

MANET  
Internet-Draft  
Intended status: Experimental  
Expires: May 27, 2016

H. Rogge  
Fraunhofer FKIE  
E. Baccelli  
INRIA  
November 24, 2015

**Packet Sequence Number based directional airtime metric for OLSRv2  
draft-ietf-manet-olsrv2-dat-metric-10**

Abstract

This document specifies an Directional Airtime (DAT) link metric for usage in OLSRv2.

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 May 27, 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

<a href="#">1.</a>	<a href="#">Introduction</a>	<a href="#">2</a>
<a href="#">2.</a>	<a href="#">Terminology</a>	<a href="#">3</a>
<a href="#">3.</a>	<a href="#">Applicability Statement</a>	<a href="#">4</a>
<a href="#">4.</a>	<a href="#">Directional Airtime Metric Rationale</a>	<a href="#">5</a>
<a href="#">5.</a>	<a href="#">Metric Functioning &amp; Overview</a>	<a href="#">6</a>
<a href="#">6.</a>	<a href="#">Protocol Constants</a>	<a href="#">7</a>
<a href="#">7.</a>	<a href="#">Protocol Parameters</a>	<a href="#">8</a>
<a href="#">7.1.</a>	<a href="#">Recommended Values</a>	<a href="#">8</a>
<a href="#">8.</a>	<a href="#">Data Structures</a>	<a href="#">8</a>
<a href="#">8.1.</a>	<a href="#">Initial Values</a>	<a href="#">9</a>
<a href="#">9.</a>	<a href="#">Packets and Messages</a>	<a href="#">10</a>
<a href="#">9.1.</a>	<a href="#">Definitions</a>	<a href="#">10</a>
9.2.	<a href="#">Requirements for using DAT metric in OLSRV2 implementations</a>	<a href="#">10</a>
<a href="#">9.3.</a>	<a href="#">Link Loss Data Gathering</a>	<a href="#">11</a>
<a href="#">9.4.</a>	<a href="#">HELLO Message Processing</a>	<a href="#">11</a>
<a href="#">10.</a>	<a href="#">Timer Event Handling</a>	<a href="#">12</a>
<a href="#">10.1.</a>	<a href="#">Packet Timeout Processing</a>	<a href="#">12</a>
<a href="#">10.2.</a>	<a href="#">Metric Update</a>	<a href="#">12</a>
<a href="#">11.</a>	<a href="#">Security Considerations</a>	<a href="#">13</a>
<a href="#">12.</a>	<a href="#">Acknowledgements</a>	<a href="#">14</a>
<a href="#">13.</a>	<a href="#">References</a>	<a href="#">14</a>
<a href="#">13.1.</a>	<a href="#">Normative References</a>	<a href="#">14</a>
<a href="#">13.2.</a>	<a href="#">Informative References</a>	<a href="#">15</a>
<a href="#">Appendix A.</a>	<a href="#">Future work</a>	<a href="#">16</a>
<a href="#">Appendix B.</a>	<a href="#">OLSR.org metric history</a>	<a href="#">17</a>
<a href="#">Appendix C.</a>	<a href="#">Linkspeed stabilization</a>	<a href="#">18</a>
<a href="#">Appendix D.</a>	<a href="#">Packet loss hysteresis</a>	<a href="#">18</a>
<a href="#">Appendix E.</a>	<a href="#">Example DAT values</a>	<a href="#">18</a>
	<a href="#">Authors' Addresses</a>	<a href="#">19</a>

**[1. Introduction](#)**

One of the major shortcomings of Optimized Link State Routing (OLSR) [[RFC3626](#)] is the lack of a granular link cost metric between OLSR routers. Operational experience with OLSR networks gathered since its publication has revealed that wireless networks links can have highly variable and heterogeneous properties. This makes a hopcount metric insufficient for effective OLSR routing.

Based on this experience, OLSRV2 [[RFC7181](#)] integrates the concept of link metrics directly into the core specification of the routing protocol. The OLSRV2 routing metric is an external process, it can be any kind of dimensionless additive cost function which reports to the OLSRV2 protocol.



Since 2004 the OLSR.org [[OLSR.org](http://OLSR.org)] implementation of OLSR included an Estimated Transmission Count (ETX) metric [[MOBICOM04](#)] as a proprietary extension. While this metric is not perfect, it proved to be sufficient for a long time for Community Mesh Networks (Appendix B). But the increasing maximum data rate of IEEE 802.11 made the ETX metric less efficient than in the past, which is one reason to move to a different metric.

This document describes a Directional Airtime routing metric for OLSRv2, a successor of the OLSR.org ETX-derived routing metric for OLSR. It takes both the loss rate and the link speed into account to provide a more accurate picture of the links within the network.

This experimental draft will allow OLSRv2 deployments with a metric defined by the IETF Manet group. It enables easier interoperability tests between implementations and will also deliver a useful baseline to compare other metrics to. [Appendix A](#) contains a few possible steps to improve the Directional Airtime Metric.

## **2. Terminology**

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

The terminology introduced in [[RFC5444](#)], [[RFC7181](#)] and [[RFC6130](#)], including the terms "packet", "message" and "TLV" are to be interpreted as described therein.

Additionally, this document uses the following terminology and notational conventions:

DAT - Directional Airtime (Metric), the link metric described in this document, which is a directional variant of ETT. It does not take reverse path loss into account.

QUEUE - a first in, first out queue of integers.

QUEUE[TAIL] - the most recent element in the queue.

add(QUEUE, value) - adds a new element to the TAIL of the queue.

remove(QUEUE) - removes the HEAD element of the queue

sum(QUEUE) - an operation which returns the sum of all elements in a QUEUE.



`diff_seqno(new, old)` - an operation which returns the positive distance between two elements of the circular sequence number space defined in [section 5.1 of \[RFC5444\]](#). Its value is either  $(new - old)$  if this result is positive, or else its value is  $(new - old + 65536)$ .

`MAX(a, b)` - the maximum of a and b.

`MIN(a, b)` - the minimum of a and b.

`UNDEFINED` - a value not in the normal value range of a variable.

`airtime` - the time a transmitted packet blocks the link layer, e.g., a wireless link.

`ETX` - Expected Transmission Count, a link metric proportional to the number of transmissions to successfully send an IP packet over a link.

`ETT` - Estimated Travel Time, a link metric proportional to the amount of airtime needed to successfully transmit an IP packet over a link, not considering layer-2 overhead created by preamble, backoff time and queuing.

### **3. Applicability Statement**

The Directional Airtime Metric was designed and tested (see [\[olsrv2 paper\]](#)) in wireless IEEE 802.11 OLSRV2 [\[RFC7181\]](#) networks. These networks employ link layer retransmission to increase the delivery probability. A dynamic rate selection algorithm selects the unicast data rate independently for each neighbor.

As specified in OLSRV2, the metric calculates only the incoming link cost. It does neither calculate the outgoing metric, nor does it decide the link status (heard, symmetric, lost).

The metric works both for nodes which can send/receive [\[RFC5444\]](#) packet sequence numbers and those which do not have this capability. In the absence of such sequence numbers the metric calculates the packet loss based on [\[RFC6130\]](#) HELLO message timeouts.

The metric must learn about the unicast data rate towards each one-hop neighbor from an external process, either by configuration or by an external measurement process. This measurement could be done by gathering cross-layer data from the operating system or an external daemon like DLEP [\[DLEP\]](#), but also by indirect layer-3 measurements like packet-pair (see [\[MOBICOM04\]](#)).



The metric uses [[RFC5444](#)] multicast control traffic to determine the link packet loss. The administrator should take care that link layer multicast transmission do not have a higher reception probability than the slowest unicast transmission without retransmission. It might, for example in 802.11g, be necessary to increase the data-rate of the multicast transmissions, e.g. set the multicast data-rate to 6 MBit/s.

The metric can only handle a certain range of packet loss and unicast data-rate. The maximum packet loss that can be encoded into the metric a loss of 7 of 8 packets (87.5%), without link layer retransmissions. The unicast data-rate that can be encoded by this metric can be between 1 kBit/s and 2 GBit/s. This metric has been designed for data-rates of 1 MBit/s and hundreds of MBit/s.

#### **4. Directional Airtime Metric Rationale**

The Directional Airtime Metric has been inspired by the publications on the ETX [[MOBICOM03](#)] and ETT [[MOBICOM04](#)] metric, but differs from both of these in several ways.

Instead of measuring the combined loss probability of a bidirectional transmission of a packet over a link in both directions, the Directional Airtime Metric measures the incoming loss rate and integrates the incoming linkspeed into the metric cost. There are multiple reasons for this decision:

- o OLSRV2 [[RFC7181](#)] defines the link metric as directional costs between routers.
- o Not all link layer implementations use acknowledgement mechanisms. Most link layer implementations who do use them use less airtime and a more robust modulation for the acknowledgement than the data transmission, which makes it more likely for the data transmission to be disrupted compared to the acknowledgement.
- o Incoming packet loss and linkspeed can be measured locally, symmetric link loss would need an additional signaling TLV in the [[RFC6130](#)] HELLO and would delay metric calculation by up to one HELLO interval.

The Directional Airtime Metric does not integrate the packet size into the link cost. Doing so is not feasible in most link-state routing protocol implementations. The routing decision of most operation systems don't take packet size into account. Multiplying all link costs of a topology with the size of a data-plane packet would never change the Dijkstra result anyways.





The queue based packet loss estimator has been tested extensively in the OLSR.org ETX implementation, see [Appendix B](#). The output is the average of the packet loss over a configured time period.

The metric normally measures the loss of a link by tracking the incoming [\[RFC5444\]](#) packet sequence numbers. Without these packet sequence numbers, the metric does calculate the loss of the link based of received and lost [\[RFC5444\]](#) HELLO messages. It uses the incoming HELLO interval time (or if not present, the validity time) to decide when a HELLO is lost.

When a neighbor router resets, its packet sequence number might jump to a random value. The metric tries to detect jumps in the packet sequence number and removes them from the data set, because the already gathered link loss data should still be valid (see [Section 9.3](#). The link loss data is only removed from memory when a Link times out completely and its Link Set tuple is removed from the database.

## 5. Metric Functioning & Overview

The Directional Airtime Metric is calculated for each link set entry, as defined in [\[RFC6130\] section 7.1](#).

The metric processes two kinds of data into the metric value, namely packet loss rate and link-speed. The link-speed is taken from an external process not defined in this document. The current packet loss rate is defined in this document by keeping track of packet reception and packet loss events. It could also be calculated by an external process with a compatible output.

Multiple incoming packet loss/reception events must be combined into a loss rate to get a smooth metric. Experiments with exponential weighted moving average (EWMA) lead to a highly fluctuating or a slow converging metric (or both). To get a smoother and more controllable metric result, this metric uses two fixed length queues to measure and average the incoming packet events, one queue for received packets and one for the estimated number of packets sent by the other side of the link.

Because the rate of incoming packets is not uniform over time, the queue contains a number of counters, each representing a fixed time interval. Incoming packet loss and packet reception event are accumulated in the current queue element until a timer adds a new empty counter to both queues and remove the oldest counter from both.

In addition to the packet loss stored in the queue, this metric uses a timer to detect a total link-loss. For every [\[RFC5444\]](#) HELLO



interval in which the metric received no packet from a neighbor, it scales the number of received packets in the queue based on the total time interval the queue represents compared to the total time of the lost HELLO intervals.

The average packet loss ratio is calculated as the sum of the 'total packets' counters divided by the sum of the 'packets received' counters. This value is then divided through the current link-speed and then scaled into the range of metrics allowed for OLSRv2.

The metric value is then used as `L_in_metric` of the Link Set (as defined in [section 8.1. of \[RFC7181\]](#)).

While this document does not add new [RFC5444](#) elements to the [RFC6130](#) HELLO or [RFC7181](#) TC messages, it works best when both the `INTERVAL_TIME` message TLV is present in the HELLO messages and when each [RFC5444](#) packet contains an interface specific sequence number. It also adds a number of new data entries to be stored for each [RFC6130](#) Link.

## 6. Protocol Constants

This specification defines the following constants, which define the range of metric values that can be encoded by the DAT metric (see Table 1). They cannot be changed without making the metric outputs incomparable and should only be changed for MANET's with a very slow or very fast link layer. See [Appendix D](#) [Appendix E](#) for example metric values.

`DAT_MAXIMUM_LOSS` - Fraction of the loss rate used in this routing metric. Loss rate will be between  $0/\text{DAT\_MAXIMUM\_LOSS}$  and  $(\text{DAT\_MAXIMUM\_LOSS}-1)/\text{DAT\_MAXIMUM\_LOSS}$ .

`DAT_MINIMUM_BITRATE` - Minimal bit-rate in Bit/s used by this routing metric.

+-----+-----+	
Name	Value
+-----+-----+	
<code>DAT_MAXIMUM_LOSS</code>	8
<code>DAT_MINIMUM_BITRATE</code>	1000
+-----+-----+	

Table 1: DAT Protocol Constants



## 7. Protocol Parameters

This specification defines the following parameters for this routing metric. These parameters are:

DAT\_MEMORY\_LENGTH - Queue length for averaging packet loss. All received and lost packets within the queue length are used to calculate the cost of the link.

DAT\_REFRESH\_INTERVAL - interval in seconds between two metric recalculations as described in [Section 10.2](#). This value SHOULD be smaller than a typical HELLO interval. The interval can be a fraction of a second.

DAT\_HELLO\_TIMEOUT\_FACTOR - multiplier relative to the HELLO\_INTERVAL (see [\[RFC6130\] Section 5.3.1](#)) after which the DAT metric considers a HELLO as lost.

DAT\_SEQNO\_RESTART\_DETECTION - threshold in number of missing packets (based on received packet sequence numbers) at which point the router considers the neighbor has restarted. This parameter is only used for packet sequence number based loss estimation. This number MUST be larger than DAT\_MAXIMUM\_LOSS.

### 7.1. Recommended Values

The proposed values of the protocol parameters are for Community Mesh Networks, which mostly use immobile routers. Using this metric for mobile networks might require shorter DAT\_REFRESH\_INTERVAL and/or DAT\_MEMORY\_LENGTH.

DAT\_MEMORY\_LENGTH := 64

DAT\_REFRESH\_INTERVAL := 1

DAT\_HELLO\_TIMEOUT\_FACTOR := 1.2

DAT\_SEQNO\_RESTART\_DETECTION := 256

## 8. Data Structures

This specification extends the Link Set of the Interface Information Base, as defined in [\[RFC6130\] section 7.1](#), by the adding the following elements to each link tuple:

L\_DAT\_received - a QUEUE with DAT\_MEMORY\_LENGTH integer elements. Each entry contains the number of successfully received packets within an interval of DAT\_REFRESH\_INTERVAL.



L\_DAT\_total - a QUEUE with DAT\_MEMORY\_LENGTH integer elements. Each entry contains the estimated number of packets transmitted by the neighbor, based on the received packet sequence numbers within an interval of DAT\_REFRESH\_INTERVAL.

L\_DAT\_packet\_time - the time when the next [RFC5444](#) packet should have arrived.

L\_DAT\_hello\_interval - the interval between two hello messages of the links neighbor as signaled by the INTERVAL\_TIME TLV [[RFC5497](#)] of NHDP messages [[RFC6130](#)].

L\_DAT\_lost\_packet\_intervals - the estimated number of HELLO intervals from this neighbor the metric has not received a single packet.

L\_DAT\_rx\_bitrate - the current bitrate of incoming unicast traffic for this neighbor.

L\_DAT\_last\_pkt\_seqno - the last received packet sequence number received from this link.

Methods to obtain the value of L\_DAT\_rx\_bitrate are out of the scope of this specification. Such methods may include static configuration via a configuration file or dynamic measurement through mechanisms described in a separate specification (e.g. [[DLEP](#)]). Any Link tuple with L\_status = HEARD or L\_status = SYMMETRIC MUST have a specified value of L\_DAT\_rx\_bitrate if it is to be used by this routing metric.

The incoming bitrate value should be stabilized by a hysteresis filter to improve the stability of this metric. See [Appendix B](#) [Appendix C](#) for an example.

This specification updates the L\_in\_metric field of the Link Set of the Interface Information Base, as defined in [section 8.1. of \[RFC7181\]](#))

### **8.1. Initial Values**

When generating a new tuple in the Link Set, as defined in [\[RFC6130\] section 12.5](#) bullet 3, the values of the elements specified in [Section 8](#) are set as follows:

- o L\_DAT\_received := 0, ..., 0. The queue always has DAT\_MEMORY\_LENGTH elements.
- o L\_DAT\_total := 0, ..., 0. The queue always has DAT\_MEMORY\_LENGTH elements.





- o L\_DAT\_packet\_time := EXPIRED (no earlier [RFC5444](#) packet received).
- o L\_DAT\_hello\_interval := UNDEFINED (no earlier NHDP HELLO received).
- o L\_DAT\_lost\_packet\_intervals := 0 (no HELLO interval without packets).
- o L\_DAT\_last\_pkt\_seqno := UNDEFINED (no earlier [RFC5444](#) packet with sequence number received).

## **9. Packets and Messages**

This section describes the necessary changes of [[RFC7181](#)] implementations with DAT metric for the processing and modification of incoming and outgoing [[RFC5444](#)] data.

### **9.1. Definitions**

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

- o "pkt\_seqno" is defined as the [[RFC5444](#)] packet sequence number of the received packet.
- o "interval\_time" is the time encoded in the INTERVAL\_TIME message TLV of a received [[RFC6130](#)] HELLO message.
- o "validity\_time" is the time encoded in the VALIDITY\_TIME message TLV of a received [[RFC6130](#)] HELLO message.

### **9.2. Requirements for using DAT metric in OLSRv2 implementations**

An implementation of OLSRv2 using the metric specified by this document SHOULD include the following parts into its [[RFC5444](#)] output:

- o an INTERVAL\_TIME message TLV in each HELLO message, as defined in [[RFC6130](#)] [section 4.3.2](#).
- o an interface specific packet sequence number as defined in [[RFC5444](#)] [section 5.1](#) which is incremented by 1 for each outgoing [[RFC5444](#)] packet on the interface.

An implementation of OLSRv2 using the metric specified by this document that inserts packet sequence numbers in some, but not all outgoing [[RFC5444](#)] packets will make this metric ignore all packets without the sequence number. Putting the INTERVAL\_TIME TLV into some, but not all Hello messages will make the timeout based loss



detection slower. This will only matter in the absence of packet sequence numbers.

### **9.3. Link Loss Data Gathering**

For each incoming [\[RFC5444\]](#) packet, additional processing SHOULD be carried out after the packet messages have been processed as specified in [\[RFC6130\]](#) and [\[RFC7181\]](#) as specified in this section.

[RFC5444] packets without packet sequence number MUST NOT be processed in the way described in this section.

The router updates the Link Set Tuple corresponding to the originator of the packet:

1. If L\_DAT\_last\_pkt\_seqno = UNDEFINED, then:
  1. L\_DAT\_received[TAIL] := 1.
  2. L\_DAT\_total[TAIL] := 1.
2. Otherwise:
  1. L\_DAT\_received[TAIL] := L\_DAT\_received[TAIL] + 1.
  2. diff := diff\_seqno(pkt\_seqno, L\_DAT\_last\_pkt\_seqno).
  3. If diff > DAT\_SEQNO\_RESTART\_DETECTION, then:
    1. diff := 1.
  4. L\_DAT\_total[TAIL] := L\_DAT\_total[TAIL] + diff.
3. L\_DAT\_last\_pkt\_seqno := pkt\_seqno.
4. If L\_DAT\_hello\_interval != UNDEFINED, then:
  1. L\_DAT\_packet\_time := current time + (L\_DAT\_hello\_interval \* DAT\_HELLO\_TIMEOUT\_FACTOR).
5. L\_DAT\_lost\_packet\_intervals := 0.

### **9.4. HELLO Message Processing**

For each incoming HELLO Message, after it has been processed as defined in [\[RFC6130\] section 12](#), the Link Set Tuple corresponding to the incoming HELLO message MUST be updated.



1. If the HELLO message contains an INTERVAL\_TIME message TLV, then:
  1. L\_DAT\_hello\_interval := interval\_time.
2. Otherwise:
  1. L\_DAT\_hello\_interval := validity\_time.
3. If L\_DAT\_last\_pkt\_seqno = UNDEFINED, then:
  1. L\_DAT\_received[TAIL] := L\_DAT\_received[TAIL] + 1.
  2. L\_DAT\_total[TAIL] := L\_DAT\_total[TAIL] + 1.
  3. L\_DAT\_packet\_time := current time + (L\_DAT\_hello\_interval \* DAT\_HELLO\_TIMEOUT\_FACTOR).

## **10. Timer Event Handling**

In addition to changes in the [[RFC5444](#)] processing/generation code, the DAT metric also uses two timer events.

### **10.1. Packet Timeout Processing**

When L\_DAT\_packet\_time has timed out, the following step MUST be done:

1. If L\_DAT\_last\_pkt\_seqno = UNDEFINED, then:
  1. L\_DAT\_total[TAIL] := L\_DAT\_total[TAIL] + 1.
2. Otherwise:
  1. L\_DAT\_lost\_packet\_intervals := L\_DAT\_lost\_packet\_intervals + 1.
3. L\_DAT\_packet\_time := L\_DAT\_packet\_time + L\_DAT\_hello\_interval.

### **10.2. Metric Update**

Once every DAT\_REFRESH\_INTERVAL, all L\_in\_metric values in all Link Set entries MUST be recalculated:

1. sum\_received := sum(L\_DAT\_received).
2. sum\_total := sum(L\_DAT\_total).



3. If `L_DAT_hello_interval`  $\neq$  `UNDEFINED` and `L_DAT_lost_packet_intervals`  $> 0$ , then:
  1. `lost_time_proportion` := `L_DAT_hello_interval` \* `L_DAT_lost_packet_intervals` / `DAT_MEMORY_LENGTH`.
  2. `sum_received` := `sum_received` \* `MAX ( 0, 1 - lost_time_proportion )`;
4. If `sum_received`  $< 1$ , then:
  1. `L_in_metric` := `MAXIMUM_METRIC`, as defined in [\[RFC7181\]](#) [section 5.6.1](#).
5. Otherwise:
  1. `loss` := `MIN(sum_total / sum_received, DAT_MAXIMUM_LOSS)`.
  2. `bitrate` := `MAX(L_DAT_rx_bitrate, DAT_MINIMUM_BITRATE)`.
  3. `L_in_metric` :=  $(2^{24} / \text{DAT\_MAXIMUM\_LOSS}) * \text{loss} / (\text{bitrate} / \text{DAT\_MINIMUM\_BITRATE})$ .
6. `remove(L_DAT_total)`
7. `add(L_DAT_total, 0)`
8. `remove(L_DAT_received)`
9. `add(L_DAT_received, 0)`

The calculated `L_in_metric` value should be stabilized by a hysteresis function. See [Appendix C](#) [Appendix D](#) for an example.

## **11. Security Considerations**

Artificial manipulation of metrics values can drastically alter network performance. In particular, advertising a higher `L_in_metric` value may decrease the amount of incoming traffic, while advertising lower `L_in_metric` may increase the amount of incoming traffic. By artificially increasing or decreasing the `L_in_metric` values it advertises, a rogue router may thus attract or repulse data traffic. A rogue router may then potentially degrade data throughput by not forwarding data as it should or redirecting traffic into routing loops or bad links. It might also attract traffic for "Man in the Middle" attacks or traffic analysis.





An attacker might also inject packets with incorrect packet level sequence numbers, pretending to be somebody else. This attack can be prevented by the true originator of the [RFC5444](#) packets by adding a [\[RFC7182\]](#) ICV Packet TLV and TIMESTAMP Packet TLV to each packet. This allows the receiver to drop all incoming packets which have a forged packet source, both packets generated by the attacker or replayed packets. The security mechanism described in [\[RFC7183\]](#) does not protect the additional sequence number of the DAT metric because it does only sign the [RFC5444](#) messages, not the [RFC5444](#) packet header.

Protection against "Man in the Middle" attacks are out of scope of this document.

## **[12.](#) Acknowledgements**

The authors would like to acknowledge the network administrators from Freifunk Berlin [[FREIFUNK](#)] and Funkfeuer Vienna [[FUNKFEUER](#)] for endless hours of testing and suggestions to improve the quality of the original ETX metric for the OLSR.org routing daemon.

This effort/activity is supported by the European Community Framework Program 7 within the Future Internet Research and Experimentation Initiative (FIRE), Community Networks Testbed for the Future Internet ([\[CONFINE\]](#)), contract FP7-288535.

The authors would like to gratefully acknowledge the following people for intense technical discussions, early reviews and comments on the specification and its components (listed alphabetically): Teco Boot (Infinity Networks), Juliusz Chroboczek (PPS, University of Paris 7), Thomas Clausen, Christopher Dearlove (BAE Systems Advanced Technology Centre), Ulrich Herberg (Fujitsu Laboratories of America), Markus Kittenberger (Funkfeuer Vienna), Joseph Macker (Naval Research Laboratory), Fabian Nack and Stan Ratliff (Cisco Systems).

## **[13.](#) References**

### **[13.1.](#) Normative References**

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [RFC 2119](#), [BCP 14](#), 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.



- [RFC5497] Clausen, T. and C. Dearlove, "Representing Multi-Value Time in Mobile Ad Hoc Networks (MANETs)", [RFC 5497](#), March 2009.
- [RFC6130] Clausen, T., Dearlove, C., and J. Dean, "Mobile Ad Hoc Network (MANET) Neighborhood Discovery Protocol (NHDP)", [RFC 6130](#), April 2011.
- [RFC7181] Clausen, T., Jacquet, P., and C. Dearlove, "The Optimized Link State Routing Protocol version 2", [RFC 7181](#), April 2014.

### **13.2. Informative References**

- [RFC3626] Clausen, T. and P. Jacquet, "Optimized Link State Routing Protocol", [RFC 3626](#), October 2003.
- [RFC7182] Ulrich, U., Clausen, T., and C. Dearlove, "Integrity Check Value and Timestamp TLV Definitions for Mobile Ad Hoc Networks (MANETs)", [RFC 7182](#), April 2014.
- [RFC7183] Ulrich, 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.
- [olsrv2\_paper] C., C., C., C., J., J., J., J., and H. H., "OLSRv2 for Community Networks: Using Directional Airtime Metric with external radios", Elsevier Computer Networks 2015 , September 2015, <<http://dx.doi.org/10.1016/j.comnet.2015.09.022>>.
- [CONFINE] "Community Networks Testbed for the Future Internet (CONFINE)", 2015, <<http://www.confine-project.eu>>.
- [DLEP] Ratliff, S., Berry, B., Harrison, G., Jury, S., and D. Satterwhite, "Dynamic Link Exchange Protocol (DLEP)", [draft-ietf-manet-dlep-17](#) , March 2013.
- [BATMAN] Neumann, A., Aichele, C., Lindner, M., and S. Wunderlich, "Better Approach To Mobile Ad-hoc Networking (B.A.T.M.A.N.)", [draft-wunderlich-openmesh-manet-routing-00](#) , April 2008.



[MOBICOM03]

De Couto, D., Aguayo, D., Bicket, J., and R. Morris, "A High-Throughput Path Metric for Multi-Hop Wireless Routing", Proceedings of the MOBICOM Conference , 2003.

[MOBICOM04]

Richard, D., Jitendra, P., and Z. Brian, "Routing in Multi-Radio, Multi-Hop Wireless Mesh Networks", Proceedings of the MOBICOM Conference , 2004.

[OLSR.org]

"The OLSR.org OLSR routing daemon", 2015,  
<<http://www.olsr.org/>>.

[FREIFUNK]

"Freifunk Wireless Community Networks", 2015,  
<<http://www.freifunk.net>>.

[FUNKFEUER]

"Austria Wireless Community Network", 2015,  
<<http://www.funkfeuer.at>>.

## **Appendix A. Future work**

As the DAT metric proved to work reasonable well for non- or slow-moving ad hoc networks [[olsrv2 paper](#)], it should be considered as a solid first step on a way to better MANET metrics. There are multiple parts of the DAT metric that need to be reviewed again in the context of real world deployments and can be subject to later improvements.

The easiest part of the DAT metric to change and test would be the timings parameters. A 1 minute interval for packet loss statistics might be a good compromise for some MANETs, but could easily be too large or too small for others. More data is needed to verify or improve the current parameter selection.

The DAT metric considers only the multicast [RFC5444](#) packet loss for estimating the link loss, but it would be good to integrate unicast data loss into the loss estimation. This information could be provided directly from the link layer. This could increase the accuracy of the loss rate estimation in scenarios, where the assumptions regarding the ratio of multicast vs. unicast loss do not hold.

The packet loss averaging algorithm could also be improved. While the DAT metric provides a stable sliding time interval to average the incoming packet loss and not giving the recent input too much



influence, However, first experiments suggest that the algorithm tends to be less agile detecting major changes of link quality. This makes it less suited for mobile networks. A more agile algorithm is needed for detecting major changes while filtering out random fluctuations regarding frame loss. However, the current "queue of counters" algorithm suggested for DAT outperforms the binary queue algorithm and the exponential aging algorithms used for the ETX metric in the OLSR [[RFC3626](#)] codebase of Olsr.org.

## **Appendix B.    OLSR.org metric history**

The Funkfeuer [[FUNKFEUER](#)] and Freifunk networks [[FREIFUNK](#)] are OLSR-based [[RFC3626](#)] or B.A.T.M.A.N. [[BATMAN](#)] based wireless community networks with hundreds of routers in permanent operation. The Vienna Funkfeuer network in Austria, for instance, consists of 400 routers covering the whole city of Vienna and beyond, spanning roughly 40km in diameter. It has been in operation since 2003 and supplies its users with Internet access. A particularity of the Vienna Funkfeuer network is that it manages to provide Internet access through a city wide, large scale Wi-Fi MANET, with just a single Internet uplink.

Operational experience of the OLSR project [[OLSR.org](#)] with these networks have revealed that the use of hop-count as routing metric leads to unsatisfactory network performance. Experiments with the ETX metric [[MOBICOM03](#)] were therefore undertaken in parallel in the Berlin Freifunk network as well as in the Vienna Funkfeuer network in 2004, and found satisfactory, i.e., sufficiently easy to implement and providing sufficiently good performance. This metric has now been in operational use in these networks for several years.

The ETX metric of a link is the estimated number of transmissions required to successfully send a packet (each packet equal to or smaller than MTU) over that link, until a link layer acknowledgement is received. The ETX metric is additive, i.e., the ETX metric of a path is the sum of the ETX metrics for each link on this path.

While the ETX metric delivers a reasonable performance, it doesn't handle well networks with heterogeneous links that have different bitrates. Since every wireless link, when using ETX metric, is characterized only by its packet loss ratio, the ETX metric prefers long-ranged links with low bitrate (with low loss ratios) over short-ranged links with high bitrate (with higher but reasonable loss ratios). Such conditions, when they occur, can degrade the performance of a network considerably by not taking advantage of higher capacity links.

Because of this the OLSR.org project has implemented the Directional Airtime Metric for OLSRv2, which has been inspired by the Estimated





Travel Time (ETT) metric [[MOBICOM04](#)]. This metric uses an unidirectional packet loss, but also takes the bitrate into account to create a more accurate description of the relative costs or capabilities of OLSRv2 links.

### [Appendix C.](#)   **Linkspeed stabilization**

The DAT metric describes how to generate a reasonable stable packet loss value from incoming packet reception/loss events, the source of the linkspeed used in this document is considered an external process.

In the presence of a layer-2 technology with variable linkspeed it is likely that the raw linkspeed will be fluctuating too fast to be useful for the DAT metric.

The amount of stabilization necessary for the linkspeed depends on the implementation of the mac-layer, especially the rate control algorithm.

Experiments with the Linux 802.11 wifi stack have shown that a simple Median filter over a series of raw linkspeed measurements can smooth the calculated value without introducing intermediate linkspeed values you would get by using averaging or an exponential weighted moving average.

### [Appendix D.](#)   **Packet loss hysteresis**

While the DAT metric use a sliding window to calculate a reasonable stable frame loss, the implementation might choose to integrate an additional hysteresis to prevent the metric flapping between two values.

In Section [Section 10.2](#) DAT calculates a fractional loss rate. The fraction of 'loss := sum\_total / sum\_received' may result in minor fluctuations in the advertised L\_in\_metric due to minimal changes in sum\_total or sum\_received which can cause undesirable protocol churn.

A hysteresis function applied to the fraction could reduce the amount of changes in the loss rate and help to stabilize the metric output.

### [Appendix E.](#)   **Example DAT values**

The DAT metric value can be expressed in terms of link speed (bit/s) or used airtime (s). When using the default protocol constants (see [Section 6](#)), DAT encodes link speeds between 119 bit/s and 2 Gbit/s.



Table Table 2 contains a few examples for metric values and their meaning as a link speed:

Metric	bit/s
MINIMUM_METRIC (1)	2 Gbit/s
MAXIMUM_METRIC (16776960)	119 bit/s
2000	1 Mbit/s

Table 2: DAT link cost examples

A path metric value could also be expressed as a link speed, but this would be unintuitive and difficult to understand. An easier way to transform a path metric value into a textual representation is to divide it by the hopcount of the path and express the path cost as average link speed together with the hopcount (see Table 3).

Metric	hops	average bit/s
4	2	1 Gbit/s
4000000	6	3 kbit/s

Table 3: DAT link cost examples

#### Authors' Addresses

Henning Rogge  
Fraunhofer FKIE

Email: [henning.rogge@fkie.fraunhofer.de](mailto:henning.rogge@fkie.fraunhofer.de)  
URI: <http://www.fkie.fraunhofer.de>

Emmanuel Baccelli  
INRIA

Email: [Emmanuel.Baccelli@inria.fr](mailto:Emmanuel.Baccelli@inria.fr)  
URI: <http://www.emmanuelbaccelli.org/>

