - 50 pkts/sec?]	Section 8

Table 4: Configuration for Local Settings

### 11.4. Network-Wide Settings

The following administrative controls MAY be used to change the operation of the network. The same settings SHOULD be used across the network. Inconsistent settings at different routers in the network will not result in protocol errors, but poor performance may result.

Name	Default	Description
ENABLE_IDLE_IN_RERR	Disabled	Section 8.4.1

Table 5: Configuration for Network-Wide Settings



### 11.5. MetricType Allocation

The metric types used by AODVv2 are identified according to Table 6. All implementations MUST use these values.

Name of MetricType	Type	Metric Value Size
Unassigned	0	Undefined
Hop Count	1	1 octet
Unallocated	2 - 254	TBD
Reserved	255	Undefined

Table 6: AODVv2 Metric Types

### 11.6. RFC 5444 Message Type Allocation

This specification defines four Message Types, to be allocated from the Experimental range of the "Message Types" namespace defined in [RFC5444], as specified in Table 7.

Name of Message	Type
Route Request (RREQ)	224
Route Reply (RREP)	225
Route Error (RERR)	226
Route Reply Acknowledgement (RREP_Ack)	227

Table 7: AODVv2 Message Types

If the AODVv2 experiment proves to be successful, types from the 0-223 range can be allocated in the future.

### 11.7. RFC 5444 Message TLV Types

This specification defines one Message TLV Type, to be allocated from the Message-Type-specific "Message TLV Types" namespace defined in [RFC5444], as specified in Table 8.

Name of TLV	Type	Length (octets)	Reference
ACK_REQ	128 (TBD)	0	Section 7.2

Table 8: AODVv2 Message TLV Types

## 11.8. RFC 5444 Address Block TLV Type Allocation

This specification defines three Address Block TLV Types, to be allocated from the Message-Type-specific "Address Block TLV Types" namespace defined in [RFC5444], as specified in Table 9.

Name of TLV	Type	Length (octets)	Reference
PATH_METRIC	129 (TBD)	depends on MetricType	Section 8
SEQ_NUM	130 (TBD)	2	Section 8
ADDRESS_TYPE	131 (TBD)	1	Section 9

Table 9: AODVv2 Address Block TLV Types

## 11.9. ADDRESS\_TYPE TLV Values

These values are used in the [RFC5444] Address Type TLV discussed in Section 9. All implementations MUST use these values.

Address Type	Value
ORIGPREFIX	0
TARGPREFIX	1
UNREACHABLE	2
PKTSOURCE	3
UNSPECIFIED	255

Table 10: AODVv2 Address Types

## 12. IANA Considerations

This document has no IANA actions.

## 13. Security Considerations

This section describes various security considerations and potential avenues to secure AODVv2 routing. The main objective of the AODVv2 protocol is for each router to communicate reachability information about addresses for which it is responsible, and for routes it has learned from other AODVv2 routers.

Networks using AODVv2 to maintain connectivity and establish routes on demand may be vulnerable to certain well-known types of threats, which will be detailed in the following. Some of the threats described can be mitigated or eliminated. Tools to do so will be described also.

With the exception of metric values, AODVv2 assures the integrity of all RteMsg data end-to-end through the use of ICVs (see Section 13.4.2).

The on-demand nature of AODVv2 route discovery automatically reduces the vulnerability to route disruption. Since control traffic for updating route tables is diminished, there is less opportunity for attack and failure.

### 13.1. Availability

Threats to AODVv2 which reduce availability are considered below.

#### 13.1.1. Denial of Service

Flooding attacks using RREQ amount to a (BLIND) denial of service for route discovery: By issuing RREQ messages for targets that don't exist, an attacker can flood the network, blocking resources and drowning out legitimate traffic. By triggering the generation of CONTROL\_TRAFFIC\_LIMIT amount of messages (for example by sending RREQs for many non-existent destinations), an attacker can prevent legitimate messages from being generated. The effect of this attack is dampened by the fact that duplicate RREQ messages are dropped (preventing the network from DDoSing itself). Processing requirements for AODVv2 messages are typically quite small, however AODVv2 routers receiving RREQs do allocate resources in the form of Neighbor Set, Local Route Set and Multicast Route Message Set entries. The attacker can maximize their impact on set growth by changing OrigPrefix or OrigPrefixLen for each RREQ. If a specific node is to be targeted, this attack may be carried out in a

DISTRIBUTED fashion, either by compromising its direct neighbors or by specifying the target's address with TargPrefix and TargPrefixLen. Note that it might be more economical for the attacker to simply jam the medium; an attack which AODVv2 cannot defend itself against.

Mitigation:

- o If AODVv2 routers always verify that the sender of the RERR message is trusted, this threat is reduced. Processing requirements would typically be dominated by calculations to verify integrity. This has the effect of reducing (but by no means eliminating) AODVv2's vulnerability to denial of service attacks.
- o Authentication of senders can prevent unauthenticated routers from launching a Denial of Service attack on another AODVv2 router. However, this does not protect the network if an attacker has access to an already authenticated router.

#### 13.1.2. Malicious RERR messages

RERR messages are designed to cause removal of installed routes. A malicious node could send an RERR message with false information to attempt to get other routers to remove a route to one or more specific destinations, therefore disrupting traffic to the advertised destinations.

Routes will be deleted if an RERR is received, withdrawing a route for which the sender is the receiver's next hop, and when the RERR includes the MetricType of the installed route, and includes either no sequence number for the route, or includes a greater sequence number than the sequence number stored with that route in the receiver's Local Route Set. Routes will also be deleted if a received RERR contains a PktSource address corresponding to a Router Client.

The information necessary to construct a malicious RERR could be learned by eavesdropping, either by listening to AODVv2 messages or by watching data packet flows.

When the RERR is multicast, it can be received by many routers in the ad hoc network, and will be regenerated when processing results in an active route being removed. This threat could have serious impact on applications communicating by way of the sender of the RERR message.

- o The set of routers which use the malicious router as a next hop may be targeted with a malicious RERR with no PktSource address included, if the RERR contains routes for which the malicious router is a next hop from the receiving router. However, since

the sender of the RERR message is either malicious or broken, it is better that it is not used as a next hop for these routes anyway.

- o A single router which does not use the malicious router as part of its route may be targeted with a malicious RERR with a PktSource address included.
- o Replayed RERR messages could be used to disrupt active routes.

Mitigation:

- o Protection against eavesdropping of AODVv2 messages would mitigate this attack to some extent, but eavesdropping of data packets can also be used to deduce the information about which routes could be targeted.
- o Protection against a malicious router becoming part of a route will mitigate the attack where a set of routers are targeted. This will not protect against the attack if a PktSource address is included.
- o By only regenerating RERR messages where active routes are removed, the spread of the malicious RERR is limited.
- o Including sequence numbers in RERR messages offers protection against attacks using replays of these RERR messages.
- o If AODVv2 routers always verify that the sender of the RERR message is trusted, this threat is reduced.

#### 13.1.3. False Confirmation of Link Bidirectionality

Links could be erroneously treated as bidirectional if malicious unsolicited or spoofed RREP messages were to be accepted. This would result in a route being installed which could not in fact be used to forward data to the destination, and may divert data packets away from the intended destination.

There is a window of RREQ\_WAIT\_TIME after an RREQ is sent, in which any malicious router could send an RREP in response, in order for the link to the malicious router to be deemed as bidirectional.

Mitigation:

- o Ignoring unsolicited RREP and RREP\_Ack messages partially mitigates against this threat.

- o If AODVv2 routers always verify that the sender of the RERR message is trusted, this threat is reduced.

#### 13.1.4. Message Deletion

A malicious router could decide not to forward an RREQ or RREP or RERR message. Not forwarding a RERR or RREP message would disrupt route discovery. Not regenerating a RERR message would result in the source of data packets continuing to maintain and use the route, and further RERR messages being generated by the sender of the non-regenerated RERR. A malicious router could intentionally disrupt traffic flows by not allowing the source of data traffic to re-discover a new route when one breaks.

Failing to send an RREP\_Ack would also disrupt route establishment, by not allowing the reverse route to be validated. Return traffic which needs that route will prompt a new route discovery, wasting resources and incurring a slight delay but not disrupting the ability for applications to communicate.

Mitigation:

- o None. also note that malicious router would have to wait for a route to break before it could perform this attack.

#### 13.2. Confidentiality

Passive inspection (eavesdropping) of AODVv2 control messages could enable unauthorized devices to gain information about the network topology, since exchanging such information is the main purpose of AODVv2.

Eavesdropping of data traffic could allow a malicious device to obtain information about how data traffic is being routed. With knowledge of source and destination addresses, malicious messages could be constructed to disrupt normal operation.

#### 13.3. Integrity

Integrity of route information can be compromised in the following types of attack:

##### 13.3.1. Message Insertion

Valid route set entries can be replaced or modified by maliciously constructed AODVv2 messages, destroying existing routes and the network's integrity. Any router may pose as another router by sending RREQ, RREP, RREP\_Ack and RERR messages in its name.

- o Sending an RREQ message with false information can disrupt traffic to OrigPrefix, if the sequence number attached is not stale compared to any existing information about OrigPrefix. Since RREQ is multicast and likely to be received by all routers in the ad hoc network, this threat could have serious impact on applications communicating with OrigPrefix. The actual threat to disrupt routes to OrigPrefix is reduced by the AODVv2 mechanism of marking RREQ-derived routes as "Unconfirmed" until the link to the next hop is confirmed.
- o Sending an RREP message with false information can disrupt traffic to TargPrefix. Since RREP is unicast, and ignored if a corresponding RREQ was not recently sent, this threat is minimized, and is restricted to receivers along the path from OrigAddr to TargAddr.
- o Sending an RREP\_Ack response message with false information can cause the route to an originator address to be erroneously accepted even though the route would contain a unidirectional link and thus not be suitable for most traffic. Since the RREP\_Ack response is unicast, and ignored if a RREP\_Ack was not sent recently to the sender of this RREP\_Ack response, this threat is minimized and is strictly local to the RREP transmitter expecting the acknowledgement. Unsolicited RREP\_Acks are ignored.
- o Sending an RERR message with false information is discussed in Section 13.1.2.

Mitigation:

- o If AODVv2 routers always verify that the sender of a message is trusted, this threat is reduced.

### 13.3.2. Message Modification - Man in the Middle

Any AODVv2 router can forward messages with modified data.

Mitigation:

- o If AODVv2 routers verify the integrity of AODVv2 messages, then the threat of disruption is minimized. A man in the middle with no knowledge of the key used to calculate an integrity check value may modify a message but the message will be rejected when it fails an integrity check.

### 13.3.3. Replay Attacks

Replaying of RREQ or RREP messages would be of less use to an attacker, since they would be dropped immediately due to their stale sequence number. RERR messages may or may not include sequence numbers and are therefore susceptible to replay attacks. RREP\_Ack messages do not include sequence numbers and are therefore susceptible to replay attacks.

Mitigation:

- o Use of timestamps or sequence numbers prevents replay attacks.

### 13.4. Protection Mechanisms

#### 13.4.1. Confidentiality and Authentication

Encryption MAY be used for AODVv2 messages. If the routers share a packet-level security association, the message data can be encrypted prior to message transmission. The establishment of such security associations is outside the scope of this specification. Encryption will not only protect against unauthorized devices obtaining information about network topology (eavesdropping) but will ensure that only trusted routers participate in routing operations.

#### 13.4.2. Integrity and Trust using ICVs

Cryptographic Integrity Check Values (ICVs) can be used to ensure integrity of received messages, protecting against man in the middle attacks. Further, by using ICVs, only those routers with knowledge of a shared secret key are allowed to participate in routing information exchanges. [RFC7182] defines ICV TLVs for use with [RFC5444].

The data contained in AODVv2 routing protocol messages MUST be verified using Integrity Check Values, to avoid the use of message data if the message has been tampered with.

#### 13.4.3. Replay Protection using Timestamps

Replay attacks MUST be prevented by using timestamps or sequence numbers in messages. [RFC7182] defines a TIMESTAMP TLV for use with [RFC5444].

The data contained in AODVv2 routing protocol messages MUST be protected with a TIMESTAMP value to ensure the protection against replaying of the message. Sequence numbers can be used as timestamps, since they are known to be strictly increasing.



#### 13.4.4. Application to AODVv2

AODVv2 implementations MUST support ICV and TIMESTAMP TLVs, unless the implementation is intended solely for an environment in which security is unnecessary. AODVv2 deployments SHOULD be configured to use these TLVs to secure messages.

Implementations of AODVv2 MUST support ICV TLVs using type-extensions 1 and 2, hash-function HASH\_FUNCTION, and cryptographic function CRYPTOGRAPHIC\_FUNCTION. An ICV MUST be included with every message. The ICV value MAY be truncated as specified in [RFC7182].

Since the msg-hop-limit and PATH\_METRIC values are mutable when included in AODVv2 messages, these values MUST be set to zero before calculating an ICV. This means that these values are not protected end-to-end and are therefore susceptible to manipulation. This form of attack is described in Section 13.3.2.

Implementations of AODVv2 MUST support a TIMESTAMP TLV using type-extension 0. The timestamp used is a sequence number, and therefore the length of the <TIMESTAMP-value> field matches the AODVv2 sequence number defined in Section 5.4. The TIMESTAMP TLV MUST be included in RREP\_Ack and RERR messages.

When more than one message is included in an RFC5444 packet, using a single ICV Packet TLV or single TIMESTAMP Packet TLV is more efficient than including ICV and TIMESTAMP Message TLVs in each message created. If the RFC5444 multiplexer is capable of adding the Packet TLVs, it SHOULD be instructed to include the Packet TLVs in packets containing AODVv2 messages. However, if the multiplexer is not capable of adding the Packet TLVs, the TLVs MUST be included as Message TLVs in each AODVv2 message in the packet.

After message generation but before transmission, the ICV and TIMESTAMP TLVs MUST be added according to each message type as detailed in the following sections. The following steps list the procedure to be performed:

1. If the TIMESTAMP is to be included, depending on AODVv2 message type as specified below, add the TIMESTAMP TLV.
  - o When a TIMESTAMP Packet TLV is being added, the Packet TLV Block size field MUST be updated.
  - o When a TIMESTAMP Message TLV is being added, the Message TLV Block size field MUST be updated.

1. The considerations in Section 8 and section 9 of [RFC7182] are followed, removing existing ICV TLVs and adjusting the size and flags fields as appropriate:
  - o When an ICV Packet TLV is being added, existing ICV Packet TLVs MUST be removed and the Packet TLV Block size MUST be updated. If the Packet TLV Block now contains no TLVs, the phastlv bit in the <pkt-flags> field in the Packet Header MUST be cleared.
  - o When an ICV Message TLV is being added, existing ICV Message TLVs are removed and the Message TLV Block Size MUST be updated.
1. Mutable fields in the message MUST have their mutable values set to zero before calculating the ICV.
  - o If the msg-hop-limit field is included in the [RFC5444] message header, msg-hop-limit MUST be set to zero before calculating the ICV.
  - o If a PATH\_METRIC TLV is included, any values present in the TLV MUST be set to zero before calculating the ICV value.
1. Depending on the message type, the ICV is calculated over the appropriate fields (as specified in sections Section 13.4.4.1, Section 13.4.4.2, Section 13.4.4.3 and Section 13.4.4.4) to include the fields <hash-function>, <cryptographic-function>, <key-id-length>, and, if present, <key-id> (in that order), followed by the entire packet or message. This value MAY be truncated (as specified in [RFC7182]).
2. Add the ICV TLV, updating size fields as necessary.
3. The changes made in Step 2 and Step 3 are reversed to re-add any existing ICV TLVs, re-adjust the relevant size and flags fields, and set the msg-hop-limit and PATH\_METRIC TLV values.

On message reception, and before message processing, verification of the received message MUST take place:

1. The considerations in Section 8 and Section 9 of [RFC7182] are followed, removing existing ICV TLVs and adjusting the size and flags fields as appropriate.
  - o When verifying the ICV value in an ICV Packet TLV, all ICV Packet TLVs present in the Packet TLV Block MUST be removed before calculating the ICV, and the Packet TLV Block size MUST be updated. If there are no remaining Packet TLVs, the Packet TLV

Block MUST be removed and the phastlv bit in the <pkt-flags> field MUST be cleared.

- o When verifying the ICV value in an ICV Message TLV, all ICV Message TLVs present in the Message TLV Block MUST be removed before calculating the ICV, and the Message TLV Block size MUST be updated.
- 1. Mutable fields in the message MUST have their mutable values set to zero before calculating the ICV.
- o If the msg-hop-limit field is included in the [RFC5444] message header, msg-hop-limit MUST be set to zero before calculating the ICV.
- o If a PATH\_METRIC TLV is included, any values present in the TLV MUST be set to zero before calculating the ICV value.
- 1. The ICV is calculated following the considerations in Section 12.2 of [RFC7182], to include the fields <hash-function>, <cryptographic-function>, <key-id-length>, and, if present, <key-id> (in that order), followed by the entire packet or message.
- o If the received ICV value is truncated, the calculated ICV value MUST also be truncated (as specified in [RFC7182]), before comparing.
- o If the ICV value calculated from the received message or packet does not match the value of <ICV-data> in the received message or packet, the validation fails and the AODVv2 message MUST be discarded and NOT processed or forwarded.
- o If the ICV values do match, the values set to zero before calculating the ICV are reset to the received values, and processing continues to Step 4.
- 1. Verification of a received TIMESTAMP value MUST be performed. The procedure depends on message type as specified in the following sub sections.
- o If the TIMESTAMP value in the received message is not valid, the AODVv2 message MUST be discarded and NOT processed or forwarded.
- o If the TIMESTAMP value is valid, processing continues as defined in Section 7.

#### 13.4.4.1. RREQ Generation and Reception

Since OrigPrefix is included in the RREQ, the ICV can be calculated and verified using the [RFC5444] contents. The ICV TLV has type extension := 1. Inclusion of an ICV TLV provides message integrity and endpoint authentication, because trusted routers MUST hold the shared key in order to calculate the ICV value, both to include when creating a message, and to validate the message by checking that the ICV is correct.

Since RREQ\_Gen's sequence number is incremented for each new RREQ, replay protection is already afforded and no extra TIMESTAMP TLV is required.

After message generation and before message transmission:

1. Add the ICV TLV as described above.

On message reception and before message processing:

1. Verify the received ICV value as described above.
2. Verification of the sequence number is handled according to Section 7.

#### 13.4.4.2. RREP Generation and Reception

Since TargPrefix is included in the RREP, the ICV can be calculated and verified using the [RFC5444] contents. The ICV TLV has type extension := 1. Inclusion of an ICV provides message integrity and endpoint authentication, because trusted routers MUST hold a valid key in order to calculate the ICV value, both to include when creating a message, and to validate the message by checking that the ICV is correct.

Since RREP\_Gen's sequence number is incremented for each new RREP, replay protection is already afforded and no extra TIMESTAMP TLV is required.

After message generation and before message transmission:

1. Add the ICV TLV as described above.

On message reception and before message processing:

1. Verify the received ICV value as described above.

2. Verification of the sequence number is handled according to Section 7.

#### 13.4.4.3. RREP\_Ack Generation and Reception

Since no sequence number is included in the RREP\_Ack, a TIMESTAMP TLV MUST be included to protect against replay attacks. The value in the TIMESTAMP TLV is set as follows:

- o For RREP\_Ack request, use Neighbor.AckSeqNum.
- o For RREP\_Ack response, use the sequence number from the TIMESTAMP TLV in the received RREP\_Ack request.

Since no addresses are included in the RREP\_Ack, and the receiver of the RREP\_Ack uses the source IP address of a received RREP\_Ack to identify the sender, the ICV MUST be calculated using the message contents and the IP source address. The ICV TLV has type extension := 2 in order to accomplish this. This provides message integrity and endpoint authentication, because trusted routers MUST hold the correct key in order to calculate the ICV value.

After message generation and before message transmission:

1. Add the TIMESTAMP TLV and ICV TLV as described above.

On message reception and before message processing:

1. Verify the received ICV value as described above.
2. Verify the received TIMESTAMP value by comparing the sequence number in the value field of the TIMESTAMP TLV as follows:
  - o For a received RREP\_Ack request, there is no need to verify the timestamp value. Proceed to message processing as defined in Section 7.
  - o For a received RREP\_Ack response, compare with the Neighbor.AckSeqNum of the Neighbor Set entry for sender of the RREP\_Ack request.
  - o If the sequence number does not match, the AODVv2 message MUST be discarded. Otherwise, Neighbor.AckSeqNum is incremented by 1 and processing continues according to Section 7.

#### 13.4.4.4. RERR Generation and Reception

Since the sender's sequence number is not contained in the RERR, a **TIMESTAMP TLV** **MUST** be included to protect against replay attacks. The value in the **TIMESTAMP TLV** is set by incrementing and using **RERR\_Gen's** sequence number.

Since the receiver of the RERR **MUST** use the source IP address of the RERR to identify the sender, the **ICV** **MUST** be calculated using the message contents and the IP source address. The **ICV TLV** has type extension := 2 in order to accomplish this. This provides message integrity and endpoint authentication, because trusted routers **MUST** hold the shared key in order to calculate the **ICV** value.

After message generation and before message transmission:

1. Add the **TIMESTAMP TLV** and **ICV TLV** as described above.

On message reception and before message processing:

1. Verify the received **ICV** value as described above.
2. Verify the received **TIMESTAMP** value by comparing the sequence number in the value field of the **TIMESTAMP TLV** with the **Neighbor.HeardRERRSeqNum**. If the sequence number in the message is lower than the stored value, the AODVv2 message **MUST** be discarded. Otherwise, the **Neighbor.HeardRERRSeqNum** **MUST** be set to the received value and processing continues according to Section 7.

#### 13.5. Key Management

The method of distribution of shared secret keys is out of the scope of this protocol. Key management is not specified for the following reasons:

Against [RFC4107], an analysis as to whether automated or manual key management should be used shows a compelling case for automated management. In particular:

- o a potentially large number of routers may have to be managed, belonging to several organisations, for example in vehicular applications.
- o a stream cipher is likely to be used, such as an AES variant.

- o long term session keys might be used by more than two parties, including multicast operations. AODVv2 makes extensive use of multicast.
- o there may be frequent turnover of devices.

On reviewing the case for manual key management against the same document, it can be seen that manual management might be advantageous in environments with limited bandwidth or high round trip times. AODVv2 lends itself to sparse ad hoc networks where transmission conditions may indeed be limited, depending on the bearers selected for use.

However, [RFC4107] assumes that the connectivity between endpoints is already available. In AODVv2, no route is available to a given destination until a router client requests that user traffic be transmitted. It is required to secure the signalling path of the routing protocol that will establish the path across which key exchange functions might subsequently be applied, which is clearly the reverse of the expected functionality. A different strategy is therefore required.

There are two possible solutions. In each case, it is assumed that a defence in depth security posture is being adopted by the system integrator, such that each function in the network as a whole is appropriately secured or defended as necessary, and that there is not complete reliance on security mechanisms built in to AODVv2. Such additional mechanisms could include a suitable wireless device security technology, so that wireless devices are authenticated and secured by their peers prior to exchanging user data, which in this case would include AODVv2 signalling traffic as a payload, and mechanisms which verify the authenticity and/or integrity of application-layer user data transported once a route has been established.

1. In the case that no AODVv2 routers have any detailed prior knowledge of any other AODVv2 router, but does have knowledge of the credentials of other organisations in which the router has been previously configured to trust, it is possible for an AODVv2 router to send an initialisation vector as part of an exchange, which could be verified against such credentials. Such an exchange could make use of Identity-Based Signatures ([I-D.ietf-manet-ibs]), based on Elliptic Curve-Based Certificateless Signatures for Identity-Based Encryption [RFC6507], which eliminate the need for a handshake process to establish trust.

2. If it is impossible to use Identity-Based Signatures, and the risk to the AODVv2 signalling traffic is considered to be low due to the use of security countermeasures elsewhere in the system, a simple pre-placed shared secret could be used between routers, which is used as-is or is used to generate some ephemeral secret based on another known variable, such as time of day if that is universally available at a level of accuracy sufficient to make such a system viable.

#### 14. Acknowledgments

AODVv2 is a descendant of the design of previous MANET on-demand protocols, especially AODV [RFC3561] and DSR [RFC4728]. Changes to previous MANET on-demand protocols stem from research and implementation experiences. Thanks to Elizabeth Belding and Ian Chakeres for their long time authorship of AODV. Additional thanks to Derek Atkins, Emmanuel Baccelli, Abdussalam Baryun, Ramon Caceres, Justin Dean, Christopher Dearlove, Fatemeh Ghassemi, Ulrich Herberg, Henner Jakob, Ramtin Khosravi, Luke Klein-Berndt, Lars Kristensen, Tronje Krop, Koojana Kuladinithi, Kedar Namjoshi, Keyur Patel, Alexandru Petrescu, Henning Rogge, Fransisco Ros, Pedro Ruiz, Christoph Sommer, Romain Thouvenin, Richard Trefler, Jiazi Yi, Seung Yi, Behnaz Yousefi, and Cong Yuan, for their reviews of AODVv2 and DYMO, as well as numerous specification suggestions.

#### 15. References

##### 15.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<http://www.rfc-editor.org/info/rfc2119>>.
- [RFC3561] Perkins, C., Belding-Royer, E., and S. Das, "Ad hoc On-Demand Distance Vector (AODV) Routing", RFC 3561, DOI 10.17487/RFC3561, July 2003, <<http://www.rfc-editor.org/info/rfc3561>>.
- [RFC5444] Clausen, T., Dearlove, C., Dean, J., and C. Adjih, "Generalized Mobile Ad Hoc Network (MANET) Packet/Message Format", RFC 5444, DOI 10.17487/RFC5444, February 2009, <<http://www.rfc-editor.org/info/rfc5444>>.
- [RFC5498] Chakeres, I., "IANA Allocations for Mobile Ad Hoc Network (MANET) Protocols", RFC 5498, DOI 10.17487/RFC5498, March 2009, <<http://www.rfc-editor.org/info/rfc5498>>.



- [RFC7182] Herberg, U., Clausen, T., and C. Dearlove, "Integrity Check Value and Timestamp TLV Definitions for Mobile Ad Hoc Networks (MANETs)", RFC 7182, DOI 10.17487/RFC7182, April 2014, <<http://www.rfc-editor.org/info/rfc7182>>.

## 15.2. Informative References

- [I-D.ietf-manet-ibs]  
Dearlove, C., "Identity-Based Signatures for MANET Routing Protocols", draft-ietf-manet-ibs-05 (work in progress), March 2016.
- [Koodli01]  
Koodli, R. and C. Perkins, "Fast handovers and context transfers in mobile networks", Proceedings of the ACM SIGCOMM Computer Communication Review 2001, Volume 31 Issue 5, 37-47, October 2001.
- [Perkins94]  
Perkins, C. and P. Bhagwat, "Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers", Proceedings of the ACM SIGCOMM '94 Conference on Communications Architectures, Protocols and Applications, London, UK, pp. 234-244, August 1994.
- [RFC2501] Corson, S. and J. Macker, "Mobile Ad hoc Networking (MANET): Routing Protocol Performance Issues and Evaluation Considerations", RFC 2501, DOI 10.17487/RFC2501, January 1999, <<http://www.rfc-editor.org/info/rfc2501>>.
- [RFC4107] Bellovin, S. and R. Housley, "Guidelines for Cryptographic Key Management", BCP 107, RFC 4107, DOI 10.17487/RFC4107, June 2005, <<http://www.rfc-editor.org/info/rfc4107>>.
- [RFC4193] Hinden, R. and B. Haberman, "Unique Local IPv6 Unicast Addresses", RFC 4193, DOI 10.17487/RFC4193, October 2005, <<http://www.rfc-editor.org/info/rfc4193>>.
- [RFC4728] Johnson, D., Hu, Y., and D. Maltz, "The Dynamic Source Routing Protocol (DSR) for Mobile Ad Hoc Networks for IPv4", RFC 4728, DOI 10.17487/RFC4728, February 2007, <<http://www.rfc-editor.org/info/rfc4728>>.
- [RFC6130] Clausen, T., Dearlove, C., and J. Dean, "Mobile Ad Hoc Network (MANET) Neighborhood Discovery Protocol (NHDP)", RFC 6130, DOI 10.17487/RFC6130, April 2011, <<http://www.rfc-editor.org/info/rfc6130>>.

[RFC6507] Groves, M., "Elliptic Curve-Based Certificateless Signatures for Identity-Based Encryption (ECCSI)", RFC 6507, DOI 10.17487/RFC6507, February 2012, <<http://www.rfc-editor.org/info/rfc6507>>.

#### Appendix A. AODVv2 Draft Updates

This section lists the changes between AODVv2 revisions ...-15.txt and ...-16.txt.

- o Changed 'regeneration' language in favor of 'forwarding'.
- o Reintroduced use of msg-hop-limit in 5444 message header.
- o Use OrigPrefix rather than OrigAddr and TargPrefix rather than TargAddr where appropriate
- o Removed validity time
- o Removed AckReq from RREP messages, use two-way RREP\_ack to check for bidirectionality
- o Unicast RREP messages
- o Removed orphaned references
- o Clarified language
- o Improved Sequence Number instructions
- o Changed 'Unknown' terminology to 'Heard'
- o Extended experiment description
- o Added detailed description of which steps to take when calculating and evaluating ICVs, particularly how to zero out the metric value

#### Authors' Addresses

Charles E. Perkins  
Futurewei Inc.  
2330 Central Expressway  
Santa Clara, CA 95050  
USA

Phone: +1-408-330-4586  
Email: [charliep@computer.org](mailto:charliep@computer.org)

Stan Ratliff  
Idirect  
13861 Sunrise Valley Drive, Suite 300  
Herndon, VA 20171  
USA

Email: [ratliffstan@gmail.com](mailto:ratliffstan@gmail.com)

John Dowdell  
Airbus Defence and Space  
Celtic Springs  
Newport, Wales NP10 8FZ  
United Kingdom

Email: [john.dowdell@airbus.com](mailto:john.dowdell@airbus.com)

Lotte Steenbrink  
HAW Hamburg, Dept. Informatik  
Berliner Tor 7  
D-20099 Hamburg  
Germany

Email: [lotte.steenbrink@haw-hamburg.de](mailto:lotte.steenbrink@haw-hamburg.de)

Victoria Mercieca  
Airbus Defence and Space  
Celtic Springs  
Newport, Wales NP10 8FZ  
United Kingdom

Email: [victoria.mercieca@airbus.com](mailto:victoria.mercieca@airbus.com)

Network Working Group  
Internet-Draft  
Intended status: Informational  
Expires: March 19, 2015

T. Clausen  
LIX, Ecole Polytechnique  
U. Herberg  
Fujitsu Laboratories of America  
September 15, 2014

Snapshot of OLSRv2-Routed MANET Management  
draft-ietf-manet-olsrv2-management-snapshot-03

Abstract

This document describes how Mobile Ad Hoc Networks (MANETs) are typically managed, in terms of pre-deployment management, as well as rationale and means of monitoring and management of MANET routers running the Optimized Link State Routing protocol version 2 (OLSRv2) and its constituent MANET Neighborhood Discovery Protocol (NHDP). Apart from pre-deployment management for setting up IP addresses and security related credentials, OLSRv2 only needs routers to agree one single configuration parameter (called "C"). Other parameters for tweaking network performance may be determined during operation of the network, and need not be the same in all routers. This, using MIB modules and related management protocols such as SNMP (or possibly other, less "chatty", protocols). In addition, for debugging purposes, monitoring data and performance related counters, as well as notifications ("traps") can be sent to the Network Management System (NMS) via standardized management protocols.

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on March 19, 2015.

Copyright Notice

Copyright (c) 2014 IETF Trust and the persons identified as the

document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

1. Introduction . . . . .	3
1.1. Statement of Purpose . . . . .	3
2. Terminology . . . . .	3
3. Pre-Deployment Management . . . . .	4
3.1. Lower Layer Alignment . . . . .	4
3.2. Interface Addresses . . . . .	4
3.3. Security Material . . . . .	5
3.4. The Value of C . . . . .	5
4. How do we Manage OLSRv2-based MANETs? . . . . .	6
4.1. Internal Management . . . . .	6
4.2. External Management . . . . .	6
5. What and Why do we Manage and Monitor? . . . . .	7
6. Typical Communication Patterns . . . . .	9
7. This Document does not Constrain how to Manage and Monitor MANETs . . . . .	11
8. IANA Considerations . . . . .	11
9. Security Considerations . . . . .	12
10. Acknowledgments . . . . .	12
11. Informative References . . . . .	12
Authors' Addresses . . . . .	14

## 1. Introduction

The MANET routing protocol OLSRv2 [RFC7181], as well as its constituent parts NHDP [RFC6130], [RFC5497], [RFC5148], [RFC5444], [RFC7182], [RFC7183], [RFC7187], [RFC7188] is designed to autonomously maintain routes across a dynamic network topology. OLSRv2 is designed so as to minimize operator intervention throughout the duration of a network deployment, and to allow for heterogeneous configuration of routers within the same network deployment: most configuration values are either of local significance only (e.g., message jitter parameters) or, when they are not, are carried in control signals exchanged between routers (e.g., information validity time).

All the same, a small set of configuration options must be established in each router prior to deployment, with some requiring agreement among all the routers within the same deployment. Furthermore, throughout the duration of a network deployment, external management and monitoring of a network may be desirable, e.g., for performance optimization or debugging purposes.

### 1.1. Statement of Purpose

Deployments of OLSRv2 are diverse, and may include community networks, constrained environments, tactical networks, etc. Each such environment may present distinctly different requirements as to management and monitoring.

This document does therefore explicitly not pretend to provide an exhaustive description of how all OLSRv2 network deployments should be managed and monitored - and does, specifically, not prescribe any management model. This document also does not address management of MANET routing protocols, other than OLSRv2 (and its constituent protocols).

What this document does, however, is to present how typical OLSRv2 network deployments are managed and monitored, using well-established management patterns and well-known protocols. In particular, this document addresses several of the considerations from [RFC5706], in particular Section 3 ("Management Considerations - How Will the Protocol Be Managed?").

## 2. Terminology

This document uses terminology from [RFC7181], [RFC6130], and [RFC5497].

### 3. Pre-Deployment Management

Prior to operation of an OLSRv2 network, or more precisely, prior to proper operation of OLSRv2 and its constituent parts, certain parameters need to be configured on each router. The following sections describe the required pre-deployment management.

#### 3.1. Lower Layer Alignment

Interoperability between routers requires alignment of lower protocol layers below OLSRv2. In particular, all routers in the same MANET topology must be pre-configured to use the same IP address family (IPv4 or IPv6). In a single OLSRv2 topology, it is not possible to mix IPv4 and IPv6 addresses, notably because [RFC5444] messages can contain either IPv4 *or* IPv6 addresses, but not both at the same time. It is, however, possible to run two instances of OLSRv2, one instance for IPv4 and another one for IPv6, within the same network.

In addition to the IP address family, other lower layer parameters may also need to be aligned, e.g., MAC as well as radio channel selections. A single OLSRv2 topology may, of course, span different link layers (or the same link layer with different configuration settings such as cryptographic keys) when routers in the topology have OLSRv2 interfaces towards these different link layers.

#### 3.2. Interface Addresses

According to [RFC6130], and as used by [RFC7181], each interface of a router must be configured with at least one IP address. [RFC6130] provides guidance as to the characteristics of such IP addresses, including the (limited) conditions under which a single IP address may be configured on multiple interfaces.

While automatic configuration of IP addresses on router interfaces is not excluded, currently no address autoconfiguration protocols have been standardized (in the IETF) to accomplish this. As a consequence, static configuration, or proprietary (as in: non-standardized) protocols ensure this.

Note that [RFC6130] and [RFC7181] permit to dynamically add or remove IP addresses as part of normal network operation. This applies for local MANET interfaces, as well as for local non-MANET interfaces or IP addresses from remote destinations reachable through this router (i.e., addresses for which this router serves as gateway). Interface addresses are managed by way of the Local Interface Set (as defined in [RFC6130]) and remote addresses by way of the Attached Network Set (as defined in [RFC7181]).

### 3.3. Security Material

Security material (keys, algorithms, etc.) must be available for generating Integrity Check Values (ICVs) for outgoing control messages, and to allow validating ICVs in incoming control messages [RFC7182] [RFC7183].

The appropriate way of making such security material available is dependent on the deployment type. For example, community networks (such as "Funkfeuer", <http://funkfeuer.at>), do currently not use any security at all. Other deployment types may use a simple manual shared key distribution mechanism, or may use a proprietary key distribution protocol. Tactical networks have much more stringent requirements for distributing key material, e.g., using manual distribution of the keys on encrypted USB flash drives, and with defensive mechanisms (up to and including mechanisms involving depleted uranium) if the keys are compromised.

In general, Automatic Key Management (AKM) as well as static/manual or other out-of-band mechanisms, can be viable options for distributing keys. Currently, no standardized AKM mechanism for MANETs exist. If the IETF standardizes such mechanisms in the future, for deployment types where such is appropriate, these can be used for distributing keys (with the obvious chicken-and-egg problem of using the routing fabric that is being constructed to distribute the keys to establish that fabric). Until such an AKM mechanism is standardized, manual key distribution as well as proprietary mechanisms prevail.

The important point to make here, however, is that by whichever method (automatic/manual, dynamic/static, ... ) a key and other security material is made available, the security mechanisms of OLSRv2, as defined by [RFC7183], will be able to properly use it for generating and validating ICVs.

### 3.4. The Value of C

The only pre-deployment configuration parameter that directly impacts protocol operation is the value of C. This value is used by each router for calculating the representation of interval and validity time, as included in control messages. All routers in a deployment must agree on the value of C, as described in [RFC5497]. Note that since all MANET routers inside a MANET must agree to the same value of C before deployment, C is denoted "constant" in [RFC5497] rather than "parameter" as in this document. From a management perspective, C can be considered as configuration parameter prior to operation of the routing protocol.



#### 4. How do we Manage OLSRv2-based MANETs?

A deployed OLSRv2 network is, as previously discussed, operating autonomously, but occasionally with internal or external management operations being desirable, described in the following two sections.

##### 4.1. Internal Management

Internal management describes a local process running on a router that automatically (i.e., without external messaging or human interaction) modifies the configuration of OLSRv2 based on different environmental factors. In particular, message intervals can be updated dynamically and without external management interaction, e.g., the HELLO interval may be updated according to the rate of topology changes measured (or, inferred: after all, the 'M' in MANET is for "Mobility") locally: if the rate is high, then a more frequent HELLO update assures that routes are more accurate. At a lower rate of topology changes, network capacity and energy capacity of the router may be conserved by increasing the HELLO interval. In addition to message intervals, minimum intervals can have a significant impact on the operation of OLSRv2, and therefore need to be adjusted with care. If, for instance, the minimum interval between two successive HELLO messages (HELLO\_MIN\_INTERVAL) is set too low, many messages may be sent within a short timeframe, potentially leading to frame collisions or exhaustion of the available bandwidth.

Depending on the use case, many different automatic configuration agents can be envisioned. As parameters in NHDP and OLSRv2 are either only used locally or, in the case of HELLO\_INTERVAL and REFRESH\_INTERVAL, are selected locally and then included in the messages exchanged between adjacent routers in their HELLO messages, none of these automatic local configuration methods needs necessarily to be standardized: different routers doing different things will interoperate.

##### 4.2. External Management

For the deployments described by this document (but see Section 7), external management operations are undertaken by a central Network Management Station (NMS).

The MIB modules developed for OLSRv2 [RFC7184] and for its constituent protocol NHDP [RFC6779] are verbose, in as much as that they expose for interrogation the complete protocol and router state, as well as enable setting all parameters (timer intervals, time-outs, jitter values etc.). They do explicitly not enable setting the value of C (as that is required to be constant and uniform across the network, see Section 3.4), nor distributing security material (see

Section 3.3).

In some deployments, the NMS communicates with individual routers by way of SNMP - and, more commonly, by way of "proprietary" simpler, less verbose and (often) less secure protocols, and over UDP. Note that this does not constitute a recommendation, but rather an observation that (apparently) SNMP has found less application in MANETs. The "Writable MIB Module IESG Statement" (<http://www.ietf.org/iesg/statement/writable-mib-module.html>) recommends to use MIB modules for read-only operations only, and to use YANG/NETCONF for read-write operations instead. While publication of the MIB modules developed for OLSRv2 and NHDP predates this statement, it may be possible to translate read-only objects from the MIB modules into YANG modules using [RFC6643]. A complete YANG model representing similar objects as in the MIB modules could be future work.

The predecessor of OLSRv2, OLSR [RFC3626] did not have an associated MIB module. Many deployments of OLSR did not support network management operations per se (i.e., configuration-on-launch was the way in which routers in these deployments were managed). Those implementations and deployments of OLSR that did support network management operations used a similar architecture to the one described in this document, but with "proprietary" protocols and APIs for parameters and router states, "proprietary" data-models, etc. It can be speculated that the "proprietary" protocols used for communication between the NMS and the MIB modules on each router also for OLSRv2, in part, exist as inherited from the protocols used for OLSR. Aligned with the recommendations from [RFC5706], management of OLSRv2 (in the form of the MIB modules for OLSRv2 and NHDP) has been developed alongside the standardization process of OLSRv2, rather than as an afterthought.

Finally, it is uncommon to see an NMS permanently active in a deployed OLSRv2 network; rather, on an "ad hoc" basis an NMS is introduced into the network, parameters configured or state interrogated, following which the NMS disappears. Part of the rationale for this is that in a MANET, network connectivity from every MANET router to an NMS cannot be guaranteed at all times due to the dynamicity of the network topology.

## 5. What and Why do we Manage and Monitor?

As indicated earlier, OLSRv2 and its constituent protocol NHDP, are reasonably robust with respect to parameter values: a deployment can operate with different parameters used in different routers in the same network. That being said, adapting these parameters according

to circumstances is (often) desired. For example, in a stable network (such as a wired network), TC messages may be sent infrequently and with long validity times, whereas in a highly dynamic network (such as in a vehicular network) TC messages may need to be sent more frequently and HELLO messages for discovering the local topology (almost) continuously. Note that for highly dynamic topologies, an alternative to sending control messages very frequently is to use long message intervals in combination with all of the permitted responsive mechanisms (e.g., to send an externally triggered HELLO when the local topology of a router changes) and with low minimum intervals. In this case, it is possible though that one control message may get lost, and then OLSRv2 needs to react in order to avoid a long convergence time. (One possibility is to reduce HELLO\_INTERVAL to minimum for a few HELLO messages, then restore it). In a similar vein, the message emission intervals and the information validity times should also be commensurate with the available network capacity: millisecond intervals between TC messages, for example, will consume all available network capacity whereas hourly intervals will be inappropriate even for a static and stable, wired, network (by way of simply new routers arriving in the network, which will not "learn" the network topology before undue long delays).

This adaptation may happen autonomously by a central NMS monitoring and adopting the parameters globally, autonomously by an NMS in each router, monitoring its local topology (and its stability) and adapting parameters locally, or by manual operator intervention.

Given the dynamic and evolutive topology of OLSRv2 networks, a highly desirable property of an NMS is the ability to display and offer visibility of the current network status - for example, to display a graphical map of which routers are currently part of the network. As a proactive protocol, OLSRv2 maintains, in each router, a topology map including all destinations and a subset of the links present in the network (particularly true in a very dense network). A typical feature of an NMS is to inquire as to the topology map of a single router. A slightly less typical feature is to inquire all (or, at least, many) routers in a network, with the purpose of presenting a complete topology map.

In addition to actively monitoring an OLSRv2 network, erroneous or unusual conditions on a router can be flagged to management, e.g., detection of an unusually high number of 1-hop or 2-hop neighborhood changes in a short amount of time, indicating potential problems in that area of the network. [RFC6779] and [RFC7184] facilitate proactively sending "notifications" (also known as traps) from the router towards an NMS. The MIB modules defined in [RFC6779] and [RFC7184] allow for defining both the threshold and the time window of how many times this erroneous condition may occur in the time

window before the notification is sent to the NMS. Once the NMS receives a notification, a network operator may investigate if there is a problem that needs to be resolved, e.g., by changing parameters via the above-described external management.

## 6. Typical Communication Patterns

This section describes typical (management) communications patterns in an operating (post-startup) network. One of the key characteristics of OLSRv2 is that it enables an efficient flooding mechanism (denoted "MPR Flooding"). For some management scenarios, this facilitates better performance by (scope-limited) flooding management requests to MANET routers, rather than sending multiple consecutive unicast messages. While the MIB modules developed for OLSRv2 and NHDP do not support such broadcast operation (due to the nature of SNMP), some of the proprietary management tools mentioned in Section 4 take advantage of this for increased performance.

The below list of such communication patterns is not claimed to be exhaustive, and depending on the deployment, different patterns may be used. However, these patterns have been observed in many deployments of OLSRv2 and its constituent parts, as well as of its predecessor OLSR.

- a) Inquire the state (complete topology graph, link states, and local links - even those not part of topology graph) of a router. An NMS would typically initiate that request. OLSRv2 contains a number of "Information Bases"; basically, tables with rows representing information about local interfaces, other routers in the MANET or the topology of the MANET as perceived by the MANET router. These tables are also reflected as objects in the MIB modules and can be inquired via, e.g., GETBULK for getting multiple rows in a single request. Depending on the number of MANET routers in the network as well as the density of the MANET, tables for one-hop and two-hop routers, as well as routers in further distance, these tables can contain a substantial amount of information, and so inquiring them will return multiple KB or more of data back to the NMS. Given the dynamic topology and often bandwidth-constrained wireless links between MANET routers, this is not a very common operation in many deployments. Moreover, this would typically only be required in debugging situations, as during regular operations, OLSRv2 updates the state automatically and reacts to changes (e.g., by triggering control message generation). This type of operation can benefit from the optimized flooding mechanism, by requesting the state from multiple routers in a region of the MANET in a single request.

- b) Inquire the history/statistics of a router. This request, initiated by an NMS, is typically a small inquiry, such as "how many local link changes have you seen within the past n minutes/seconds/hours". This may be very rare, or it may be several times per minute per router for a while: if the NMS is trying to, e.g., "tune" message intervals and timers, by sending this request to a group of topologically close routers - until, that is, the NMS decides that the topology has stabilized and will ease up. Again, this feature of requesting performance related information is supported by the MIB modules for OLSRv2 and NHDP. While SNMP does not support sending the inquiry via optimized flooding, proprietary protocols take advantage of the optimized flooding mechanism, to reduce the number of unicast requests.
- c) Change the configuration of a router. Other than in the above case in b) (tuning), this really happens only when somehow a router gets a new uplink to an external network, and either a new gateway is added into the network, and/or a new prefix needs to be distributed to the routers. The MIB modules for OLSRv2 and NHDP allow to set all configuration parameters of each router. Optimized flooding may significantly reduce the amount of unicast requests, but are not supported by SNMP.
- d) Visualizing the network as a router sees it. As in a MANET, routers may move and link quality may vary due to link layer characteristics, the network topology may change frequently. In a naive way, this would essentially be the NMS setting up a connection to the router in question, and getting a copy of all routing protocol control messages to construct its own topology graph as would have done that router. Typically, it consists of the router sending a notification to the NMS when a topological change happens, i.e., when either of its information bases change. Even better, it consists of these notifications being "filtered" to only send for those changes that actually impact the usable topology. The latter case is supported by the MIB modules for OLSRv2 and NHDP in the form of notifications (also called "traps") that are sent from the MANET router to the NMS. While these notifications alone do not allow the NMS to visualize the topology, they may suffice to inform the NMS of an unusual change of the topology, and the NMS may inquire the current topology via the process described in a).
- e) Rekeying There is currently no (standard) mechanism for automated key management. One of the reasons for this may be that it is difficult to come up with a single such that will satisfy the requirements for all the different deployments. However, in MANET deployments rekeying is something that can be observed, e.g., as part of the parameter configuration. The particularity of this

is, that it often is a "broadcast configuration operation" where new key material is supposed to be sent to everybody, and not just a single router, e.g., leveraging the optimized flooding mechanism of OLSRV2.

#### 7. This Document does not Constrain how to Manage and Monitor MANETs

As explained in Section 1, this document describes how, what and why some (typical) OLSRV2 networks are managed and monitored as of 2014. As such, the document is reflective, not prescriptive: it does not stipulate requirements for how to manage OLSRV2 networks, nor does it claim to be a complete list of all management patterns or protocols. Other ways of managing an OLSRV2 network are very well imaginable - now, or in future deployments of OLSRV2.

As an example of such a "future management scenario", rather than managing and monitoring routers from a central NMS, a distributed, autonomous management system between multiple routers can be envisioned. In particular, monitoring data that is used to debug network problems and to tweak inefficiencies could be distributed amongst a group of routers in the same network. This would both address problems of single point of failure when using only a single NMS, as well provide additional information about groups of multiple routers, rather than a single router. An example use for such a distributed information flow would be to identify areas of a network wherein, e.g., due to different router densities, message sending interval parameters could be exchanged and optimal values negotiated between routers, so as to obtain locally optimized performance.

While such a management model is highly interesting, it is also at present entirely fictional - at least outside the realm of research. It is included to, both, indicate directions being explored (but not exploited), and to insist that the intent of this document is not to prescribe how MANETs are to be managed, in the presence or in the future, but to describe the (known) state of how MANETs are managed, presently.

#### 8. IANA Considerations

This document has no actions for IANA.

[This section may be removed by the RFC Editor.]

## 9. Security Considerations

This document does not specify a protocol, nor does it provide recommendations for how to manage an OLSRv2 deployment - rather, it reflects how some known deployments of OLSRv2 (and its predecessor, OLSR) have been known to be managed.

With that being said, managing an OLSRv2 network requires the ability to inspect and affect the internal state of the routers therein, by way of mechanisms other than the protocol signals specified for OLSRv2 [RFC7181] and NHDP [RFC6130].

When affecting the state of the OLSRv2 routing process, a management process can be considered as an "outside process" to OLSRv2 and is then expected to respect (at least) the constraints given in Section 5.5, Section 5.6, and in Appendix A of [RFC7181], as well as in Section 5.5 and in Appendix B of [RFC6130]. The example from Section 4.1 of setting excessively short message intervals, leading to channel capacity exhaustion and frame collisions, demonstrates that such an outside process can harm network stability considerably when not carefully protected against unauthorized or unintended usage.

For both inspecting and affecting the state of an OLSRv2 routing process by way of a management interface, great care is necessary to avoid divulging information that should not be exposed, and in opening additional vulnerabilities by way of the management interface. In part, to be able to benefit from securing existing management interfaces, protocols, and implementations, migration to a standardized management framework is desired, and was one of the motivators for standardizing MIB modules for OLSRv2 and NHDP in the first place.

## 10. Acknowledgments

The authors would like to gratefully acknowledge the following people for intense technical discussions, early reviews, and comments on the documents: Alan Cullen (BAE Systems), Christopher Dearlove (BAE Systems), Adrian Farrel (Juniper), David Harrington (Comcast), and Jurgen Schoenwaelder (Jacobs University).

## 11. Informative References

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

- [RFC5148] Clausen, T., Dearlove, C., and B. Adamson, "Jitter Considerations in Mobile Ad Hoc Networks (MANETs)", RFC 5148, February 2008.
- [RFC5444] Clausen, T., Dearlove, C., Dean, J., and C. Adjih, "Generalized MANET Packet/Message Format", RFC 5444, February 2009.
- [RFC5497] Clausen, T. and C. Dearlove, "Representing Multi-Value Time in Mobile Ad Hoc Networks (MANETs)", RFC 5497, March 2009.
- [RFC5706] Harrington, D., "Guidelines for Considering Operations and Management of New Protocols and Protocol Extensions", RFC 5706, November 2009.
- [RFC6130] Clausen, T., Dearlove, C., and J. Dean, "Mobile Ad Hoc Network (MANET) Neighborhood Discovery Protocol (NHDP)", RFC 6130, April 2011.
- [RFC6643] Schoenwaelder, J., "Translation of Structure of Management Information Version 2 (SMIv2) MIB Modules to YANG Modules", RFC 6643, July 2012.
- [RFC6779] Herberg, U., Cole, R., and I. Chakeres, "Definition of Managed Objects for the Neighborhood Discovery Protocol", RFC 6779, May 2012.
- [RFC7181] Clausen, T., Dearlove, C., Jacquet, P., and U. Herberg, "The Optimized Link State Routing Protocol Version 2", RFC 7181, April 2014.
- [RFC7182] Herberg, U., Clausen, T., and C. Dearlove, "Integrity Check Value and Timestamp TLV Definitions for Mobile Ad Hoc Networks (MANETs)", RFC 7182, April 2014.
- [RFC7183] Herberg, U., Dearlove, C., and T. Clausen, "Integrity Protection for the Neighborhood Discovery Protocol (NHDP) and Optimized Link State Routing Protocol Version 2 (OLSRv2)", RFC 7183, April 2014.
- [RFC7184] Herberg, U., Cole, R., and T. Clausen, "Definition of Managed Objects for the Optimized Link State Routing Protocol Version 2", RFC 7184, April 2014.
- [RFC7187] Dearlove, C. and T. Clausen, "Routing Multipoint Relay Optimization for the Optimized Link State Routing Protocol Version 2 (OLSRv2)", RFC 7187, April 2014.



[RFC7188] Dearlove, C. and T. Clausen, "Optimized Link State Routing Protocol Version 2 (OLSRv2) and MANET Neighborhood Discovery Protocol (NHDP) Extension TLVs", RFC 7187, April 2014.

#### Authors' Addresses

Thomas Clausen  
LIX, Ecole Polytechnique  
91128 Palaiseau Cedex,  
France

Phone: +33-6-6058-9349  
Email: T.Clausen@computer.org  
URI: <http://www.thomasclausen.org>

Ulrich Herberg  
Fujitsu Laboratories of America  
1240 E Arques Ave  
Sunnyvale CA 94086,  
US

Phone:  
Email: [ulrich@herberg.name](mailto:ulrich@herberg.name)  
URI: <http://www.herberg.name>



Mobile Ad hoc Networking (MANET)  
Internet-Draft  
Updates: 7188, 7631  
(if approved)  
Intended status: Experimental  
Expires: March 31, 2016

C. Dearlove  
BAE Systems Applied Intelligence  
Laboratories  
T. Clausen  
LIX, Ecole Polytechnique  
September 28, 2015

Multi-Topology Extension for the Optimized Link State Routing Protocol  
version 2 (OLSRv2)  
draft-ietf-manet-olsrv2-multitopology-07

Abstract

This specification describes an extension to the Optimized Link State Routing Protocol version 2 (OLSRv2) to support multiple routing topologies, while retaining interoperability with OLSRv2 routers that do not implement this extension.

This specification updates RFCs 7188 and 7631 by modifying and extending TLV registries and descriptions.

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on March 31, 2016.

Copyright Notice

Copyright (c) 2015 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents

carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

1. Introduction . . . . .	4
1.1. Motivation and Experimentation . . . . .	4
2. Terminology and Notation . . . . .	5
3. Applicability Statement . . . . .	6
4. Protocol Overview and Functioning . . . . .	6
5. Parameters . . . . .	8
6. Information Bases . . . . .	9
6.1. Local Attached Network Set . . . . .	9
6.2. Link Sets . . . . .	9
6.3. 2-Hop Sets . . . . .	9
6.4. Neighbor Set . . . . .	9
6.5. Router Topology Set . . . . .	10
6.6. Routable Address Topology Set . . . . .	10
6.7. Attached Network Set . . . . .	10
6.8. Routing Sets . . . . .	11
7. TLVs . . . . .	11
7.1. Message TLVs . . . . .	11
7.1.1. MPR_TYPES TLV . . . . .	11
7.1.2. MPR_WILLING TLV . . . . .	11
7.2. Address Block TLVs . . . . .	12
7.2.1. LINK_METRIC TLV . . . . .	12
7.2.2. MPR TLV . . . . .	13
7.2.3. GATEWAY TLV . . . . .	13
8. HELLO Messages . . . . .	14
8.1. HELLO Message Generation . . . . .	14
8.2. HELLO Message Processing . . . . .	14
9. TC Messages . . . . .	15
9.1. TC Message Generation . . . . .	15
9.2. TC Message Processing . . . . .	16
10. MPR Calculation . . . . .	16
11. Routing Set Calculation . . . . .	16
12. Management Considerations . . . . .	17
13. IANA Considerations . . . . .	18
13.1. Expert Review: Evaluation Guidelines . . . . .	18
13.2. Message TLV Types . . . . .	18
13.3. Address Block TLV Types . . . . .	19
14. Security Considerations . . . . .	21
15. Acknowledgments . . . . .	21
16. References . . . . .	22
16.1. Normative References . . . . .	22
16.2. Informative References . . . . .	22
Authors' Addresses . . . . .	23

## 1. Introduction

The Optimized Link State Routing Protocol, version 2 [RFC7181] (OLSRv2) is a proactive link state routing protocol designed for use in mobile ad hoc networks (MANETs) [RFC2501]. One of the significant improvements of OLSRv2 over its Experimental precursor [RFC3626] is the ability of OLSRv2 to route over other than minimum hop routes, using a link metric.

A limitation that remains in OLSRv2 is that it uses a single link metric type for all routes. However in some MANETs it would be desirable to be able to route packets using more than one link metric type. This specification describes an extension to OLSRv2 that is designed to permit this, while maintaining maximal interoperability with OLSRv2 routers not implementing this extension.

The purpose of OLSRv2 can be described as to create and maintain a Routing Set, which contains all the necessary information to populate an IP routing table. In a similar way, the role of this extension can be described as to create and maintain multiple Routing Sets, one for each link metric type supported by the router maintaining the sets.

### 1.1. Motivation and Experimentation

Multi-topology routing is a natural extension to a link state routing protocol, as for example to OSPF (see [RFC4915]). However multi-topology routing for OLSRv2 does not yet benefit from extensive operational, or even experimental, experience. This specification is published to facilitate collecting such experience, with the intent that once suitable experimental evidence has been collected, an OLSRv2 Multi-Topology Routing Extension will be proposed for advancement onto Standards Track.

Any experiments using this protocol extension are encouraged. Reports from such experiments planned with pre-specified objectives and scenarios (including link, position and mobility information) are particularly encouraged. Results from such experiments, documenting the following, are of particular importance:

- o Operation in networks that contain both routers implementing this extension, and routers implementing only [RFC7181], in particular are there any unexpected interactions that can break the network?
- o Operation in networks with dynamic topologies, both due to mobility and due to link metric changes for reasons other than mobility.

- o Operation in realistic deployments, and details thereof, including in particular indicating how many concurrent topologies were required.
- o Behavior of routing sets, including measures of successful route establishment.

In addition, reports from experiments covering the following are also of value:

- o Which link metric types were useful, and how the metrics to associate with a given link were established.
- o How packet types were associated with link metric types (whether using DiffServ on an alternative mechanism).
- o Any data link layer issues, and any cross-layer issues, including whether NHDP link quality was used, and how.
- o Transport and higher layer issues observed, if any.
- o Resource requirements observed from running the protocol, including processing, storage, and bandwidth.
- o Network performance, including packet delivery results.
- o Any other implementation issues.

The first bullet in the latter list applies to unextended [RFC7181] as well as this extension, and potentially to other MANET routing protocols. This may also allow experimentation with link metric types that are not compromises to handle multiple traffic types.

## 2. Terminology and Notation

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].

This specification uses the terminology of [RFC5444], [RFC6130] and [RFC7181], which is to be interpreted as described in those specifications.

Additionally, this specification uses the following terminology:

Router - A MANET router that implements [RFC7181].

MT-OLSRv2 - The protocol defined in this specification as an extension to [RFC7181].

This specification introduces the notation `map[A -> B]` to represent an associative mapping. The domain of this mapping (A) is, in this specification, always a set of link metric types that the router supports: either `IFACE_METRIC_TYPES` or `ROUTER_METRIC_TYPES`, as defined in Section 5. The codomain of this mapping (B) is a set of all possible values of an appropriate type, in this specification this type is always one of:

- o boolean (true or false),
- o willingness (a 4 bit unsigned integer from 0 to 15);
- o number of hops (an 8 bit unsigned integer from 0 to 255), or
- o link metric (either a representable link metric value, as described in [RFC7181], or `UNKNOWN_METRIC`).

### 3. Applicability Statement

The protocol described in this specification is applicable to a MANET for which OLSRv2 is otherwise applicable (see [RFC7181], Section 3), but in which multiple topologies are maintained, each characterized by a different choice of link metric type. It is assumed, but outside the scope of this specification, that the network layer is able to choose which topology to use for each packet, for example using the DiffServ Code Point (DSCP) defined in [RFC2474]. This selection of topology **MUST** be consistent, that is each router receiving a packet must make the same choice of link metric type, in order that each packet uses a single topology. This is necessary to avoid the possibility of a packet "looping" in the network.

### 4. Protocol Overview and Functioning

The purpose of this specification is to extend OLSRv2 [RFC7181] so as to enable a router to establish and maintain multiple routing topologies in a MANET, each topology associated with a link metric type. Routers in the MANET may each form part of some or all of these topologies, and each router will maintain a Routing Set for each topology that it forms part of, allowing separate routing of packets for each topology.



MT-OLSRv2 is designed to interoperate with OLSRv2; a MANET can be created containing both routers that implement MT-OLSRv2 (MT-OLSRv2 routers) and routers that do not implement MT-OLSRv2, and may be unaware of its existence (non-MT-OLSRv2 routers). MANETs may also be created that are known to contain only MT-OLSRv2 routers. In both cases, but especially the former, management may be required to ensure that the MANET will function as required, and does not, for example, unnecessarily fragment. (Such issues already arise in an OLSRv2-based MANET using multiple interfaces.)

OLSRv2 is an extension of NHDP [RFC6130]. However the extension in this specification does not modify NHDP, it only modifies Protocol Sets that are specific to OLSRv2, or elements in Protocol Tuples that were added by OLSRv2 and which are not included in nor used by NHDP. In addition it does not use or modify the link quality mechanism in [RFC6130].

Each router implementing this specification selects a set of link metric types for each of its OLSRv2 interfaces. If all routers in the MANET implement MT-OLSRv2, then there are no restrictions within this specification on how these sets of link metrics are selected. (However the issues described in the preceding paragraph still apply.) However in MANETs containing non-MT-OLSRv2 routers, the single link metric used by these non-MT-OLSRv2 routers must be included in the set of link metrics for each OLSRv2 interface of an MT-OLSRv2 router that may be heard on an OLSRv2 interface of a non-MT-OLSRv2 router in the MANET.

Each router then determines an incoming link metric for each link metric type selected for each of its OLSRv2 interfaces. These link metrics are distributed using link metric TLVs contained in all HELLO messages sent on OLSRv2 interfaces, and in all TC messages. Both HELLO and TC messages generated by an MT-OLSRv2 router (other than one using only the single metric type used by non-MT-OLSRv2 routers) include an MPR\_TYPES Message TLV that indicates that this is an MT-OLSRv2 router and which metric types it supports (on the sending OLSRv2 interface for a HELLO message).

In addition to link and neighbor metric values for each link metric type, router MPR (multipoint relay) and MPR selector status, and advertised neighbor status, is maintained per supported neighbor metric type, for each symmetric 1-hop neighbor. Each router may choose a different willingness to be a routing MPR for each link metric type that it supports.

A network using MT-OLSRv2 will usually require greater management than one using unmodified OLSRv2. In particular, the use of multiple metric types across the MANET must be managed, by administrative

configuration or otherwise. As also for other decisions that may be made when using OLSRv2, a bad collective choice of metric type use will make the MANET anywhere from inefficient to non-functional, so care will be needed in selecting supported link metric types across the MANET.

The meanings of link metric types are at the discretion of the MANET operator, they could be used, for example, to represent packets of different types, packets in streams of different rates, or packets with different trust requirements. Note that packets will generally not be delivered to routers that do not support that link metric type, and the MANET, and the packets sent in it, will need to be managed accordingly (especially if containing any non-MT-OLSRv2 routers).

## 5. Parameters

The parameters used in [RFC7181], including from its normative references, are used in this specification with the following changes.

Each OLSRv2 interface will support a number of link metric types, corresponding to Type Extensions of the LINK\_METRIC TLV defined in [RFC7181]. The router parameter LINK\_METRIC\_TYPE, used by routers that do not implement MT-OLSRv2, and used with that definition in this specification, is replaced in routers implementing MT-OLSRv2 by an interface parameter array IFACE\_METRIC\_TYPES and a router parameter array ROUTER\_METRIC\_TYPES. Each element in these arrays is a link metric type (i.e., a type extension used by the LINK\_METRIC TLV [RFC7181]).

The interface parameter array IFACE\_METRIC\_TYPES contains the link metric types supported on that OLSRv2 interface. The router parameter array ROUTER\_METRIC\_TYPES is the union of all of the IFACE\_METRIC\_TYPES. Both arrays MUST be without repetitions.

If in a given deployment there may be any routers that do not implement MT-OLSRv2, then IFACE\_METRIC\_TYPES MUST first include LINK\_METRIC\_TYPE if that OLSRv2 interface may be able to communicate with any routers that do not implement MT-OLSRv2. In that case, ROUTER\_METRIC\_TYPES MUST also first include LINK\_METRIC\_TYPE.

In addition, the router parameter WILL\_ROUTING is extended to an array of values, one each for each link metric type in the router parameter list ROUTER\_METRIC\_TYPES.

## 6. Information Bases

The Information Bases specified in [RFC7181], which extend those specified in [RFC6130], are further extended in this specification. With the exception of the Routing Set, the extensions in this specification are the replacement of single values (boolean, willingness, number of hops, or link metric) from [RFC7181] with elements representing multiple values (associative mappings from a set of metric types to their corresponding values). The following subsections detail these extensions.

Note that, as in [RFC7181], an implementation is free to organize its internal data in any manner it chooses, it needs only to behave as if it were organized as described in [RFC7181] and this specification.

### 6.1. Local Attached Network Set

Each element `AL_dist` becomes a map[ROUTER\_METRIC\_TYPES -> number of hops].

Each element `AL_metric` becomes a map[ROUTER\_METRIC\_TYPES -> link metric].

### 6.2. Link Sets

Each element `L_in_metric` becomes a map[IFACE\_METRIC\_TYPES -> link metric].

Each element `L_out_metric` becomes a map[IFACE\_METRIC\_TYPES -> link metric].

The elements of `L_in_metric` MUST be set following the same rules that apply to the setting of the single element `L_in_metric` in [RFC7181].

### 6.3. 2-Hop Sets

Each element `N2_in_metric` becomes a map[ROUTER\_METRIC\_TYPES -> link metric].

Each element `N2_out_metric` becomes a map[ROUTER\_METRIC\_TYPES -> link metric].

### 6.4. Neighbor Set

Each element `N_in_metric` becomes a map[ROUTER\_METRIC\_TYPES -> link metric].

Each element `N_out_metric` becomes a map[ROUTER\_METRIC\_TYPES -> link

metric].

Each element `N_will_routing` becomes a map[ROUTER\_METRIC\_TYPES -> willingness].

Each element `N_routing_mpr` becomes a map[ROUTER\_METRIC\_TYPES -> boolean].

Each element `N_mpr_selector` becomes a map[ROUTER\_METRIC\_TYPES -> boolean].

Each element `N_advertised` becomes a map[ROUTER\_METRIC\_TYPES -> boolean].

#### 6.5. Router Topology Set

Each element `TR_metric` becomes a map[ROUTER\_METRIC\_TYPES -> link metric].

Note that some values of `TR_metric` may now take the value `UNKNOWN_METRIC`. When used to construct a Routing Set, where just the corresponding link metric value from this mapping is used, Router Topology Tuples whose corresponding value from `TR_metric` is `UNKNOWN_METRIC` are ignored.

#### 6.6. Routable Address Topology Set

Each element `TA_metric` becomes a map[ROUTER\_METRIC\_TYPES -> link metric].

Note that some values of `TA_metric` may now take the value `UNKNOWN_METRIC`. When used to construct a Routing Set, where just the corresponding link metric value from this mapping is used, Routable Address Topology Tuples whose corresponding value from `TA_metric` is `UNKNOWN_METRIC` are ignored.

#### 6.7. Attached Network Set

Each element `AN_dist` becomes a map[ROUTER\_METRIC\_TYPES -> number of hops].

Each element `AN_metric` becomes a map[ROUTER\_METRIC\_TYPES -> link metric].

Note that some values of `AN_metric` may now take the value `UNKNOWN_METRIC`. When used to construct a Routing Set, where just the corresponding link metric value from this mapping is used, Attached Network Tuples whose corresponding value from `AN_metric` is

UNKNOWN\_METRIC are ignored.

## 6.8. Routing Sets

There is a separate Routing Set for each link metric type in ROUTER\_METRIC\_TYPES.

## 7. TLVs

This specification makes the following additions and extensions to the TLVs defined in [RFC7181].

### 7.1. Message TLVs

One new Message TLV is defined in this specification, and one existing Message TLV is extended by this specification.

#### 7.1.1. MPR\_TYPES TLV

The MPR\_TYPES TLV is used in both HELLO messages sent over OLSRv2 interfaces and TC messages. A message MUST NOT contain more than one MPR\_TYPES TLV.

The presence of this TLV in a message is used to indicate that the router supports MT-OLSRv2, in the same way that the presence of the MPR\_WILLING TLV is used to indicate that the router supports OLSRv2, as specified in [RFC7181]. For this reason, the MPR\_TYPES TLV has been defined with the same Type as the MPR\_WILLING TLV, but with Type Extension = 1.

This TLV may take a Value field of any size. Each octet in its Value field will contain a link metric type that is supported, either on any OLSRv2 interface, when included in a TC message, or on the OLSRv2 interface on which an including HELLO message is sent. These octets MAY be in any order, except that if there may be any routers in the MANET not implementing MT-OLSRv2, then the first octet MUST be LINK\_METRIC\_TYPE.

#### 7.1.2. MPR\_WILLING TLV

The MPR\_WILLING TLV, which is used in HELLO messages, is specified in [RFC7181], and extended in this specification as enabled by [RFC7188].

The interpretation of this TLV, specified by [RFC7181], and which uses all of its single octet Value field, is unchanged. That interpretation uses bits 0-3 of its Value field to specify its

willingness to be a flooding TLV, and bits 4-7 of its Value field to be a routing TLV. Those latter bits are, when using this specification, interpreted as its willingness to be a routing TLV using the link metric type LINK\_METRIC\_TYPE.

The extended use of this message TLV, as defined by this specification, defines additional 4 bit sub-fields of the Value field, starting with bits 4-7 of the first octet and continuing with bits 0-3 of the second octet, to represent willingness to be a routing MPR using the link metric types specified in this OLSRv2 interface's IFACE\_METRIC\_TYPES parameter, ordered as reported in the included MPR\_TYPES Message TLV. Note that this means that the link metric type LINK\_METRIC\_TYPE will continue to occupy bits 4-7 of the first octet. (If there is no such TLV included, then the router does not support MT-OLSRv2, and only the first octet of the Value field will be used.)

If the number of link metric types in this OLSRv2 interface's IFACE\_METRIC\_TYPES parameter is even, then there will be an unused 4 bit sub-field in bits 4-7 of the last octet of a full sized Value field. These bits will not be used, they SHOULD all be cleared ('0').

If the Value field in an MPR\_WILLING TLV is shorter than its full length, then, as specified in [RFC7188], missing Value octets, i.e., missing willingness values, are considered as zero, i.e., as WILL\_NEVER. This is the correct behavior. (In particular it means that an OLSRv2 router that is not implementing MT-OLSRv2 will not act as a routing MPR for any link metric that it does not recognize.)

## 7.2. Address Block TLVs

New Type Extensions are defined for the LINK\_METRIC TLV defined in [RFC7181], and the Value fields of the MPR TLV and the GATEWAY TLV, both defined in [RFC7181], are extended, as enabled by [RFC7188].

### 7.2.1. LINK\_METRIC TLV

The LINK\_METRIC TLV is used in HELLO messages and TC messages. This TLV is unchanged from the definition in [RFC7181].

Only a single Type Extension was specified by [RFC7181] (link metric type) 0 as defined by administrative action. This specification extends this range to 0-7. This specification will work with any combination of Type Extensions both within and without that range (assuming that the latter are defined as specified in [RFC7181]).

### 7.2.2. MPR TLV

The MPR TLV is used in HELLO messages, and indicates that an address with which it is associated is of a symmetric 1-hop neighbor that has been selected as an MPR.

The Value field of this address block TLV is, in [RFC7181], defined to be one octet long, with the values 1, 2 and 3 defined. [RFC7188] redefines this Value field to be a bitfield where bit 7 (the lsb) denotes flooding status, bit 6 denotes routing MPR status, and bits 5-0 are unallocated (respecting the semantics of the bits/values 1, 2 and 3 from [RFC7181]).

This specification, as enabled by [RFC7188], extends the MPR TLV to have a variable-length Value field. For interoperability with a router not implementing MT-OLSRv2, the two least significant bits of the first octet in the Value field of this TLV MUST be the TLV Value of the MPR TLV, generated according to [RFC7181].

Subsequent bits (in increasing significance within an octet, then continuing with the least significant bit in the next octet, if required) in the TLV Value field indicate which link metric types, for which the corresponding address is selected as a routing MPR, link metric types (including the first) being indicated in, and used in the same order as, the Value field of an MPR\_TYPES Message TLV, excluding the link metric type LINK\_METRIC\_TYPE, which already occupies the second bit.

### 7.2.3. GATEWAY TLV

The GATEWAY TLV is used in TC messages to indicate that a network address is of an attached network.

The Value field of this address block TLV is, in [RFC7181] defined to be one octet long, containing the number of hops to that attached network.

This specification, as enabled by [RFC7181], allows the extension the GATEWAY TLV to have a variable-length Value field when the number of hops to each attached network is different for different link metric types. For interoperability with a router not implementing MT-OLSRv2, the first octet in the Value field of this TLV MUST be the TLV Value of the GATEWAY TLV generated according to [RFC7181].

Any subsequent octets in the TLV Value field indicate the number of hops to the attached network for each other link metric type, link metric types (including the first) being indicated in the Value field of an MPR\_TYPES Message TLV.

Type	Value
GATEWAY	Number of hops to attached network for each link metric type.

Table 1: GATEWAY TLV definition

## 8. HELLO Messages

The following changes are made to the generation and processing of HELLO messages compared to that described in [RFC7181] by routers that implement MT-OLSRv2.

### 8.1. HELLO Message Generation

A generated HELLO message to be sent on an OLSRv2 interface (whose IFACE\_METRIC\_TYPES parameter will be that used) is extended by:

- o Adding an MPR\_TYPES Message TLV. The Value octets will be the link metric types in IFACE\_METRIC\_TYPES. This TLV MAY be omitted if the only link metric type included would be LINK\_METRIC\_TYPE.
- o Extending the MPR\_WILLING Message TLV Value field to report the willingness values from the WILL\_ROUTING parameter list that correspond to the link metric types in IFACE\_METRIC\_LIST, in the same order as reported in the MPR\_TYPES TLV, each value (also including one representing WILL\_FLOODING) occupying 4 bits.
- o Including LINK\_METRIC Address Block TLVs that report all values in L\_in\_metric, L\_out\_metric, N\_in\_metric and N\_out\_metric elements that are not equal to UNKNOWN\_METRIC, with the TLV Type Extension being the link metric type, and otherwise following the rules for such inclusions specified in [RFC7181].
- o Including MPR Address Block TLVs such that for each link metric type in IFACE\_METRIC\_TYPES, and for the choice of flooding MPRs, the indicated addresses MUST be of the MPRs in an MPR set as specified for a single link metric type in [RFC7181].

### 8.2. HELLO Message Processing

On receipt of a HELLO message on an OLSRv2 interface, a router implementing MT-OLSRv2 MUST, in addition to the processing described in [RFC7181]:



1. If in this deployment there may be any routers that do not implement MT-OLSRv2, the HELLO message contains an MPR\_TYPES Message TLV, and the first link metric type that it reports is not LINK\_METRIC\_TYPE, then the HELLO message MUST be silently discarded.
2. Determine the list of link metric types supported by the sending router on its corresponding OLSRv2 interface, either from an MPR\_TYPES Message TLV or, if not present, the single link metric type LINK\_METRIC\_TYPE.
3. For those link metric types supported by both routers, set the appropriate L\_out\_metric, N\_in\_metric, N\_out\_metric, N\_will\_routing, N\_mpr\_selector, N\_advertised, N2\_in\_metric and N2\_out\_metric values as described for single such elements in [RFC7181].
4. For any other metric types supported by the receiving router only (i.e. in IFACE\_METRIC for the receiving OLSRv2 interface), set the elements listed in the previous point to their default values, i.e., UNKNOWN\_METRIC, WILL\_NEVER (not WILL\_DEFAULT), or false.

## 9. TC Messages

The following changes are made to the generation and processing of TC messages compared to that described in [RFC7181] by routers that implement MT-OLSRv2.

### 9.1. TC Message Generation

A generated TC message is extended by:

- o Adding an MPR\_TYPES TLV. The value octets will be the link metric types in ROUTER\_METRIC\_TYPES. This MAY be omitted if the only link metric type included would be LINK\_METRIC\_TYPE.
- o Including LINK\_METRIC TLVs that report all values of N\_out\_metric that are not equal to UNKNOWN\_METRIC, with the TLV Type Extension being the link metric type, and otherwise following the rules for such inclusions specified in [RFC7181].
- o When not all the same, including a number of hops per reported (in an MPR\_TYPES Message TLV) link metric type in the Value field of each GATEWAY TLV included, in the same order as reported in the MPR\_TYPES TLV.

## 9.2. TC Message Processing

On receipt of a TC message, a router implementing this extension MUST, in addition to the processing specified in [RFC7181]:

- o If in this deployment there may be any routers that do not implement MT-OLSRv2, the TC message contains an MPR\_TYPES Message TLV, and the first link metric type that it reports is not LINK\_METRIC\_TYPE, then the TC message MUST be silently discarded.
- o Set the appropriate TR\_metric, TA\_metric, AN\_dist and AN\_metric elements using the rules for setting the single elements of those types specified in [RFC7181].
- o For any other metric types supported by the receiving router that do not have an advertised outgoing neighbor metric of that type, set the corresponding elements of TR\_metric, TA\_metric and AN\_metric to UNKNOWN\_METRIC. (The corresponding element of AN\_dist may be set to any value.)

## 10. MPR Calculation

Routing MPRs are calculated for each link metric type in ROUTER\_METRIC\_TYPES. Links to symmetric 1-hop neighbors via OLSRv2 interfaces that do not support that link metric type are not considered. The determined status (routing MPR or not routing MPR) for each link metric type is recorded in the relevant element of N\_routing\_mpr.

Each router may make its own decision as to whether or not to use a link metric, or link metrics, for flooding MPR calculation, and if so which and how. This decision MUST be made in a manner that ensures that flooded messages will reach the same symmetric 2-hop neighbors as would be the case for a router not supporting MT-OLSRv2.

Note that it is possible that a 2-Hop Tuple in the Information Base for a given OLSRv2 interface does not support any of the link metric types that are in the router's corresponding IFACE\_METRIC\_TYPES, but nevertheless that 2-Hop Tuple MUST be considered when determining flooding MPRs.

## 11. Routing Set Calculation

A Routing Set is calculated for each link metric type in ROUTER\_METRIC\_TYPES. The calculation may be as for [RFC7181], except that where an element is now represented by a map, the value from the

map for the selected link metric type is used. Where this is a link metric of value UNKNOWN\_METRIC, that protocol Tuple is ignored for the calculation.

## 12. Management Considerations

MT-OLSRv2 may require greater management than unextended OLSRv2. In particular a MANET using MT-OLSRv2 requires the following management considerations:

- o Deciding which link metric, and hence which Routing Set to use, for received packets, hence how to use the Routing Sets to configure the network layer (IP). All routers MUST make the same decision for the same packet. An obvious approach is to map each DiffServ Code Point (DSCP) [RFC2474] to a single link metric. (This may be a many to one mapping.)
- o Selecting which link metrics to support on each OLSRv2 interface and implementing that decision. (Different interfaces may have different physical and data link layer properties, and this may inform the selection of link metrics to support, and their values.) If the MANET may contain non-MT-OLSRv2 routers, which is also subject to management, then the rules for link metric assignment to OLSRv2 interfaces in this specification for that case MUST be followed.
- o Ensuring that the MANET is sufficiently connected, by ensuring that, for example, sufficiently many routers implement each metric type required (this being easier in, for example, a denser network). Note that if there is any possibility that there are any routers not implementing MT-OLSRv2, then the MANET will be connected, to the maximum extent possible, using the link metric type LINK\_METRIC\_TYPE, but this will only serve to deliver packets that use that link metric type.
- o Non-MT-OLSRv2 routers SHOULD be managed so as not produce packets that will be routed by a topology that they are not part of. However if they were to do so then such packets will be routed until either they reach their destination, or they reach an MT-OLSRv2 router. In the latter case the packet will then either be dropped (if that MT-OLSRv2 router is not part of that topology, or is not aware of the destination within that topology) or will be routed by that topology to the destination. Such a packet will not loop.
- o If a packet is created for a destination that is not part of the corresponding topology then it may or may not be delivered (if the

originating router is a non-MT-OLSRv2 router) or will not be transmitted (if the originating router is an MT-OLSRv2 router). Routers SHOULD be managed so that this does not occur.

### 13. IANA Considerations

This specification adds one new Message TLV, allocated as a new Type Extension to an existing Message TLV, using a new name. It also modifies the Value field of an existing Message TLV, and of an existing Address Block TLV. Finally, this specification makes additional allocations from the LINK\_METRIC Address Block TLV Type registry.

#### 13.1. Expert Review: Evaluation Guidelines

For the registry where an Expert Review is required, the designated expert SHOULD take the same general recommendations into consideration as are specified by [RFC5444] and [RFC7631].

#### 13.2. Message TLV Types

This specification modifies the Message TLV Type 7, replacing Table 4 of [RFC7631] by Table 2, changing the description of the Type Extension MPR\_WILLING and adding the Type Extension TLV\_TYPES. Each of these TLVs MUST NOT be included more than once in a Message TLV Block.

Type Extension	Name	Description	Reference
0	MPR_WILLING	First (most significant) half octet of Value field specifies the originating router's willingness to act as a flooding MPR; subsequent half octets specify the originating router's willingness to act as a routing MPR, either for the link metric types reported in an MPR_TYPES TLV (in the same order), or (if no MPR_TYPES TLV is present) for the single administratively agreed link metric type	[RFC7181] [RFC7631] This specification
1	MPR_TYPES	The link metric types supported on this OLSRv2 interface of this router (one octet each).	This specification
2-223 224-255		Unassigned Reserved for Experimental Use	[RFC7181]

Table 2: Type 7 Message TLV Type Extensions

## 13.3. Address Block TLV Types

Table 7 of [RFC7188] is replaced by Table 3.

Bit	Value	Name	Description
First octet bit 7	First octet 0x01	Flooding	If set then the neighbor with that network address has been selected as flooding MPR
From first octet bit 6	From first octet 0x02	Routing	If set then the neighbor with that network address has been selected as routing MPR, either for the link metric types reported in an MPR_TYPES TLV (in the same order), or (if no MPR_TYPES TLV is present) then (first octet bit 6, value 0x02) for the single administratively agreed link metric type

Table 3: MPR TLV Bit Values

Table 14 of [RFC7631] is replaced by Table 4. The only changes are to the Description and the References for the GATEWAY TLV.

Type Extension	Name	Description	References
0	GATEWAY	Specifies that a given network address is reached via a gateway on the originating router. The number of hops is indicated by the Value field, one octet per link metric type reported in an MPR_TYPES Message TLV (in the same order) or (if no MPR_TYPES Message TLV is present) using a single octet	[RFC7181] This specification
1-223		Unassigned	
224-255		Reserved for Experimental Use	[RFC7631]

Table 4: Type 10 Address Block TLV Type Extensions

Table 13 of [RFC7181] is replaced by Table 5. The only change is to allocate 8 Type Extensions as assigned by administrative action, in order to support administratively determined multi-topologies.

Name	Type	Type Extension	Description	Allocation Policy
LINK_METRIC	7	0-7	Link metric meaning assigned by administrative action.	
LINK_METRIC	7	8-223	Unassigned.	Expert Review
LINK_METRIC	7	224-255	Unassigned.	Experimental Use

Table 5: Address Block TLV Type assignment: LINK\_METRIC

#### 14. Security Considerations

This extension to OLSRv2 allows a router to support more than one link metric type for each link advertised in HELLO and TC messages, and for routers to support different sets of types. Link metric values of additional types are reported by the inclusion of additional TLVs in the messages sent by a router, which will report known values of all supported types.

HELLO and TC message processing is then extended simply to record, for each supported type, all of the received link metric values for each link. Protocol internal processing (specifically MPR set and shortest path calculations) then operate as specified in [RFC7181] for each link metric type that the router supports.

Consequently the security considerations, including the security architecture and the mandatory security mechanisms, from [RFC7181] are directly applicable to MT-OLSRv2.

Furthermore, this extension does not introduce any additional vulnerabilities over those of [RFC7181], because each link metric type is used independently, and each one could have been the single link metric type supported by an implementation of [RFC7181] receiving the same information, as received information of an unsupported type is ignored by all routers.

#### 15. Acknowledgments

The authors would like to thank (in alphabetical order): Juliusz Chroboczek (University of Paris Diderot), Alan Cullen (BAE Systems),

Susan Hares (Huawei) and Henning Rogge (FGAN) for discussions and suggestions.

## 16. References

### 16.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC5444] Clausen, T., Dearlove, C., Dean, J., and C. Adjih, "Generalized MANET Packet/Message Format", RFC 5444, February 2009.
- [RFC6130] Clausen, T., Dean, J., and C. Dearlove, "Mobile Ad Hoc Network (MANET) Neighborhood Discovery Protocol (NHDP)", RFC 6130, April 2011.
- [RFC7181] Clausen, T., Dearlove, C., Jacquet, P., and U. Herberg, "The Optimized Link State Routing Protocol version 2", RFC 7181, April 2014.
- [RFC7188] Dearlove, C. and T. Clausen, "Optimized Link State Routing Protocol version 2 (OLSRv2) and MANET Neighborhood Discovery Protocol (NHDP) Extension TLVs", RFC 7188, April 2014.
- [RFC7631] Dearlove, C. and T. Clausen, "TLV Naming in the MANET Generalized Packet/Message Format", RFC 7631, January 2015.

### 16.2. Informative References

- [RFC2474] Nichols, K., Blake, S., Baker, F., and D. Black, "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers", RFC 2474, December 1998.
- [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.
- [RFC4915] Psenak, P., Mirtorabi, S., Roy, A., Nguyen, L., and P. Pillay-Esnault, "Multi-Topology (MT) Routing in OSPF",



RFC 4915, June 2007.

Authors' Addresses

Christopher Dearlove  
BAE Systems Applied Intelligence Laboratories  
West Hanningfield Road  
Great Baddow, Chelmsford  
United Kingdom

Phone: +44 1245 242194  
Email: [chris.dearlove@baesystems.com](mailto:chris.dearlove@baesystems.com)  
URI: <http://www.baesystems.com/>

Thomas Heide Clausen  
LIX, Ecole Polytechnique

Phone: +33 6 6058 9349  
Email: [T.Clausen@computer.org](mailto:T.Clausen@computer.org)  
URI: <http://www.ThomasClausen.org/>



Internet Engineering Task Force  
Internet-Draft  
Intended status: Informational  
Expires: November 6, 2014

B. Snyder  
iDirect Technologies  
N. Akiya  
Cisco Systems  
May 5, 2014

BFD Proxy for Connections over Monitored Links  
draft-snyder-bfd-proxy-connections-monitored-links-00

Abstract

This document describes a Bidirectional Forwarding Detection (BFD) proxy mechanism to allow intermediate networking equipment (ex: Satellite HUB/Modem) to intercept BFD packets and to generate BFD packets to relay the health of connection monitored links.

Note that this is an informational document that does not propose any changes to the BFD protocol.

Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on November 6, 2014.

Copyright Notice

Copyright (c) 2014 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

1. Introduction . . . . .	2
1.1. Terminology . . . . .	2
1.2. Background . . . . .	3
2. Overview . . . . .	4
3. BFD Proxy Placement . . . . .	5
4. BFD Proxy Procedures . . . . .	5
4.1. BFD Control Packet Interception . . . . .	5
4.2. OAM Object . . . . .	6
4.3. BFD Proxying . . . . .	6
4.4. Outroute Considerations . . . . .	8
4.5. Inroute Considerations . . . . .	8
5. Possible Integration Improvements . . . . .	9
6. Security Considerations . . . . .	9
7. IANA Considerations . . . . .	10
8. Acknowledgements . . . . .	10
9. References . . . . .	10
9.1. Normative References . . . . .	10
9.2. Informative References . . . . .	10
Authors' Addresses . . . . .	10

## 1. Introduction

### 1.1. Terminology

The following acronyms/terminologies are used in this document:

- o BFD - Bidirectional Forwarding Detection
- o DLEP - Dynamic Link Exchange Protocol
- o L2 - Layer 2
- o L3 - Layer 3
- o Outroute - The broadcast link from hub to modem(s) in a satellite network.

- o Downstream - Synonymous to Outroute.
- o OTA - Over the Air
- o Inroute - The unicast uplink that a modem transmits to the hub side on in a satellite network.
- o Upstream - Synonymous to Inroute.

## 1.2. Background

Bidirectional Forwarding Detection (BFD) is an application agnostic and link type independent keep alive protocol which has widely been implemented and deployed. The BFD protocol can be configured with a fast interval to provide rapid failure detection or configured with a slower interval to provide slower failure detection. The faster the interval, the more BFD packets are transmitted and received, consuming more system and network resources.

Some links have connection monitoring functionality of its own, and some of these connection monitored links have constraints (ex: limited or expensive bandwidth). Applications over such links often still desire rapid failure detection through exchanging keep-alive packets (ex: BFD). However, the consequence of such can significantly degenerate the value of the links. For example, running BFD over a link with limited bandwidth can result in a significant portion of the bandwidth being consumed by BFD packets.

One example of such scenario is:

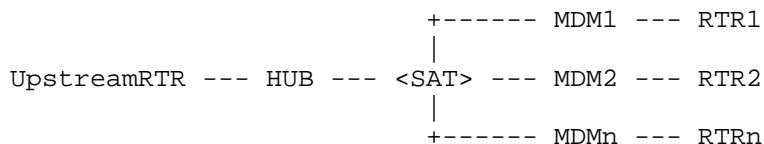


Figure 1: Star Satellite Network

The HUB components consist of a protocol stack which processes and inspects all outbound packets in order to optimize traffic for a high delay low bandwidth environments. (Ex: TCP proxy, compression, encryption). This stack also contains a L2 switch to demultiplex outroute traffic towards the proper modem via a MAC learning switch. In this component is also station keeping algorithms and QOS schedulers.

The MDM components have the same protocol stack (without the demultiplexing required) to optimize the traffic flow for the TDMA

inroutes. The interfaces of a modem are 1 RF interface and 1 to 8 ethernet interfaces.

When routers connected to HUB and MDMs run BFD to monitor the L3 reachability through the Satellite network, expensive Satellite bandwidth gets consumed with large number of BFD packets traversing over it.

Dynamic Link Exchange Protocol (DLEP), [I-D.ietf-manet-dlep], tackles this problem by introducing a protocol that can communicate the state of monitored links to routing devices. DLEP also maintains and communicates an extensive set of information (ex: link quality). A wide range of DLEP responsibilities result in a large effort for vendors to develop this protocol. DLEP, in addition, will require further effort to get integrated into various applications (ex: IGP) for the information to be beneficial.

BFD, on the other hand, has widely been implemented and deployed. If applications, already capable of speaking BFD, only require keeping track of a connection state over monitored links, and not any other information provided by DLEP, then the BFD proxy, described in this document, can be implemented on intermediate networking equipment to allow:

- o Connected network equipment (i.e. routers) to continue using BFD for continuity check.
- o BFD packets to consume minimal bandwidth on monitored links.

## 2. Overview

This informational document describes a BFD proxy mechanism that allows for connection monitoring intermediate networking equipment (ex: Satellite HUB/Modem) to use BFD packets in order to communicate the state of monitored links whilst significantly reducing the network bandwidth consumed by BFD packets.

The BFD proxy is a link state aware module that resides on the intermediate networking equipment, and intercepts all BFD packets coming in from the connected network equipment.

The first task of the BFD proxy is to transmit BFD control packets to the connected network equipment in order to communicate the state of monitored links, based on its knowledge of link state. The BFD proxy can inject BFD STATE change events towards the connected network equipment. When the device under monitoring is present, the BFD proxy can inject BFD packets with BFD\_UP state. When the device

under monitoring has left the network, the BFD proxy can inject BFD packets with BFD\_DOWN state.

The second task, to reduce network bandwidth is handled both at the BFD level (by proxying) and at the L3 level. By proxying the BFD control packets one can keep all the BFD overhead off the monitored (and often expensive) bandwidth links. The use of BFD also allows for network designers to configure L3 keep-alive/HELLO timers to be increased thereby reducing OTA bandwidth usage of un-proxied data flows. With BFD monitoring and alerting, L3 convergence is bound by a combination of link state awareness and IGP Hello time (in either direction). The monitored link's state (ex: satellite modem) can be immediately propagated when transitioning between in and out of network. Additionally, configurations and protocols will be discussed that have been determined to be optimal to this use case.

This document will also suggest multiple integration improvements that all interested parties (routing vendors and modem vendors) could implement to further optimize convergence time and bandwidth usage. The network configuration is that of a star design, where thousands of CE routers each behind a satellite remote will attach to one hub upstream router via desired L3 protocols. Whilst, many networks do utilize mobility and roaming, they are always aware of whom they are connecting too (either one or more possible HUBs, but only one at a time). As the goal is simply to assist the routers in understanding radio link state to optimize routing convergence, BFD is the optimal way of meeting this need.

### 3. BFD Proxy Placement

The BFD proxy module MUST be placed on a system such that it meets following two criteria:

1. The BFD proxy module can access the state of monitored links and neighbors reachable through it.
2. The BFD proxy module can access all single-hop BFD control packets coming in from the connected network equipment.

### 4. BFD Proxy Procedures

#### 4.1. BFD Control Packet Interception

The BFD proxy module MUST intercept all single-hop BFD control packets (referred to as BFD packets from hereon) coming in from the connected network equipment. Criteria to identify single-hop BFD control packets are:

1. IP/UDP Packet
2. IP TTL 255 ([RFC5881] and for [RFC5082])
3. UDP destination port 3784 ([RFC5881])

#### 4.2. OAM Object

The BFD proxy module SHOULD maintain an OAM object per neighbor reachable through monitored links. This OAM object is to have the state of the neighbor (i.e. available or not available), stores local BFD discriminator value and caches the latest BFD packet intercepted. When the BFD proxy module intercepts a BFD packet, destination MAC address is used to locate the OAM object. If corresponding OAM object is not found, then perform local checks to see if one should get created. If the check passes, create the OAM object. Otherwise do not create one.

#### 4.3. BFD Proxying

Upon intercepting a BFD packet and locating a corresponding OAM object, the BFD proxy module is to follow procedures described in this sub-section.

1. If there is no OAM object, no further action is taken.
2. If the state of the neighbor in the OAM object is "not-available", then no further action is taken.
3. If the State field of intercepted BFD control packet is:
  - \* ADMIN\_DOWN: Forward the intercepted packet OTA to alert the real destination.
  - \* DOWN: Create a BFD packet and copy the contents from intercepted packet, with the following modifications:
    - + Swap source and destination MAC addresses.
    - + Swap source and destination IP addresses.
    - + Set "my discriminator" field.
    - + Clear "your discriminator" field.

Send constructed BFD packet to the connected network equipment.



- \* INIT: If "your discriminator" does not match expected value, then no further action is taken. Otherwise, create a BFD packet and copy the contents from the intercepted packet, with the following modifications:
  - + Swap source and destination MAC addresses.
  - + Swap source and destination IP addresses.
  - + Swap "my discriminator" and "your discriminator" fields.
  - + Set "State" field to UP.Send constructed BFD packet to the connected network equipment.
- \* UP: If "your discriminator" does not match the expected value, then no further action is taken. Otherwise, create a BFD packet and copy the contents from the intercepted packet, with following modifications:
  - + Swap source and destination MAC addresses.
  - + Swap source and destination IP addresses.
  - + Swap "my discriminator" and "your discriminator" fields.Send constructed BFD packet to the connected network equipment.

In addition, following procedures MAY be applied:

- o When a BFD control packet is sent to the connected network equipment, the UDP checksum is set to 0 to avoid the recalculation.
- o When the state of the neighbor in the OAM object changes from "available" to "not-available", then the BFD proxy module SHOULD send unsolicited BFD control packet with state field as DOWN to the connected network equipment. If this is not done, then absence of a "reply" BFD control packet from the BFD proxy will cause the sending router to timeout the connection after 3 drops (or whatever the multiplier is set too).
- o Once the BFD proxy is intercepting BFD control packets and is in UP state, Poll sequence MAY be initiated to increase values in Minimum TX Interval and Minimum RX Interval fields to reduce the

number of BFD control packets on the link connecting the network equipment and the intermediate network equipment.

- o Since on a Satellite Star Network configuration the outroute and inroute have different bandwidth considerations, there are unique integration concerns which are described below

#### 4.4. Outroute Considerations

In a star satellite network, the outroute is a broadcast channel which all remotes receive. While there need not be any restrictions on L3 routing protocols, it does naturally follow that an IGP is a good choice. Specifically, one which allows for asynchronous timers.

Terrestrial convergence timing with BFD (sub second) is in the most common error cases (rainfade, mobility switching) not a realistic goal as the RF algorithms that determine out of network will take on the order of seconds (15 in this specific case). Therefore should a modem leave the network for any reason, the minimum convergence time at the hub side is 15 seconds plus BFD timing to recognize the link loss. Hence, the goal being to minimize bandwidth overhead to make this as short as possible above layer 1 timing. A further consideration is convergence timing when a modem comes back into network. If the L3 timers are made too high, then it can take too long to recognize a positive network state. The outroute being a broadcast medium, can work well within these parameters if for instance the outroute L3 hello timing was every 5 seconds. That's only 1 multicast hello packet to cover the entire network and will bound the convergence time to within 5 seconds.

#### 4.5. Inroute Considerations

On the inroute, network bandwidth is much harder to come by, because the aggregate throughput of all inroutes is shared amongst all modems (potentially numbering in the tens of thousands), and is very expensive. Also, it is unicast to the hub side only. Therefore any decisions made here on timing and data transmissions must scale to the tens of thousands in design principles. This fact is the catalyst for preferring asynchronous timers. Ideally, one can rely on the hello packet of a multicast outroute to kick off convergence, and the hello timing of the inroute can be tuned down as much as possible, to optimize inroute usage. This is possible with EIGRP and IS-IS protocols. Unfortunately, BGP and OSPF require synchronized timers, which means it is impossible to weigh equally the convergence timing while protecting inroute bandwidth.

Additionally, further integration simplicity can optionally be achieved if desired. Notice the timing of 15 seconds to recognize

modem link state is also 3 (a common multiplier setting) times the 5 seconds (common hello message timing). Therefore, it is possible, if one is only interested in monitoring link state, to not utilize BFD on all the remote LANs, as 15 seconds is enough time for the L3 messaging to alert the router to a network issue and just about the same time that the hub side will notice. This is useful to simplify operational complexity and management of the thousands or tens of thousands of installed networks. If one would like BFD to monitor modem LAN state as well, then it would be required regardless.

## 5. Possible Integration Improvements

The following improvements could help with overhead and convergence timing in all monitored network environments. They can require changes on routing or modem equipment to further optimize these types of networks:

- o BFD timer - Allowing for connected network equipment to configure a high BFD interval value. One of BFD's missions is to support sub second failure notification. This document puts forth a useful situation in which BFD is a great help, but does not require such strict timing. In fact, it would scale better with much looser restrictions on timer configuration.
- o BFD demand mode implementation - If vendors had implemented demand mode, it would be possible for the BFD proxy to send D bit to the connected network to significantly minimize BFD packets traversing over local link connected to the network equipment, without tweaking Minimum TX Interval and Minimum RX Interval values. This would reduce processing of BFD packets by the BFD proxy module even further.
- o BFD protocol - Adding into the core protocol the notion of a proxier could assist with support of authentication in this use case, if desired.

## 6. Security Considerations

The proxying by the BFD proxy module will require additional considerations (i.e. knowing authentication types/keys of each neighbor) to handle BFD packets with BFD authentication data (described in Section 6.7 of [RFC5880]. This document only describes procedures to handle BFD packets without BFD authentication data. However, because the mechanism is only applicable to single-hop BFD ([RFC5881]) and GTSM (i.e. check for TTL=255) already provides fairly strong security, lack of BFD authentication support is not considered threatening.

## 7. IANA Considerations

This document does not define any code points.

## 8. Acknowledgements

Authors would like to thank Adrian Farrel for providing a suggestion to generalize the solution to all monitored links.

## 9. References

### 9.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC5880] Katz, D. and D. Ward, "Bidirectional Forwarding Detection (BFD)", RFC 5880, June 2010.
- [RFC5881] Katz, D. and D. Ward, "Bidirectional Forwarding Detection (BFD) for IPv4 and IPv6 (Single Hop)", RFC 5881, June 2010.

### 9.2. Informative References

- [I-D.ietf-manet-dlep] Ratliff, S., Cisco, C., Harrison, G., Jury, S., and D. Satterwhite, "Dynamic Link Exchange Protocol (DLEP)", draft-ietf-manet-dlep-05 (work in progress), February 2014.
- [RFC5082] Gill, V., Heasley, J., Meyer, D., Savola, P., and C. Pignataro, "The Generalized TTL Security Mechanism (GTSM)", RFC 5082, October 2007.

## Authors' Addresses

Brian Snyder  
iDirect Technologies  
  
Email: bsnyder@idirect.net

Nobo Akiya  
Cisco Systems  
  
Email: nobo@cisco.com

Mobile Ad hoc Networks Working Group  
Internet-Draft  
Expires: January 1, 2015

R. Taylor  
J. Dowdell  
Airbus Defence & Space  
June 30, 2014

Layer-3 Extensions to DLEP  
draft-taylor-manet-l3-dlep-00

Abstract

There exists a class of devices where DLEP functionality is desired but as the devices operate at layer-3, supporting the core DLEP specification with its requirement that modems operate as transparent layer-2 bridges is inappropriate.

This document introduces two optional extensions to the core DLEP specification. Each extension may be used in isolation without breaking backwards compatibility.

By relaxing the requirement that all DLEP destinations be identified by MAC address, and the addition of a new extension TLV describing available destination routes, the functionality of DLEP can be implemented by layer-3 forwarding devices.

Note:

- o This document is intended as an extension to the core DLEP specification, and readers are expected to be fully conversant with the operation of core DLEP.
- o The DLEP specification is still in draft, and this document serves a secondary purpose to explore and validate the extension mechanisms detailed in DLEP. This document will therefore require further update as the core DLEP draft progresses towards standards track.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on January 1, 2015.

#### Copyright Notice

Copyright (c) 2014 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

#### Table of Contents

1. Introduction . . . . .	2
1.1. Requirements . . . . .	3
2. Non MAC-Address Destination Identifiers . . . . .	3
2.1. Non MAC TLV . . . . .	4
2.2. Destination Identifiers In Existing Data Items . . . . .	4
3. External Route Advertisement . . . . .	5
3.1. Route TLV . . . . .	6
4. Security Considerations . . . . .	7
5. IANA Considerations . . . . .	7
5.1. Registration . . . . .	7
6. Normative References . . . . .	7
Authors' Addresses . . . . .	7

#### 1. Introduction

The Dynamic Link Exchange Protocol [DLEP] describes a protocol for modems to advertise the status of wireless links between reachable destinations to attached routers. The core specification of the protocol assumes that the participating modems operate as a transparent bridge, and that destinations are identified by MAC address.

There exists some classes of devices where this reachability model is too restrictive but the benefits of the DLEP protocol are desired,

such as destination availability sensing, credit windowing, and/or link metrics. Examples of such devices include modems with some advanced, possibly proprietary, routing capability implemented within the device; or modems with cryptographic capability, where the DLEP functionality is required on the clear-text side but the destinations are actually addressed on the cipher-text side via some tunnelling technology.

To enable such devices to take advantage of the DLEP protocol this specification adds two extensions to the DLEP protocol: Non MAC-address destination identifiers and external route advertisement. Both extensions are marked as OPTIONAL in this document, meaning that either one, or the other, or both may be implemented by a conforming router or modem.

A criticism of this extension could be that such layer-3 devices should instead be running one or more instances of a layer-3 routing protocol to exchange routes; in that case the core functionality of DLEP would have to be implemented in a separate, but very similar, protocol. This document attempts to avoid such a cloning of the DLEP core functionality by extending the DLEP specification with optional mechanisms to allow such layer-3 devices to operate.

### 1.1. Requirements

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14, RFC 2119 [RFC2119].

## 2. Non MAC-Address Destination Identifiers

In the core DLEP specification it is stated that 'The MAC address TLV MUST appear in all destination-oriented signals'. The extension described here replaces the semantics of the MAC address in the TLV with a unique Destination Identifier.

The requirements of a destination identifier is that each destination MUST be unique within the DLEP session and not reused during the lifetime of a session. Multicast or group destinations are not supported by this extension; such functionality should be implemented by using layer-3 multicast addresses.

During DLEP Peer Initialization, a modem that wishes to advertise that it implements this extension MUST include the new Non MAC TLV that indicates that all destinations advertised by the device are not MAC addresses and therefore not addressable at layer-2. Each

destination identifier MUST have the length of the number of octets specified in the Non MAC TLV presented during session initialization

By supporting this extension, the modem indicates that any peer router at a destination is not addressable via the destination identifier presented in any of the destination orientated signals (e.g. Destination Update), and therefore MUST include at least one IPv4 or IPv6 Address TLV in the Destination Up signal.

## 2.1. Non MAC TLV

This OPTIONAL TLV is only valid in the Peer Initialization signal, and indicates that any destination addresses used during the lifetime of the session are not MAC addresses. The length field specifies the length in octets of all destination identifiers to be used during the session.

If the receiving DLEP router does not support this TLV then it SHOULD respond with a failure status in the corresponding Peer Initialization ACK signal as specified in the core DLEP specification.

The Non MAC TLV contains the following fields:

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	2	3	4	5	6	7	8	9
TLV Type = TBD										Length = 1										Id Length																			

TLV Type: TBD

Length: 1

**Id Length:** The length in octets of destination identifiers.

## 2.2. Destination Identifiers In Existing Data Items

The MAC Address TLV can be present in several DLEP signals: Destination Up, Destination Up ACK, Destination Down, Destination Down ACK, Destination Update, Link Characteristics Request, and Link Characteristics ACK. With this extension the use of the MAC Address TLV remains the same, but its format is adjusted. This adjustment is backwards-compatible with the core DLEP specification.

The MAC Address TLV is updated as follows:



```

      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
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|TLV Type = TBD | Length > 0 (6) | Dest ID |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|               Destination Identifier (cont...) |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

TLV Type: TBD (Same as DLEP core specification)

Length:

0 (As specified in Non MAC TLV if present, else 6)

Destination Identifier: Unique identifier of the destination.

### 3. External Route Advertisement

A modem operating as a layer-3 routing device may well have one or more accessible subnets addressable from a neighbouring modem, and it is often the case that these accessible routes need to be advertised throughout the radio net. To facilitate this advertisement, this specification includes the Route TLV.

The purpose of the external route advertisement is not to convert DLEP into a routing protocol but rather to enable routes to be advertised during the DLEP session. The method for discovering and propagating routes around the network is out of the scope of this document.

Using the Route TLV, an attached router can receive information about routes external to a peer router at a DLEP destination via the Destination Up and Destination Update signals. An attached peer router may also inject new routes in the DLEP session by using the Route TLV in the Peer Initialization and Peer Update signals. The Route TLV may be included in any DLEP signal where an IPv4 or IPv6 Address TLV may be used: Destination Up, Destination Update, Peer Initialization, and Peer Update signals.

Because external routes may be sourced from running routing protocol instances, this extension re-uses the structure and type codes of the UPDATE message specified in BGP-4 [RFC4271]. It is the opinion of the authors that BGP provides a common denominator in routing functionality and avoids the requirement for new IANA registries for data items already in use by BGP.

Unlike a BGP-4 UPDATE message, a Route TLV data item also allows the provision of DLEP metrics for an external route. These metrics MUST

follow all the rules for core DLEP metric data items. It should be noted that the metrics describe the state of the link between the destination router and the source of the route and MUST NOT include or aggregate the metrics for the link between the DLEP destination and the local modem with the metrics for the external route. This ensures that the responsibility for accumulating metrics for routes is with attached routers and not modems.

### 3.1. Route TLV

The Route TLV is an OPTIONAL data item. It is also made up of several OPTIONAL components. Its layout is heavily influenced by the structure of the BGP-4 UPDATE message.

The Route TLV contains the following fields:

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	2	3	4	5	6	7	8	9
TLV Type = TBD										Length										Withdrawn																			
Routes Length										Withdrawn Routes (variable)																													
Total Path Attributes Length										Path Attributes (variable)																													
Total Metrics Length										Route Metrics (variable)																													
Network Layer Reachability Information (variable)																																							

TLV Type: TBD

Length: Variable

Withdrawn Routes Length: As BGP-4 UPDATE Message

Withdrawn Routes: As BGP-4 UPDATE Message

Total Path Attribute Length: As BGP-4 UPDATE Message

Path Attributes: As BGP-4 UPDATE Message

Total Metrics Length: This 2-octets unsigned integer indicates the total length of the DLEP metric data item TLVs in octets. A value of 0 indicates that there is no metric information included in this route TLV.

Route Metrics: This variable length field contains a list of DLEP metric TLV data items, such as Maximum Data Rate (Receive). There MUST NOT be duplicate entries.

Network Layer Reachability Information: As BGP-4 UPDATE Message

#### 4. Security Considerations

As an extension to the core DLEP protocol, the security considerations of that protocol apply to this extension. This extension adds no additional security mechanisms or features.

General BGP security considerations are discussed in [RFC4271] and [RFC4272].

#### 5. IANA Considerations

This section specifies requests to IANA.

##### 5.1. Registration

This specification defines new DLEP TLVs that require new number assignment from the DLEP Data Items repository:

- o Non MAC TLV
- o Route Advertisement TLV

#### 6. Normative References

- [DLEP] Ratliff, S., Berry, B., Harrison, G., Jury, S., and D. Satterwhite, "Dynamic Link Exchange Protocol (DLEP)", draft-ietf-manet-dlep-05, February 2014.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC4271] Rekhter, Y., Li, T., and S. Hares, "A Border Gateway Protocol 4 (BGP-4)", RFC 4271, January 2006.
- [RFC4272] Murphy, S., "BGP Security Vulnerabilities Analysis", RFC 4272, January 2006.

#### Authors' Addresses

Rick Taylor  
Airbus Defence & Space  
Quadrant House  
Celtic Springs  
Coedkernew  
Newport NP10 8FZ  
UK

Email: [rick.taylor@cassidian.com](mailto:rick.taylor@cassidian.com)

John Dowdell  
Airbus Defence & Space  
Quadrant House  
Celtic Springs  
Coedkernew  
Newport NP10 8FZ  
UK

Email: [john.dowdell@cassidian.com](mailto:john.dowdell@cassidian.com)

Network Working Group  
Internet-Draft  
Intended status: Experimental  
Expires: January 30, 2015

J. Yi  
LIX, Ecole Polytechnique  
B. Parrein  
University of Nantes  
July 29, 2014

Multi-path Extension for the Optimized Link State Routing Protocol  
version 2 (OLSRv2)  
draft-yi-manet-olsrv2-multipath-02

Abstract

This document specifies a multi-path extension to the Optimized Link State Routing Protocol version 2 (OLSRv2) to discover multiple disjoint paths, so as to improve reliability of the OLSRv2 protocol. The interoperability with OLSRv2 is retained.

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on January 30, 2015.

Copyright Notice

Copyright (c) 2014 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as

described in the Simplified BSD License.

## Table of Contents

1. Introduction . . . . .	3
1.1. Motivation and Experiments to Be Conducted . . . . .	3
2. Terminology . . . . .	5
3. Applicability Statement . . . . .	5
4. Protocol Overview and Functioning . . . . .	6
5. Parameters and Constants . . . . .	6
5.1. Router Parameters . . . . .	6
6. Packets and Messages . . . . .	7
6.1. HELLO and TC messages . . . . .	7
6.1.1. MP_OLSRv2 TLV . . . . .	7
6.2. Datagram . . . . .	7
6.2.1. Source Routing Header in IPv4 . . . . .	7
6.2.2. Source Routing Header in IPv6 . . . . .	8
7. Information Bases . . . . .	8
7.1. MP-OLSRv2 Router Set . . . . .	8
7.2. Multi-path Routing Set . . . . .	8
8. Protocol Details . . . . .	9
8.1. HELLO and TC Message Generation . . . . .	9
8.2. HELLO and TC Message Processing . . . . .	9
8.3. Datagram Processing at the MP-OLSRv2 Originator . . . . .	10
8.4. Multi-path Dijkstra Algorithm . . . . .	10
8.5. Datagram Forwarding . . . . .	11
9. Configuration Parameters . . . . .	12
10. Implementation Status . . . . .	12
10.1. Multi-path extension based on nOLSRv2 . . . . .	13
10.2. Multi-path extension based on olsrd . . . . .	13
10.3. Multi-path extension based on umOLSR . . . . .	13
11. Security Considerations . . . . .	14
12. IANA Considerations . . . . .	14
12.1. HELLO Message-Type-Specific TLV Type Registries . . . . .	14
12.2. TC Message-Type-Specific TLV Type Registries . . . . .	14
13. References . . . . .	15
13.1. Normative References . . . . .	15
13.2. Informative References . . . . .	15
Appendix A. An example of Multi-path Dijkstra Algorithm . . . . .	16
Authors' Addresses . . . . .	17

## 1. Introduction

The Optimized Link State Routing Protocol version 2 (OLSRv2) [RFC7181] is a proactive link state protocol designed for use in mobile ad hoc networks (MANETs). It generates routing messages periodically to create and maintain a Routing Set, which contains routing information to all the possible destinations in the routing domain. For each destination, there exists a unique Routing Tuple, which indicates the next hop to reach the destination.

This document specifies an extension of the OLSRv2 protocol [RFC7181], to provide multiple disjoint paths for a source-destination pair. Because of the characteristics of MANETs [RFC2501], especially the dynamic topology, having multiple paths is helpful for increasing network throughput, improving transmission reliability and load balancing.

The Multi-path OLSRv2 (MP-OLSRv2) specified in this document uses multi-path Dijkstra algorithm to explore multiple disjoint paths from source to the destination based on the topology information obtained through OLSRv2, and forward the datagrams in a load-balancing manner using source routing. MP-OLSRv2 is designed to be interoperable with OLSRv2.

### 1.1. Motivation and Experiments to Be Conducted

This document is an experimental extension of OLSRv2 that can increase the data forwarding reliability in dynamic and high-load MANET scenarios by transmitting packet over multiple disjoint paths using source routing. This mechanism is used because:

- o Disjoint paths can avoid single route failure.
- o By having control of that paths at the source, the delay can be provisioned.
- o Certain scenarios require some routers must (or must not) be used.
- o An very important application of this extension is combination with Forward Error Correction coding. This requires disjoint paths. The single path routing is not adapted because the packet drop is normally continuous, in which forward correction coding is not helpful.

While existed deployments, running code and simulations have proven the interest of multipath extension for OLSRv2 in certain networks, more experiments and experiences are still needed to understand the mechanisms of the protocol. The multipath extension for OLSRv2 is

expected to be revised and improved to the Standard Track, once sufficient operational experience is obtained. Other than the general experiences including the protocol specification, interoperability with original OLSRv2 implementations, the experiences in the following aspects are highly appreciated:

- o Optimal values for the number of multiple paths (NUMBER\_OF\_PATHS) to be used. This depends on the network topology and router density.
- o Optimal values for the cost functions. Cost functions are applied to punish the costs of used links and nodes so as to obtain disjoint paths. What kind of disjointness is desired (node-disjoint or link-disjoint) may depends on the layer 2 protocol used, and can be achieved by setting different sets of cost functions.
- o Optimal choice of "key" routers for loose source routing. In some cases, loose source routing is use to reduce overhead or for interoperability with OLSRv2. Other than the basic rules defined in the following of this document, optimal choices of routers to put in the source routing header can be further studied.
- o Use of other metric other than hop-count. This multipath extension can be used not only for hop-count metric type, but other metric types that meet the requirement of OLSRv2, such as [I-D.ietf-manet-olsrv2-dat-metric]. The metric type used has also co-relation with the choice of cost functions as indicated in the previous bullet.
- o The impacts to the delay variation due to multi-path routing. [RFC2991] brings out some concerns of multi-path routing, especially variable latencies. Although current experiments result show that multi-path routing can reduce the jitter in dynamic scenarios, some transport protocols or applications may be sensitive to the packet re-ordering.
- o The disjoint multiple path protocol has interesting application with Forward Error Correction (FEC) Coding, especially for services like video/audio streaming. The combination of FEC coding mechanisms and this extension is thus encouraged.
- o In addition to IP source routing based approach, it can be interesting to try multi-path routing in MANET using label-switched flow in the future.
- o The usage of multi-topology information. By using [I-D.ietf-manet-olsrv2-multitopology], multiple topologies using



different metric types can be obtained. It is encouraged to experiment the use of multiple metrics for building multiple paths also.

## 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].

This document uses the terminology and notation defined in [RFC5444], [RFC6130], [RFC7181]. Additionally, it defines the following terminology:

OLSRv2 Routing Process - The routing process based on [RFC7181], without multi-path extension specified in this document.

MP-OLSRv2 Routing Process - The routing process based on this specification as an extension to [RFC7181].

## 3. Applicability Statement

As an extension of OLSRv2, this specification is applicable to MANETs for which OLSRv2 is applicable (see [RFC7181]). It can operate on single, or multiple interfaces, to discover multiple disjoint paths from a source router to a destination router.

MP-OLSRv2 is specially designed for networks with dynamic topology and slow data rate links. By providing multiple paths, higher aggregated bandwidth can be obtained, and the routing process is more robust to packet loss.

In a router supporting MP-OLSRv2, MP-OLSRv2 does not necessarily replace OLSRv2 completely. The extension can be applied for certain applications that are suitable for multi-path routing (mainly video or audio streams), based on the information such as DiifServ Code Point [RFC2474].

Compared to OLSRv2, this extension does not introduce new message type in the air, and is interoperable with OLSRv2 implementations that do not have this extension.

#### 4. Protocol Overview and Functioning

This specification requires OLSRv2 [RFC7181] to:

- o Identify all the reachable routers in the network.
- o Identify a sufficient subset of links in the networks, so that routes can be calculated to all reachable destinations.
- o Provide a Routing Set containing shortest routes from this router to all destinations.

Based on the above information acquired by OLSRv2, the MP-OLSRv2 Routing Process is able to calculate multiple paths to certain destinations based on multi-path Dijkstra algorithm: the Dijkstra algorithm is performed multiple times . In each iteration, the cost of used links are increased (i.e., punished), so that they can be avoided to be chosen in the next iteration. The multi-path Dijkstra algorithm can generate multiple disjoint paths from a source to a destination , and such information is kept in Multi-path Routing Set. The algorithm is invoked on demand, i.e., only when there is data traffic to be sent from the source to the destination, and there is no available routing tuples in the Multi-path Routing Set.

The datagram is forwarded based on source routing. When there is a datagram to be sent to a destination, the source router acquires a path from the Multi-path Routing Set (in Round-Robin fashion here) . The path information is stored in the datagram header as source routing header. The intermediate routers listed in the source routing header (SRH) read the SRH and forward the datagram to the next hop indicated in the SRH.

#### 5. Parameters and Constants

In addition to the parameters and constants defined in [RFC7181], this specification uses the parameters and constants described in this section.

##### 5.1. Router Parameters

**NUMBER\_OF\_PATHS**    The number of paths desired by the router.

**MAX\_SRC\_HOPS**    The maximum number of hops allowed to put in the source routing header.

fp Incremental function of multi-path Dijkstra algorithm. It is used to increase costs of links belonging to the previously computed path.

fe Incremental function of multi-path Dijkstra algorithm. It is used to increase costs of links who lead to routers of the previous computed path.

MR\_HOLD\_TIME It is the minimal time that a Multi-path Routing Tuple SHOULD be kept in the Multi-path Routing Set.

MP\_OLSR\_HOLD\_TIME It is the minimal time that a MP-OLSRv2 Router Tuple SHOULD be kept in the MP-OLSRv2 Router Set.

## 6. Packets and Messages

This extension employs the routing control messages HELLO and TC (Topology Control) as defined in OLSRv2 [RFC7181]. To support source routing, a source routing header is added to each datagram routed by this extension. Depending on the IP version used, the source routing header is defined in following of this section.

### 6.1. HELLO and TC messages

HELLO and TC messages used by MP-OLSRv2 Routing Process share the same format as defined in [RFC7181]. In addition, one Message TLV is defined, to identify the originator of the HELLO or TC message is running MP-OLSRv2.

#### 6.1.1. MP\_OLSRv2 TLV

An MP\_OLSRv2 TLV is a Message TLV that signals the message is generated by an MP-OLSRv2 Routing Process. It does not include any value.

Every HELLO or TC message generated by MP-OLSRv2 Routing Process MUST has one MP\_OLSRv2 TLV.

### 6.2. Datagram

#### 6.2.1. Source Routing Header in IPv4

In IPv4 [RFC0791] networks, the MP-OLSRv2 routing process employs loose source routing, as defined in [RFC0791]. It exists as an option header, with option class 0, and option number 3.

The source route information is kept in the "route data" field of the

loss source route header.

#### 6.2.2. Source Routing Header in IPv6

In IPv6 [RFC2460] networks, the MP-OLSRv2 routing process employs the source routing header as defined in [RFC6554], with IPv6 Routing Type 3.

The source route information is kept in the "Addresses" field of the routing header.

### 7. Information Bases

Each MP-OLSRv2 routing process maintains the information bases as defined in [RFC7181]. Additionally, two more information bases are defined for this specification.

#### 7.1. MP-OLSRv2 Router Set

The MP-OLSRv2 Router Set records the routers running the MP-OLSRv2 Routing Process. It consists of MP-OLSRv2 Router Tuples:

(MP\_OLSR\_addr, MP\_OLSR\_valid\_time)

where:

MP\_OLSR\_addr - it is the network address of the router that runs MP-OLSRv2 Routing Process;

MP\_OLSR\_valid\_time - it is the time until which the MP-OLSRv2 Router Tuples is considered valid.

#### 7.2. Multi-path Routing Set

The Multi-path Routing Set records the full path information of different paths to the destination. It consists of Multi-path Routing Tuples:

(MR\_dest\_addr, MR\_valid\_time, MR\_path\_set)

where:

MR\_dest\_addr - it is the network address of the destination, either the network address of an interface of a destination router or the network address of an attached network;

MR\_valid\_time - it is the time until which the Multi-path Routing Tuples is considered valid;

MP\_path\_set - it contains the multiple paths to the destination. It consists of Path Tuples.

Each Path Tuple is defined as:

(PT\_cost, PT\_address[1], PT\_address[2], ..., PT\_address[n])

where:

PT\_cost - the cost of the path to the destination;

PT\_address[1...n] - the addresses of intermediate router to be visited numbered from 1 to n.

## 8. Protocol Details

This protocol is based on OLSRv2, and extended to discover multiple disjoint paths from the source to the destination router. It retains the basic routing control packets formats and processing of OLSRv2 to obtain topology information of the network. The main differences between OLSRv2 routing process is the datagram processing at the source router and datagram forwarding.

### 8.1. HELLO and TC Message Generation

HELLO and TC messages are generated according to the section 15.1 or section 16.1 of [RFC7181].

A single Message-Type-Specific TLV with Type := MP\_OLSRv2 is added to the message.

### 8.2. HELLO and TC Message Processing

HELLO and TC messages are processed according to the section 15.3 and 16.3 of [RFC7181].

For every HELLO or TC message received, if there exists a TLV with Type := MP\_OLSRv2, create or update (if the tuple exists already) the MP-OLSR Router Tuple with

- o MP\_OLSR\_addr = originator of the HELLO or TC message

and set the MP\_OLSR\_valid\_time := current\_time + MP\_OLSR\_HOLD\_TIME.

### 8.3. Datagram Processing at the MP-OLSRv2 Originator

When the MP-OLSRv2 routing process receives a datagram from upper layers or interfaces connecting other routing domains, find the Multi-path Routing Tuple where:

- o MR\_dest\_addr = destination of the datagram, and
- o MR\_valid\_time < current\_time.

If a matching Multi-path Routing Tuple is found, a Path Tuple is chosen from the MR\_path\_set in Round-robin fashion (if there are multiple datagrams to be sent). Or else, the Multi-path Dijkstra Algorithm defined in Section 8.4 is invoked, to generate the desired Multi-path Routing Tuple.

The addresses in PT\_address[1...n] of the chosen Path Tuple are thus added to the datagram header in order as source routing header, following the rules:

- o Only the addresses exist in MP-OLSR Router Set can be added to the source routing header.
- o If the length of the path (n) is greater than MAX\_SRC\_HOPS, only the key routers in the path are kept. By default, the key routers are uniformly chosen in the path.
- o The routers with higher priority (such as higher willingness of routing) are preferred.
- o The routers that are considered not appropriate for forwarding indicated by external policies should be avoided.

### 8.4. Multi-path Dijkstra Algorithm

The Multi-path Dijkstra Algorithm is invoked when there is no available Multi-path Routing Tuple to a desired destination d. The general principle of the algorithm is at step i to look for the shortest path  $P_i$  to the destination d. Based on Dijkstra algorithm, the main modification consists in adding two cost functions namely incremental functions fp and fe in order to prevent the next steps to use similar path. fp is used to increase costs of arcs belonging to the previously path  $P_i$  (or which opposite arcs belong to it). This encourages future paths to use different arcs but not different vertices. fe is used to increase costs of the arcs who lead to vertices of the previous path  $P_i$ . It is possible to choose different fp and fe to get link-disjoint path or node-disjoint routes as necessary. A recommendation of configuration of fp and fe is given

in Section 5.

To get `NUMBER_OF_PATHS` distinct paths, for each path  $P_i$  ( $i = 1, \dots, \text{NUMBER\_OF\_PATHS}$ ) do:

1. Run Dijkstra algorithm to get the shortest path  $P_i$  for the destination  $d$ .
2. Apply cost function  $fp$  to the links in  $P_i$ .
3. Apply cost function  $fe$  to the links who lead to routers used in  $P_i$ .

A simple example of Multi-path Dijkstra Algorithm is illustrated in Appendix A.

By invoking the algorithm depicted above, `NUMBER_OF_PATHS` distinct paths is obtained, and added to the Multi-path Routing Set, by creating a Multi-path Routing Tuple with:

- o `MR_dest_addr` := destination  $d$
- o `MR_valid_time` := current time + `MR_HOLD_TIME`
- o Each Path Tuple in the `MP_path_set` corresponds to a path obtained in multi-path Dijkstra algorithm, with `PT_cost` := cost of the path to the destination  $d$ .

#### 8.5. Datagram Forwarding

On receiving a datagram with source routing header, the Destination Address field of the IP header is first compared to the addresses of the local interfaces. If no matching address is found, the datagram is forwarded according OLSRv2 routing process. If a matching local address is found, the datagram is processed as follows:

1. Obtain the next source address `Address[i]` in the source route header. How to obtain the next source address depends on the IP version used. In IPv4, the position of the next source address is indicated by the "pointer" field of the source routing header [RFC0791]. In IPv6, the position is indicated by "Segments Left" field of the source routing header. If no next source address is found, the forwarding process is finished.
2. Swap `Address[i]` and destination address in the IP header.
3. Forward the datagram to the destination address according to the OLSRv2 Routing Tuple information through `R_local_iface_addr` where

- \* R\_dest\_addr = destination address in the IP header

## 9. Configuration Parameters

This section gives default values and guideline for setting parameters defined in Section 5. Network administrator may wish to change certain, or all the parameters for different network scenarios. As an experimental track protocol, the users of this protocol are also encouraged to explore different parameter setting in various network environments, and provide feedback.

- o NUMBER\_OF\_PATHS = 3. This parameter defines the number of parallel paths used in datagram forwarding. Setting it to one makes the specification identical to OLSRv2. Setting it to too big value can lead to unnecessary computational overhead and inferior paths.
- o MAX\_SRC\_HOPS = 10.
- o MR\_HOLD\_TIME = 10 seconds.
- o  $fp(c) = 4 * c$ , where  $c$  is the original cost of the link.
- o  $fe(c) = 2 * c$ , where  $c$  is the original cost of the link.

The setting of cost functions  $fp$  and  $fc$  defines the preference of obtained multiple disjoint paths. If  $id$  is the identity functions, 3 cases are possible:

- o if  $id=fe<fp$  paths tend to be link disjoint;
- o if  $id<fe=fp$  paths tend to be node-disjoint;
- o if  $id<fe<fp$  paths also tend to be node-disjoint, but when is not possible they tend to be arc disjoint.

## 10. Implementation Status

This section records the status of known implementations of the protocol defined by this specification at the time of posting of this Internet-Draft, and based on a proposal described in [RFC6982]. The description of implementations in this section is intended to assist the IETF in its decision processes in progressing drafts to RFCs. Please note that the listing of any individual implementation here does not imply endorsement by the IETF. Furthermore, no effort has been spent to verify the information presented here that was supplied



by IETF contributors. This is not intended as, and must not be construed to be, a catalog of available implementations or their features. Readers are advised to note that other implementations may exist.

According to [RFC6982], "this will allow reviewers and working groups to assign due consideration to documents that have the benefit of running code, which may serve as evidence of valuable experimentation and feedback that have made the implemented protocols more mature. It is up to the individual working groups to use this information as they see fit".

Until April 2014, there are 3 open source implementations of the protocol specified in this document, for both testbed and simulation use.

#### 10.1. Multi-path extension based on nOLSRv2

The implementation is conducted by University of Nantes, France, and is based on Niigata University's nOLSRv2 implementation. It is an open source implementation. The code is available at <http://jiaziyi.com/index.php/research-projects/mp-olsr> .

It can be used for Qualnet simulations, and be exported to run in testbed. All the specification is implemented in this implementation.

Implementation experience and test data can be found at [ADHOC11].

#### 10.2. Multi-path extension based on olsrd

The implementation is conducted under SEREADMO (Securite des Reseaux Ad Hoc & Mojette) project, and supported by French research agency (RNRT2803). It is based on olsrd (<http://www.olsr.org/>) implementation, and is open sourced. The code is available at <http://jiaziyi.com/index.php/research-projects/sereadmo> .

The implementation is for testing the specification in the field. All the specification is implemented in this implementation.

Implementation experience and test data can be found at [ADHOC11].

#### 10.3. Multi-path extension based on umOLSR

The implementation is conducted by University of Nantes, France, and is based on um-olsr implementation (<http://masimum.inf.um.es/fjrm/development/um-olsr/>). The code is available at <http://jiaziyi.com/index.php/research-projects/mp-olsr>

under GNU GPL license.

The implementation is just for network simulation for NS2 network simulator. All the specification is implemented in this implementation.

Implementation experience and test data can be found at [WCNC08].

## 11. Security Considerations

This document does currently not specify any security considerations....

## 12. IANA Considerations

This specification defines two Message TLV Types, which must be allocated from the Message TLV Types repository of [RFC5444].

### 12.1. HELLO Message-Type-Specific TLV Type Registries

IANA is requested to create a registry for Message-Type-Specific Message TLV for HELLO messages, in accordance with Section 6.2.1 of [RFC5444], and with initial assignments and allocation policies as specified in Table 1.

Type	Description	Allocation Policy
128	MP_OLSRv2	
129-223	Unassigned	Expert Review

Table 1: HELLO Message-Type-specific Message TLV Types

### 12.2. TC Message-Type-Specific TLV Type Registries

IANA is requested to create a registry for Message-Type-Specific Message TLV for TC messages, in accordance with Section 6.2.1 of [RFC5444], and with initial assignments and allocation policies as specified in Table 2.

Type	Description	Allocation Policy
128	MP_OLSRv2	
129-223	Unassigned	Expert Review

Table 2: TC Message-Type-specific Message TLV Types

### 13. References

#### 13.1. Normative References

- [RFC0791] Postel, J., "Internet Protocol", STD 5, RFC 791, September 1981.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC5444] Clausen, T., Dearlove, C., Dean, J., and C. Adjih, "Generalized Mobile Ad Hoc Network (MANET) Packet/Message Format", RFC 5444, February 2009.
- [RFC6130] Clausen, T., Dearlove, C., and J. Dean, "Mobile Ad Hoc Network (MANET) Neighborhood Discovery Protocol (NHDP)", RFC 6130, April 2011.
- [RFC6554] Hui, J., Vasseur, JP., Culler, D., and V. Manral, "An IPv6 Routing Header for Source Routes with the Routing Protocol for Low-Power and Lossy Networks (RPL)", RFC 6554, March 2012.
- [RFC7181] Clausen, T., Dearlove, C., Jacquet, P., and U. Herberg, "The Optimized Link State Routing Protocol Version 2", RFC 7181, April 2014.

#### 13.2. Informative References

- [ADHOC11] Yi, J., Adnane, A-H., David, S., and B. Parrein, "Multipath optimized link state routing for mobile ad hoc networks", In Elsevier Ad Hoc Journal, vol.9, n. 1, 28-47, January, 2011.
- [I-D.ietf-manet-olsrv2-dat-metric] Rogge, H. and E. Baccelli, "Packet Sequence Number based directional airtime metric for OLSRv2", draft-ietf-manet-olsrv2-dat-metric-01 (work in progress),

July 2014.

- [I-D.ietf-manet-olsrv2-multitopology]  
Dearlove, C. and T. Clausen, "Multi-Topology Extension for the Optimized Link State Routing Protocol version 2 (OLSRv2)", draft-ietf-manet-olsrv2-multitopology-04 (work in progress), July 2014.
- [RFC2460] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, December 1998.
- [RFC2474] Nichols, K., Blake, S., Baker, F., and D. Black, "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers", RFC 2474, December 1998.
- [RFC2501] Corson, M. and J. Macker, "Mobile Ad hoc Networking (MANET): Routing Protocol Performance Issues and Evaluation Considerations", RFC 2501, January 1999.
- [RFC2991] Thaler, D. and C. Hopps, "Multipath Issues in Unicast and Multicast Next-Hop Selection", RFC 2991, November 2000.
- [RFC6982] Sheffer, Y. and A. Farrel, "Improving Awareness of Running Code: The Implementation Status Section", RFC 6982, July 2013.
- [WCNC08] Yi, J., Cizeron, E., Hama, S., and B. Parrein, "Simulation and performance analysis of MP-OLSR for mobile ad hoc networks", In Proceeding of IEEE Wireless Communications and Networking Conference, 2008.

#### Appendix A. An example of Multi-path Dijkstra Algorithm

This appendix gives an example of multi-path Dijkstra algorithm. The network topology is depicted in Figure 1.

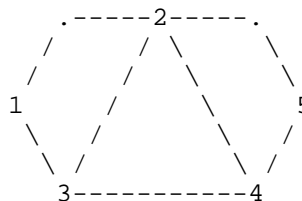


Figure 1: Network Topology for the on-demand example

The initial cost of all the links is set to 1. The incremental functions  $fp$  and  $fe$  are defined as  $fp(c)=4c$  and  $fe(c)=2c$  in this example. Two routes from node 1 to node 5 are demanded.

On the first run of the Dijkstra algorithm, the shortest path 1->2->5 with cost 2 is obtained.

The incremental function  $fp$  is applied to increase the cost of the link 1-2 and 2-5, from 1 to 4.  $fe$  is applied to increase the cost of the link 1-3, 2-3, 2-4, 4-5, from 1 to 2.

On the second run of the Dijkstra algorithm, the second path 1->3->4->5 with cost 5 is obtained.

#### Authors' Addresses

Jiazi Yi  
LIX, Ecole Polytechnique  
91128 Palaiseau Cedex,  
France  
  
Phone: +33 1 77 57 80 85  
Email: [jiazi@jiaziyi.com](mailto:jiazi@jiaziyi.com)  
URI: <http://www.jiaziyi.com/>

Benoit Parrein  
University of Nantes  
IRCCyN lab - IVC team  
Polytech Nantes, rue Christian Pauc, BP50609  
44306 Nantes cedex 3  
France  
  
Phone: +33 (0) 240 683 050  
Email: [Benoit.Parrein@polytech.univ-nantes.fr](mailto:Benoit.Parrein@polytech.univ-nantes.fr)  
URI: <http://www.irccyn.ec-nantes.fr/~parrein>



Mobile Ad hoc Networking (MANET)  
Internet-Draft  
Intended status: Informational  
Expires: January 5, 2015

J. Yi  
T. Clausen  
LIX, Ecole Polytechnique  
U. Herberg  
Fujitsu Laboratories of America  
July 4, 2014

Security Threats for Simplified Multicast Forwarding (SMF)  
draft-yi-manet-smf-sec-threats-00

Abstract

This document analyzes security threats of the Simplified Multicast Forwarding (SMF), including the vulnerabilities of duplicate packet detection and relay set selection mechanisms. This document is not intended to propose solutions to the threats described.

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on January 5, 2015.

Copyright Notice

Copyright (c) 2014 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as

described in the Simplified BSD License.

## Table of Contents

1. Introduction . . . . .	3
2. Terminology . . . . .	3
3. SMF Threats Overview . . . . .	4
4. Threats to Duplicate Packet Detection . . . . .	5
4.1. Threats to Identification-based Duplicate Packet Detection . . . . .	5
4.1.1. Pre-activation Attacks (Pre-Play) . . . . .	6
4.1.2. De-activation Attacks (Sequence Number wrangling) . . . . .	6
4.2. Threats to Hash-based Duplicate Packet Detection . . . . .	7
4.2.1. Replay Attack . . . . .	7
4.2.2. Attack on Hash-Assistant Value . . . . .	8
5. Threats to Relay Set Selection . . . . .	8
5.1. Relay Set Selection Common Threats . . . . .	9
5.2. Threats to E-CDS Algorithm . . . . .	9
5.2.1. Link Spoofing . . . . .	9
5.2.2. Identity Spoofing . . . . .	9
5.3. Threats to S-MPR Algorithm . . . . .	10
5.4. Threats to MPR-CDS Algorithm . . . . .	10
6. Security Considerations . . . . .	10
7. IANA Considerations . . . . .	11
8. References . . . . .	11
8.1. Normative References . . . . .	11
8.2. Informative References . . . . .	11
Authors' Addresses . . . . .	12



## 1. Introduction

This document analyzes security threats of the Simplified Multicast Forwarding (SMF) mechanism [RFC6621]. SMF aims at providing basic Internet Protocol (IP) multicast forwarding, in a way which is suitable for limited wireless mesh and Mobile Ad hoc NETWORKS (MANET). SMF is constituted of two major functional components: Duplicate Packet Detection and Relay Set Selection.

SMF is typically used in decentralized wireless environments, and is potentially exposed to different kinds of attacks and misconfigurations. Some of the threats are of particular significance as compared to wired networks. In [RFC6621], SMF does not define any explicit security measures for protecting the integrity of the protocol.

This document is based on the assumption that no additional security mechanism such as IPsec is used in the IP layer, as not all MANET deployments may be suitable to deploy common IP protection mechanisms (e.g., because of limited resources of MANET routers to support the IPsec stack). The document analyzes possible attacks on and misconfigurations of SMF and outlines the consequences of such attacks/misconfigurations to the state maintained by SMF in each router (and, thus, made available to protocols using this state).

This document aims at analyzing and describing the potential vulnerabilities of and attack vectors for SMF. While completeness in such an analysis always is a goal, no claims of being complete are made. The goal of this document is to be helpful for when deploying SMF in a network and needing to understand the risks thereby incurred - as well as for providing a reference and documented experience with SMF as input for possibly future developments of SMF.

This document is not intended to propose solutions to the threats described. [RFC7182] provides a framework, which can be used with SMF, and which - depending on how it is used - may offer some degree of protection against the threats described in this document related to identity spoofing.

## 2. Terminology

This document uses the terminology and notation defined in [RFC2119], [RFC5444], [RFC6621] and [RFC4949].

Additionally, this document introduces the following terminology:

SMF router: A MANET router, running SMF as specified in [RFC6621].

Attacker: A device that is present in the network and intentionally seeks to compromise the information bases in SMF routers.

Compromised SMF router: An attacker, present in the network and which generates syntactically correct SMF control messages. Control messages emitted by a compromised SMF router may contain additional information, or omit information, as compared to a control message generated by a non-compromised SMF router located in the same topological position in the network.

Legitimate SMF router: An SMF router, which is not a compromised SMF Router.

### 3. SMF Threats Overview

SMF requires an external dynamic neighborhood discovery mechanism in order to maintain suitable topological information describing its immediate neighborhood, and thereby allowing it to select reduced relay sets for forwarding multicast data traffic. Such an external dynamic neighborhood discovery mechanism MAY be provided by lower-layer interface information, by a concurrently operating MANET routing protocol which already maintains such information such as [RFC7181], or by explicitly using MANET Neighborhood Discovery Protocol (NHDP) [RFC6130]. If NHDP is used for neighborhood discovery by SMF, SMF implicitly inherits the vulnerabilities of NHDP, as discussed in [RFC7186]. This document assumes that NHDP is used.

Based on neighborhood discovery mechanisms, SMF specified two major functional components: Duplicate Packet Detection (DPD) and Relay Set Selection (RSS).

DPD is required by SMF in order to be able to detect duplicate packets and eliminate their redundant forwarding. An Attacker has several ways in which to harm the DPD mechanisms:

- o It can "deactivate" DPD, so as to make it such that duplicate packets are not correctly detected, and that as a consequence they are (redundantly) transmitted, increasing the load on the network, draining the batteries of the routers involved, etc.
- o It can "pre-activate" DPD, so as to make DPD detect a later arriving (valid) packet as being a duplicate, which therefore won't be forwarded"

The attacks on DPD are detailed in Section 4.

RSS produces a reduced relay set for forwarding multicast data packets across the MANET. SMF supports the use of several relay set algorithms, including E-CDS (Essential Connected Dominating Set), S-MPR (Source-based Multi-point Relay, as known from [RFC3626] and [RFC7181]), or MPR-CDS. An Attacker can disrupt the RSS algorithm, by degrading it to classical flooding, or by "masking" certain part of the routers from the multicasting domain. The attacks to RSS algorithms are illustrated in Section 5.

#### 4. Threats to Duplicate Packet Detection

Duplicate Packet Detection (DPD) is required for packet dissemination in MANET because the packets may be transmitted via the same physical interface as the one over which they were received. A router may also receive multiple copies of the same packets from different neighbors. DPD is thus used to check if an incoming packet has been received or not.

DPD is achieved by a router maintaining a record of recently processed multicast packets, and comparing later received multicast herewith. A duplicate packet detected is silently dropped, and is not inserted into the forwarding path of that router, nor is it delivered to an application. DPD, as proposed by SMF, supports both IPv4 and IPv6 and for each suggests two duplicate packet detection mechanisms: 1) header content identification-based DPD (I-DPD), using packet headers, in combination with flow state, to estimate temporal uniqueness of a packet, and 2) hash-based DPD (H-DPD), employing hashing of selected header fields and payload for the same effect.

As they are distinct mechanisms, the threats to I-DPD and H-DPD are discussed separately.

##### 4.1. Threats to Identification-based Duplicate Packet Detection

I-DPD uses a specific DPD identifier in the packet header to identify a packet. By default, such packet identification is not provided by the IP packet header (for both IPv4 and IPv6). Therefore, additional identification header, such as the fragment header, a hop-by-hop header option, or IPSec sequencing, must be employed in order to support I-DPD. The uniqueness of a packet can then be identified by the [source IP address] of the packet originator, and the [sequence number] (from the fragment header, hop-by-hop header option, or IPSec). By doing so, each intermediate router can keep a record of recently received packet, and determine the coming packet has been received or not.

#### 4.1.1.1. Pre-activation Attacks (Pre-Play)

In a wireless environment, or across any other shared channel, a compromised SMF router can perceive the identification tuple [source IP, sequence number] of a packet. If sequence number progression is predictable, then it is trivial to generate and inject invalid packets with "future" identification information into the network. If these invalid packets arrive before the legitimate packets that they're spoofing, the latter will be treated as a duplicates and discarded. This can prevent multicast packets from reaching parts of the network.

Figure 1 gives an example of pre-activation attack. A, B, and C are legitimate SMF routers, and X is the compromised SMF router. The line between the routers presents the packet forwarding. Router A is the source and originates a multicast packet with sequence number n. When router X receives the packet, it generates an invalid packet with the the source address of A, and sequence number n. If the invalid packet arrives at router C before the forwarding of router B, the valid packet will be dropped by C as duplicate packet. In a wireless environment, jitter is commonly used to avoid systematic collisions at MAC layer [RFC5148], thus an attacker can increase the probability that its invalid packets arrive first by retransmitting them without jittering.

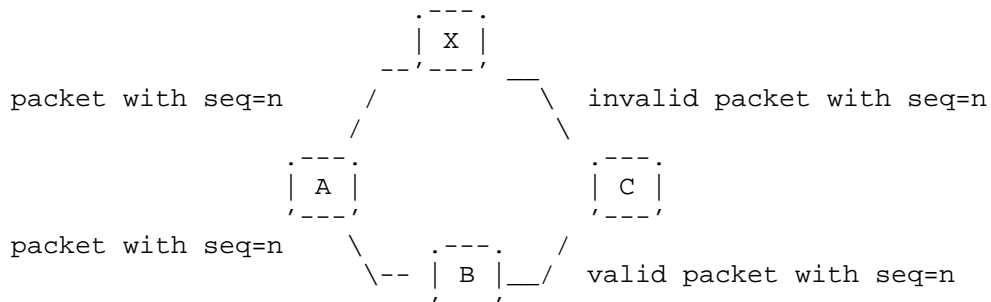


Figure 1

#### 4.1.1.2. De-activation Attacks (Sequence Number wrangling)

A compromised SMF router can also seek to de-activate DPD, by modifying the sequence number in packets that it forwards. Thus, routers will not be able to detect an actual duplicate packet as a duplicate - rather, they will treat them as new packets, i.e., process and forward them. This is similar to DoS attack. The consequence of this attack is an increased channel load, the origin

of which appears to be a router other than the compromised SMF router.

Given the topology shown in Figure 1, on receiving packet with seq=n, the attacker X can forward the packet with modified sequence number n+i. This has two consequences: firstly, router C will not be able to detect the packet forwarded by X is a duplicate packet; secondly, the consequent packet with seq=n+i generated by router A probably will be treated as duplicate packet, and dropped by router C.

#### 4.2. Threats to Hash-based Duplicate Packet Detection

When it is not feasible to have explicit sequence numbers in packet headers, hash-based DPD can be used. A hash of the non-mutable fields in the header of and the data payload can be generated, and recorded at the intermediate routers. A packet can thus be uniquely identified by the source IP address of the packet, and its hash-value.

The hash algorithm used by SMF is being applied only to provide a reduced probability of collision and is not being used for cryptographic or authentication purposes. Consequently, a digest collision is still possible. In case the source router or gateway identifies that it recently has generated or injected a packet with the same hash-value, it inserts a "Hash-Assist Value (HAV)" IPv6 header option into the packet, such that calculating the hash also over this HAV will render the resulting value unique.

##### 4.2.1. Replay Attack

A replay attack implies that control traffic from one region of the network is recorded and replayed in a different region at (almost) the same time, or in the same region at a different time.

One possible replay attack is based on the Time-to-Live (TTL, for IPv4) or hop limit (for IPv6) field. As routers only forward packets with TTL > 1, a compromised SMF router can forward an otherwise valid packet, while drastically reducing the TTL hereof. This will inhibit recipient routers from later forwarding the same multicast packet, even if received with a different TTL - essentially a compromised SMF router thus can instruct its neighbors to block forwarding of valid multicast packets. As the TTL of a packet is intended to be manipulated by intermediaries forwarding it, classic methods such as integrity check values (e.g., digital signatures) are typically calculated with setting TTL fields to some pre-determined value (e.g., 0) - such is for example the case for IPsec Authentication Headers - rendering such an attack more difficult to both detect and counter. If the compromised SMF router has access to a "wormhole"

through the network (a directional antenna, a tunnel to a collaborator or a wired connection, allowing it to bridge parts of a network otherwise distant) it can make sure that the packets with such an artificially reduced TTL arrive before their unmodified counterparts.

#### 4.2.2. Attack on Hash-Assistant Value

The HAV header is helpful when a digest collision happens. However, it also introduces a potential vulnerability. As the HAV option is only added when the source or the ingressing SMF router detects that the coming packet has digest collision with previously generated packets, it actually can be regarded as a "flag" of potential digest collision. A compromised SMF router can discover the HAV header, and be able to conclude a hash collision is possible if the HAV header is removed. By doing so, other SMF routers receiving the modified packet will be treated as duplicate packet, and be dropped because it has the same hash value with precedent packet.

In the example of Figure Figure 2, Router A and B are legitimate SMF routers, X is a compromised SMF router. A generate two packets P1 and P2, with the same hash value  $h(P1)=h(P2)=x$ . Based on SMF specification, a hash-assistant value (HAV) is added to the latter packet P2, so that  $h(P2+HAV)=x'$ , to avoid digest collision. When the attacker X detects the HAV of P2, it is able to conclude that a collision is possible by removing the HAV header. By doing so, packet P2 will be treated as duplicate packet by router B, and be dropped.

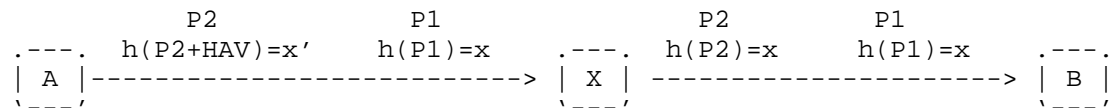


Figure 2

#### 5. Threats to Relay Set Selection

A framework for RSS mechanism, rather than a specific RSS algorithm is provided by SMF. It is normally achieved by distributed algorithms that can dynamically generate a topological Connected Dominating Set based on 1-hop and 2-hop neighborhood information. In this section, the common threats to the RSS framework are first discussed. Then the three commonly used algorithms: Essential

Connection Dominating Set (E-CDS) algorithm, Source-based Multipoint Relay (S-MPR) and Multipoint Relay Connected Dominating Set (MPR-CDS) are analyzed.

#### 5.1. Relay Set Selection Common Threats

The common threats to RSS algorithms, including Denial of Service attack, eavesdropping, message timing attack and broadcast storm have been discussed in [RFC7186].

#### 5.2. Threats to E-CDS Algorithm

The "Essential Connected Dominating Set" (E-CDS) algorithm [RFC5614] forms a single CDS mesh for the SMF operating region. It requires 2-hop neighborhood information (the identify of the neighbors, the link to the neighbors and neighbors' priority information) collected through NHDP or another process.

An SMF Router select itself as a relay, if:

- o The SMF Router has a higher priority than all of its symmetric neighbors, or
- o There does not exist a path from the neighbor with largest priority to any other neighbor, via neighbors with greater priority.

A Compromised SMF Router can disrupt the E-CDS algorithm by link spoofing or identity spoofing.

##### 5.2.1. Link Spoofing

Link spoofing implies that a Compromised SMF Router advertises non-existing links to another router (present in the network or not).

A Compromised SMF Router can declare itself with high route priority, and spoofs the links to as many Legitimate SMF Routers as possible to declare high connectivity. By doing so, it can prevent Legitimate SMF Routers from self-selecting as relays. As the "super" relay in the network, the Compromised SMF Router can manipulate the traffic relayed by it.

##### 5.2.2. Identity Spoofing

Identity spoofing implies that a compromised SMF router determines and makes use of the identity of other legitimate routers, without being authorised to do so. The identity of other routers can be obtained by overhearing the control messages or source/destination

address from datagram. The compromised SMF router can then generate control or datagram traffic, pretending to be a legitimate router.

Because E-CDS self-selection is based on the router priority value, a compromised SMF router can spoof the identity of other legitimate routers, and declares a different router priority value. If it declares a higher priority of a spoofed router, it can prevent other routers from selecting themselves as relays. On the other hand, if the compromised router declares lower priority of a spoofed router, it can enforces other routers to selecting themselves as relays, to degrade the multicast forwarding to classical flooding.

### 5.3. Threats to S-MPR Algorithm

The source-based multipoint relay (S-MPR) set selection algorithm enables individual routers, using 2-hop topology information, to select relays from their set of neighboring routers. MPRs are selected so that forwarding to the router's complete 2-hop neighbor set is covered.

An SMF router forwards a multicast packet if and only if:

- o the packet is not received before, and
- o the neighbor from which the packet was received has selected the router as MPR.

Because MPR calculation is based on the willingness declared by the SMF routers, and the connectivity of the routers, it can be disrupted by both link spoofing and identity spoofing. The threats and its impacts have been illustrated in section 5.1 of [RFC7186].

### 5.4. Threats to MPR-CDS Algorithm

MPR-CDS is a derivative from S-MPR. The main difference between S-MPR and MPR-CDS is that while S-MPR forms a different broadcast tree for each source in the network, MPR-CDS forms a unique broadcast tree for all sources in the network.

As MPR-CDS combines E-CDS and S-MPR, the vulnerabilities of E-CDS and S-MPR that discussed in Section 5.2 and Section 5.3 apply to MPR-CDS also.

## 6. Security Considerations

This document does not specify a protocol or a procedure. The document, however, reflects on security considerations for SMF for



packet dissemination in MANETs.

## 7. IANA Considerations

This document contains no actions for IANA.

## 8. References

### 8.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC5614] Ogier, R. and P. Spagnolo, "Mobile Ad Hoc Network (MANET) Extension of OSPF Using Connected Dominating Set (CDS) Flooding", RFC 5614, August 2009.
- [RFC6621] Macker, J., "Simplified Multicast Forwarding", RFC 6621, May 2012.
- [RFC7186] Yi, J., Herberg, U., and T. Clausen, "Security Threats for the Neighborhood Discovery Protocol (NHDP)", RFC 7186, April 2014.

### 8.2. Informative References

- [RFC3626] Clausen, T. and P. Jacquet, "The Optimized Link State Routing Protocol", RFC 3626, October 2003.
- [RFC4949] Shirey, R., "Internet Security Glossary, Version 2", RFC 4949, August 2007.
- [RFC5148] Clausen, T., Dearlove, C., and B. Adamson, "Jitter Considerations in Mobile Ad Hoc Networks (MANETs)", RFC 5148, February 2008.
- [RFC5444] Clausen, T., Dearlove, C., Dean, J., and C. Adjih, "Generalized MANET Packet/Message Format", RFC 5444, February 2009.
- [RFC6130] Clausen, T., Dean, J., and C. Dearlove, "Mobile Ad Hoc Network (MANET) Neighborhood Discovery Protocol (NHDP)", RFC 6130, April 2011.
- [RFC7181] Clausen, T., Dearlove, C., Jacquet, P., and U. Herberg, "The Optimized Link State Routing Protocol version 2",

RFC 7181, April 2014.

[RFC7182] Herberg, U., Clausen, T., and C. Dearlove, "Integrity Check Value and Timestamp TLV Definitions for Mobile Ad Hoc Networks (MANETs)", RFC 7182, April 2014.

#### Authors' Addresses

Jiazi Yi  
LIX, Ecole Polytechnique  
91128 Palaiseau Cedex,  
France

Phone: +33 1 77 57 80 85  
Email: [jiazi@jiaziyi.com](mailto:jiazi@jiaziyi.com)  
URI: <http://www.jiaziyi.com/>

Thomas Heide Clausen  
LIX, Ecole Polytechnique  
91128 Palaiseau Cedex,  
France

Phone: +33 6 6058 9349  
Email: [T.Clausen@computer.org](mailto:T.Clausen@computer.org)  
URI: <http://www.thomasclausen.org/>

Ulrich Herberg  
Fujitsu Laboratories of America  
1240 E Arques Ave  
Sunnyvale, CA 94085  
USA

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

