     Packet Sequence Number based directional airtime metric for OLSRv2
              draft-rogge-baccelli-olsrv2-ett-metric-04

Abstract

   This document specifies an directional airtime link metric for usage
   in OLSRv2.

Status of this Memo

Copyright Notice

Table of Contents

1. Introduction

   One of the major shortcomings of OLSR [RFC3626] is the missing of a
   link cost metric between mesh nodes. Operational experience with
   mesh networks gathered since the standardization of OLSR has revealed
   that wireless networks links can have highly variable and
   heterogeneous properties. This makes a hopcount metric insufficient
   for effective mesh routing.

   Based on this experience, OLSRv2 [OLSRV2] 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] 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 A). 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 routing metric for [RFC3626]. It
   takes both the loss rate and the link speed into account to provide a
   more accurate picture of the mesh network links.

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].

   The terminology introduced in [RFC5444], [OLSRV2] 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:

   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.

UNDEFINED - a value not in the normal value range of a variable.
   Might be -1 for this protocol.

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 transmit an IP packet over a link, not
   considering layer-2 overhead created by preamble, backoff time and
   queuing.

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.

3. Applicability Statement

   The Directional Airtime Metric was designed and tested in wireless
   IEEE 802.11 mesh networks. These networks employ link layer
   retransmission to increase the delivery probability and multiple
   unicast data rates.

   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.

If [RFC5444] control traffic is used to determine the link packet loss, the administrator should take care that link layer multicast transmission do not not have a higher reception probability than the slowest unicast transmission. It might be necessary to increase the data-rate of the multicast transmissions, e.g. set the multicast data-rate to 6 MBit/s if you use IEEE 802.11g only.

The metric can only handle a certain range of packet loss and unicast data-rate. Maximum packet loss is "ETX 4" (1 of 4 packets is successfully sent to the receiver, without link layer retransmissions), the unicast data-rate can be between 1024 Bit/s and 4 GBit/s. The metric has been designed for data-rates of 1 MBit/s and hundreds of MBit/s.

4. Directional Airtime Metric Rational

The Directional Airtime Metric has been inspired by the publications on the ETX [MOBICOM03] and ETT [MOBICOM04] metric, but has several key differences.

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 [OLSRV2] defines the link metric as directional costs between nodes.

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 A. The output is the
average of the packet loss over a configured time period.

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. While the link-speed is taken from
an external process, the current packet loss rate is calculated by
keeping track of packet reception and packet loss events.

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 NHDP 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 [OLSRV2]).

6. Protocol Parameters

   This specification defines the following parameters, which can be
   changed without making the metric outputs incomparable with each
   other:

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

   DAT_REFRESH_INTERVAL - interval in seconds between two metric
      recalculations as described in Section 11. This value SHOULD be
      smaller than a typical HELLO interval.

   DAT_HELLO_TIMEOUT_FACTOR - timeout factor for HELLO interval at
      which point a HELLO is definitely considered lost. The value must
      be a floating point number between 1.0 and 2.0, large enough to
      take the delay and jitter for message aggregation into account.

   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.

6.1. Recommended Values

   The proposed values of the protocol parameters are for Community Mesh
   Networks, which mostly use immobile mesh nodes. 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

7. Protocol Constants

   This specification defines the following constants, which cannot be
   changed without making the metric outputs incomparable:

   DAT_MAXIMUM_LOSS - Fraction of the loss rate used in this routing
      metric. Loss rate will be between 0/DAT_MAXIMUM_LOSS and
      (DAT_MAXIMUM_LOSS-1)/DAT_MAXIMUM_LOSS: 4.

   DAT_MINIMUM_BITRATE - Minimal bit-rate in Bit/s used by this routing
      metric: 1024.

8. Data Structures

   This specification extends the Link Set Tuples of the Interface
   Information Base, as defined in [RFC6130] section 7.1, by the
   following additional elements for each link tuple when being used
   with this metric:

   L_DAT_received is 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 is 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_hello_time is the time when the next hello will be expected.

   L_DAT_hello_interval is 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_hello_messages is the estimated number of lost hello
      messages from this neighbor, based on the value of the hello
      interval.

   L_DAT_rx_bitrate is the current bitrate of incoming unicast traffic
      for this neighbor.

   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.

   When using packet sequence numbers to estimate the loss rate, the
   Link Set Tuples get another field:

   L_DAT_last_pkt_seqno is the last received packet sequence number
      received from this link.

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_last_pkt_seqno := UNDEFINED (no earlier packet received).

   o L_DAT_hello_time := EXPIRED (no earlier NHDP HELLO received).

   o L_DAT_hello_interval := UNDEFINED (no earlier NHDP HELLO
      received).

   o L_DAT_lost_hello_messages := 0 (no HELLO interval without
      packets).

9. Packets and Messages

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.

9.2. Requirements

   An implementation of OLSRv2 using the metric specified by this
   document MUST 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.

9.3. Link Loss Data Gathering

   While this metric was designed for measuring the packet loss based on
   the [RFC5444] packet sequence number, some implementations might not
   be able to add the packet sequence number to their output.

9.3.1. Packet Sequence based link loss

   An implementation of OLSRv2, using the metric specified by this
   document with packet sequence based link loss, MUST include the
   following element into its [RFC5444] output:

   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.

   For each incoming [RFC5444] packet, additional processing MUST be
   carried out after the packet messages have been processed as
   specified in [RFC6130] and [OLSRV2].

   [RFC5444] packets without packet sequence number MUST NOT be
   processed in this way by this metric.

   The router MUST update 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 := seq_diff(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_hello_time := current time + (L_DAT_hello_interval *
       DAT_HELLO_TIMEOUT_FACTOR).

   5. L_DAT_lost_hello_messages := 0.

9.3.2. HELLO based Link Loss

   A metric might just use the incoming NHDP HELLO messages of a
   neighbor to calculate the link loss. Because this method uses fewer
   events to calculate the metric, the variance of the output will
   increase. It might be necessary to increase the value of
   DAT_MEMORY_LENGTH to compensate for this.

   For each incoming HELLO message, after it has been processed as
   defined in [RFC6130] section 12, the Link Set Tuple as defined in
   section 7.1 corresponding to the incoming HELLO message must be
   updated.

   1. L_DAT_received[TAIL] := L_DAT_received[TAIL] + 1.

   2. L_DAT_total[TAIL] := L_DAT_total[TAIL] + 1.

   3. L_DAT_lost_hello_messages := 0.

9.3.3. Other Measurement of Link Loss

   Instead of using incoming [RFC5444] packets or [RFC6130] messages,
   the routing daemon can also use other sources to measure the link
   layer lossrate (e.g. [DLEP]).

   To use a source like this with the DAT metric, the routing daemon has
   to add incoming total traffic (or the sum of received and lost
   traffic) and lost traffic to the queued elements in the extension of
   the Link Set Tuple defined in Section 8 corresponding to originator
   of the traffic.

   The routing daemon should also set L_DAT_lost_hello_messages to zero
   every times new packages arrive.

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.

   Only HELLO messages with an INTERVAL_TIME message TLVs must be
   processed.

   1. L_DAT_hello_interval := interval_time.

10. HELLO Timeout Processing

   When L_DAT_hello_time has timed out, the following step MUST be done:

   1. L_DAT_lost_hello_messages := L_DAT_lost_hello_messages + 1.

   2. L_DAT_hello_time := L_DAT_hello_time + L_DAT_hello_interval.

11. 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_total).

   2. sum_total := sum(L_DAT_received).

   3. If L_DAT_hello_interval != UNDEFINED and
       L_DAT_lost_hello_messages > 0, then:

       1. lost_time_proportion := L_DAT_hello_interval *
           L_DAT_lost_hello_messages / 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 [OLSRV2] section
           5.6.1.

   5. Otherwise:

       1. loss := sum_total / sum_received.

       2. If loss > DAT_MAXIMUM_LOSS, then:

           1. loss := DAT_MAXIMUM_LOSS.

       3. bitrate := L_DAT_rx_bitrate.

       4. If bitrate < DAT_MINIMUM_BITRATE, then:

1. bitrate := DAT_MINIMUM_BITRATE.

5. L_in_metric := (2^24 / DAT_MAXIMUM_LOSS) * loss / (bitrate / 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)

12. IANA Considerations

This document contains no actions for IANA.

13. 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.

An attacker might also inject packets with incorrect packet level sequence numbers, pretending to be somebody else. This attack could be prevented by the true originator of the RFC5444 packets by adding a [RFC6622] 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.

14. 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

15. References

15.1. Normative References

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

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

[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.

[RFC6622]   Ulrich, U. and T. Clausen, "Integrity Check Value and
            Timestamp TLV Definitions for Mobile Ad Hoc Networks
            (MANETs)", RFC 6622, May 2012.

[OLSRV2]   Clausen, T., Jacquet, P., and C. Dearlove, "The Optimized
            Link State Routing Protocol version 2",
            draft-ietf-manet-olsrv2-19 , March 2013.

15.2. Informative References

[CONFINE]   "Community Networks Testbed for the Future Internet
            (CONFINE)", 2013, <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-04 , March 2013.

   [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", 2013,
                 <http://www.olsr.org/>.

   [FREIFUNK]
                 "Freifunk Wireless Community Networks", 2013,
                 <http://www.freifunk.net>.

   [FUNKFEUER]
                 "Austria Wireless Community Network", 2013,
                 <http://www.funkfeuer.at>.

Appendix A. OLSR.org metric history

   The Funkfeuer [FUNKFEUER] and Freifunk networks [FREIFUNK] are OLSR-
   based [RFC3626] or B.A.T.M.A.N. based wireless community networks
   with hundreds of routers in permanent operation. The Vienna
   Funkfeuer network in Austria, for instance, consists of 400 routers
   (around 600 routes) 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 mesh network, 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 mesh links.

Authors' Addresses

Henning Rogge
Fraunhofer FKIE

Email: henning.rogge@fkie.fraunhofer.de
URI: http://www.fkie.fraunhofer.de

Emmanuel Baccelli
INRIA

Email: Emmanuel.Baccelli@inria.fr
URI: http://www.emmanuelbaccelli.org/