Multimetric OLSR and ETT
Case Study Final Report

Multimetric OLSR and ETT

Andreas Lavén Department of Computer Science
Karlstad University
Karlstad, Sweden
Email: a.laven@gmail.com

Peter Hjärtquist Department of Computer Science
Karlstad University
Karlstad, Sweden
Email: peter@hjartquist.se

Abstract
This report gives an overview on an extension to Olsrd, which is an implementation of the routing protocol OLSR, and the integration of the Expected Transmission Time metric, ETT. The extension provides capabilities to distribute an arbitrary number of routing metrics. The extension is called Multimetric OLSR, and its design and implementation will be explained in this report. An explanation of how it is used is also given.

Index Terms
Routing, Metrics, Mesh Networks

1 Introduction

In wireless mesh network the standard metric, which is minimum hop count, is not sufficient for use as routing metric. The reason is that when using the minimum hop count metric, slow or highly loaded links may be used since all links are assumed to be of equal capacity. Using slow links may incur a noticeable decrease of the throughput, whereas a better suited metric would indicate that these links are sub-optimal and another route can be used instead possibly yielding a higher throughput. Examples of such metrics are Expected Transmission Count (ETX) and Expected Transmission Time (ETT), which consider the quality of the links and other information.

There is a framework for Olsrd called MF-OLSR[3][4] providing the possibility to use other metrics than hop count. The extended message format used in MF-OLSR may contain an arbitrary number of metrics, however, the implementation does not. MF-OLSR distributes a total of three metrics, one metric that is configurable and two fixed metrics. The fixed metrics are link quality (LQ) and neighbor link quality (NLQ). LQ is the measured link quality from the current node to its neighbor and NLQ is the link quality in the opposite direction. These two metrics can be used to estimate the ETX value. The third metric can be separate from LQ and NLQ, such as the cost metric, and may be used for calculating the routing table.

The proposed Multimetric OLSR is an extension to Olsrd which is based on MF-OLSR. The major advantage of Multimetric OLSR in addition to MF-OLSR is the possibility to distribute an arbitrary number of metrics. Each node broadcasts its metric information to its one hop neighbors, which will broadcast the information further and eventually each node in the network has received the information. MF-OLSR only provides the capability to distribute one configurable metric. Because of the feature of distributing more metrics, Multimetric OLSR gives the possibility for routing algorithms to be more advanced. It is also possible to use Multimetric OLSR to distribute values other than metrics used for routing table calculations, such as channel information which could be used for a channel assignment algorithm. It is also possible to distribute metrics used for calculations other than routing.

The goal of this work is to extend Olsrd to support multiple metrics, and to integrate the ETT metric into this framework.

2 Background

In this section, two metrics are introduced. These metrics are ETX and ETT.

2.1 ETX

ETX, Expected Transmission Count, is an attempt to create a metric better suited for wireless networks. The metric represents the number of times a node expects to send or resend a packet for the receiving node to receive a non-corrupt packet. However, the metric does not consider link speed or available bandwidth. ETX is calculated as

$$ETX = \frac{1}{LQ * NLQ} \tag{1}$$

Where LQ is the link quality and NLQ is the neighbor link quality.

2.2 ETT

ETT, Expected Transmission Time, is a metric that is built on ETX. ETT tries to estimate the time it would take to send a packet on a link by multiplying the ETX with the size of the packet divided by the link capacity as shown in Equation (2). This gives the time it should take to send a packet including the resends if needed.

ETT is calculated using the following algorithm.

$$ETT = ETX * \frac{S}{B} \tag{2}$$

Here, S is the packet size, which is measured in bytes and B is the link capacity, which is measured using packet-pair technique [5]. The packet-pair technique uses two back-to-back probes, one small packet (defaults to 137 bytes) followed by a larger (defaults to 1137 bytes) which are sent unicast to every neighbor. The receiving node measures the inter-arrival time and reports it back to the sending node. Then the link capacity is estimated by Equation (3):

$$B = S_L / \min_{1 \ldots n} d_i \tag{3}$$

where B is the link capacity, S_L is the size of the large probe packet, d_i is a delay sample and n is a predefined number of delay samples. The ETT metric has not been implemented in this project, but an already existing implementation[5] has been adapted to fit with the multimetric framework. During this project, only few changes to the ETT-plug-in have been made, like general adaptations and fixes for a newer Olsrd version and the way metric values are set. These changes are done to make the ETT-plug-in match the Multimetric OLSR structure.

3 Design

The major functionality of the Multimetric OLSR extension is distribution of metrics. The Multimetric OLSR is able to distribute an arbitrary number of metrics. This is done by extending the Hello and TC messages by a list of metrics. There are different kinds of metrics. These metrics can be handled differently by the Multimetric OLSR. For example, it is possible to provide both a configured cost value and metrics measured during run-time. ETX and ETT are examples of metrics that are measured during run-time.

Even though an arbitrary number of metrics can be distributed by Multimetric OLSR, only one metric can be used for the routing table calculation. The metric which is used for routing is decided during compile-time. Which metrics to distribute is also decided during compile-time. If the metric used for the routing table calculation is not distributed from a node in the network, a conservative default value will be used instead. A conservative default value is used in order to maintain the nodes connectivity, while having a low priority. This is useful when it is the only node providing a unique feature in the network, such as Internet access.

The ETT metric is integrated into Multimetric OLSR by having the implementation for ETT as an Olsrd plug-in.

4 Implementation

The Multimetric OLSR is built upon Olsrd 0.5.6-r4. One major advantage with that version compared to the Olsrd version MF-OLSR is based on the *cost* variable. This variable carries the values used for routing table calculations. Assigning a value from a metric to the cost variable makes that value used during routing table calculation.

Multimetric OLSR is implemented by copying and mixing the original Olsrd packet parser and generating functions with the MF-OLSR functions. Thereby, the functionality of MF-OLSR is adapted

to the new structure of the newer version of Olsrd. The original packet parser and generating functions are disabled and only the Multimetric OLSR functions are used.

4.1 Message Formats

The message formats used in Multimetric OLSR are the same as is used in MF-OLSR. Multimetric OLSR should be able to co-exist with MF-OLSR in a network because of the common message formats. However, the two implementations might not handle the distributed metrics in the same way. The message formats are not equal to the Olsrd messages, thus it is a bad idea to use the original Olsrd and any of its multimetric frameworks at the same time in the same network. Due to the different message formats, Olsrd will interpret metrics as addresses and the multimetric frameworks will interpret addresses in Olsrd packets as metrics.

Olsrd uses three different kinds of messages. These are Hello, Topology Control (TC) and Host and Network Association (HNA) messages. The Hello messages are used to announce existence of a node and which neighbors it has. The TC messages are used to disseminate neighbor information throughout the network. The HNA messages are used to announce gateways and their reachable networks. Multimetric OLSR uses the same types of messages but not the same message formats.

The message formats used are provided in standard IETF message format style in appendix. Each message type is recognized by its unique message type number. The Hello messages are type 203 and the first 3 bytes of the message format are the same as in OLSR. For each Neighbor Interface Address there can be an arbitrary number of metrics. The number of metrics associated with an address is calculated by subtracting 4, the size of an IPv4 address, from the Link Message Size and then dividing the result by 4 which is the size of a serialized metric. Each metric consists of 16 bits value and 8 bits type. The remaining 8 bits are reserved for future use.

TC messages are type 204. These messages also start with the same structure as OLSR but only the first 2 bytes. For each Advertised Neighbor Main Address there can be an arbitrary number of metrics. Each metric consists of 16 bits value, 8 bits More metrics and 8 bits type. The More metric field is used to flag for more metrics to come. If the More metric field is set to 0, there are no more metrics in the list. A More metric field set to 1 indicates there is at least one more metric in the metric list.

HNA messages are type 205 and look a bit different from the other messages. The main difference is that the HNA messages can only contain a single metric per announced network. For each announced network there is one gateway metric of the specified gateway metric type. The gateway metric represents how good the gateway is for routing to that network in contrast to the other metrics which represents how good a link between two nodes are. The last field is reserved for future use.

5 Usage

Instructions on how to configure and extend the Multimetric OLSR is given in this section. All configurations must be done before compiling Multimetric OLSR.

5.1 Add a Metric

Metrics are categorized in two categories. One category is for internal metrics, handled internally by Multimetric OLSR and the other category, plug-in metrics, is for metrics whose code is placed in an Olsrd plug-in. Because of the different placements of the code, the procedure for adding a metric differs. There are two recommended changes that are common for both categories. All metrics require a unique identifier and it is recommended to add the identifier in mf_defs.h by adding a statement similar to

```
#define LMT_ETX 5
```

which defines the identifier for the ETX metric. The identifier is used to identify what the metric type the corresponding value corresponds to. The identifier should be known to all nodes in the network. LMT_ETX is an abbreviation for Link Metric Type Expected Transmission Count and is merely a textual representation of the identifier for the metric and is used for convenience in the source code.

While printing the metric table on screen, Multimetric OLSR needs to know the name of all the used metrics in order to get a nice output. Making the name known is done by adding another case statement in the switch statement in the function mf_olsr_lmt_to_string in the file named mf_olsr_main.c similar to

```
case (LMT_ETX): return "ETX";
```

which states the name of the ETX metric.

Internal Metric

Internal metrics, such as cost and ETX, are handled solely by Multimetric OLSR. All code calculating the metric values must be put inside the Olsrd code. It is suggested to put the code for calculating metric values in the function mf_olsr_update_own_tc_metrics in mf_olsr_main.c.

External Metric

External metrics are handled solely by a plug-in providing the metric specific code. Plug-ins calculate the metric values all by themselves in any way they want and sets the cost variable.

5.2 What Metrics to Distribute

Specifying what metrics to distribute is simply accomplished by adding the identifying integer (or preferably, a #define value) corresponding the desired metric to the array named active_lmt located at the top of mf_olsr_main.c. The active_lmt array may look like

```
int active_lmt[] = {LMT_ETX, LMT_LQ, LMT_NLQ, -1};
```

which defines the ETX, LQ and NLQ metrics to be distributed. Though it is possible, it is not recommended to remove the link quality (LQ) or neighbor link quality (NLQ) metrics as it will break the expected transmission count (ETX) metric. The array must absolutely end with the value -1 to indicate that there are no more metrics. Using a multimetric framework such as Multimetric OLSR implies usage of metrics hence Multimetric OLSR requires that metrics are used. Thus, having an active metrics list of only -1 (or completely empty) results in undefined behavior, most likely incorrect messages and probably crashes. The array must be set before compiling MultiMetric OLSR.

The available metrics for distribution are currently LQ, NLQ, ETX, ETT and COST, where COST is a configured value representing how willing an interface is to send a packet.

5.3 What Routing Metric to Use

It is possible to distribute several metrics, however, it is only possible to use one metric for calculating the routing tables. The routing table calculation is performed using the original Olsrd routing table calculation algorithm and code. Specifying what metric to use is done by, before compiling, changing the #define value called DEF_LINK_METRIC_TYPE which is located in mf_defs.h. Setting the routing metric to ETX makes the define value look like

```
#define DEF_LINK_METRIC_TYPE LMT_ETX
```

5.4 Configuration File Parameters

Multimetric OLSR can be configured using a few configuration file parameters. The most important configuration file parameter is *MetricFrameworkLevel* which enables or disables the multimetric framework. Setting it to a positive integer enables the multimetric framework and setting it to zero disables it. In MultiMetric OLSR, there is only one more parameter and it sets the cost per interface for using with the Cost metric.

The relevant parts of the configuration file may look like

```
 MetricFrameworkLevel 1
Interface "ath0"
{
Cost 10
}
```

where MetricFrameworkLevel set to 1 enables the framework and Cost set to 10 defines that the cost for sending a packet on the interface named ath0 is 10.

6 Discussion

The Multimetric OLSR can be used for many purposes. One purpose is to improve routing algorithms through better metrics. By using Multimetric OLSR, more advanced routing algorithms can be used, for example routing algorithms using primary and secondary metrics. Another example is routing algorithms using metrics depending on other metrics, such as ETT which depends on ETX.

There is no limitations for the Multimetric OLSR only to distribute metrics used for routing. All metrics used by plug-ins can be distributed by Multimetric OLSR.

One other possible application of Multimetric OLSR is to distribute channel information. This is useful in multi-radio multi-channel wireless networks where each node may have multiple wireless interfaces. The usage of multiple channels may increase the capacity of the network, due to the possibility for several nodes to send packets simultaneously without interfering with each other. In order to increase the capacity, a channel assignment algorithm is needed to make nodes select different channels for their receiving interface.

For example, Multimetric OLSR could be used in conjunction with the bonding module provided by Net-X [6], which is prepared for a multi-radio multi-channel environment. To increase the capacity of the network, the nodes within a local environment should not use the same channel at their respective listening interface because that would force sending nodes to use the same channel. This clash prevents multiple nodes to transmit simultaneously which would be possible if different orthogonal channels were assigned to each listening interface. The Multimetric OLSR can be used to distribute which channel is used at the listening interface.

7 Conclusions

With the Multimetric OLSR it is possible to distribute an arbitrary number of metrics in Olsrd. The possibility to distribute more metrics can have many advantages. When more metrics are available for nodes, more advanced routing algorithms can be used to select the best path.

By using the ETT metric provided by the ETT plug-in, the routing algorithm considers not only the link quality, but also the capacity of the link.

It is also possible to use Multimetric OLSR to distribute other metrics than for the routing algorithm. For an example, channel information can be distributed for a channel assignment algorithm.

Future work for this project could be to extend the HNA message structure so it may contain multiple metrics. Also, this extension only supports IPv4 hence support for IPv6 is a feasible feature. Moreover, it is only possible to configure the Multimetric OSLR before compilation. It is feasible to be able to configure Multimetric OLSR using the Olsrd configuration file.

References

[1] UniK OLSR daemon software, Available: http://www.olsr.org/ [Accessed Sept. 16, 2009].

[2] Thomas Clausen and Philippe Jacquet, "Optimized Link State Routing Protocol (OLSR)", IETF RFC 3626, Oct. 2003, Available: http://www.ietf.org/rfc/rfc3626.txt [Accessed Sept. 17, 2009].

[3] Thomas Aure, "Metric-based routing framework and radio-aware shortest path selection for OLSR", May 2008, Available: http://www.velocitynetworks.org/olsr/master-thomasau.pdf [Accessed Sept. 16, 2009].

[4] "MF-OLSR source code archive", http://www.velocitynetworks.org/olsr/eolsrd-0.5.zip [Accessed Sept. 16, 2009].

[5] Pedro Esposito, Miguel Elias Mitre Campista, Igor Moraes, Marcelo Rubinstein, Universidade do Estado do Rio de Janeiro, Luis Henrique Maciel Kosmalski Costa, Otto Duarte, "Implementing the Expected Transmission Time Metric for OLSR Wireless Mesh Networks", Available: http://www.gta.ufrj.br/ftp/gta/TechReports/ECM08.pdf [Accessed Sept. 16, 2009].

[6] Pradeep Kyasanur, Chandrakanth Chereddi and Nitin H. Vaidya, "Net-X: System eXtensions for Supporting Multiple Channels, Multiple Interfaces and Other Interface Capabilities", August 2006, Available: http://www.crhc.illinois.edu/wireless/papers/kyasanur2006Tech.pdf [Accessed Sept. 16, 2009].