

Mobile Ad hoc Networking (MANET) C. Dearlove
Internet-Draft BAE Systems Advanced Technology
Intended status: Informational Centre
Expires: March 20, 2009 T. Clausen
LIX, Ecole Polytechnique, France
P. Jacquet
INRIA, France
September 16, 2008

Link Metrics for OLSRv2
draft-dearlove-olsrv2-metrics-03

Status of This Memo

By submitting this Internet-Draft, each author represents that any applicable patent or other IPR claims of which he or she is aware have been or will be disclosed, and any of which he or she becomes aware will be disclosed, in accordance with Section 6 of BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

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."

The list of current Internet-Drafts can be accessed at
<http://www.ietf.org/ietf/lid-abstracts.txt>.

The list of Internet-Draft Shadow Directories can be accessed at
<http://www.ietf.org/shadow.html>.

This Internet-Draft will expire on March 20, 2009.

Internet-Draft

OLSRv2 Link Metrics

September 2008

Abstract

This document describes how link metrics may be added, in a relatively straightforward manner, to the specification of OLSRV2, in order to allow routing by other than minimum hop count routes. In addition to metric signaling and use, the most significant change is a separation of the routing and flooding functions of MPRs.

Internet-Draft

OLSRv2 Link Metrics

September 2008

Table of Contents

1. Introduction	4
2. Applicability	6
3. Motivational Scenarios	7
4. Link Metrics	9
4.1. Link Metric Types	9
4.2. Directional Link Metrics	11
4.3. Reporting Link Metrics	12
4.4. Defining Link Metrics	13
4.5. Link Metric TLVs	13
4.6. Link Metric Values	15
5. MPRs with Link Metrics	17
5.1. Flooding MPRs	17
5.2. Routing MPRs	19
5.3. Relationship Between MPR Sets	21
6. Implementation	24
6.1. Parameters and Constants	25
6.2. Local Information Base	25
6.3. Interface Information Base	26
6.4. Node Information Base	27
6.5. Topology Information Base	27
6.6. Processing and Information Base	28
6.7. Metric Representation	28
6.8. MPR Representation	29
6.9. HELLO Message Generation	30
6.10. HELLO Message Processing	30
6.11. MPR Calculation and Neighbor Set Update	31
6.12. TC Message Generation	32
6.13. TC Message Processing	32
6.14. Routing Set Calculation	32
7. IANA Considerations	34
8. Security Considerations	35
9. References	36
9.1. Normative References	36
9.2. Informative References	36
Appendix A. MPR Routing Property	37
Appendix B. Routing MPR Calculation	39
Appendix C. Example Algorithm for Calculating the Routing Set	42
C.1. Add Local Symmetric Links	42
C.2. Add Remote Symmetric Links	43
C.3. Add Attached Networks	45
Appendix D. Constraints	46
Appendix E. Acknowledgements	48

Internet-Draft

OLSRv2 Link Metrics

September 2008

1. Introduction

The Optimized Link State Routing Protocol [OLSRv2] is a proactive routing protocol for Mobile Ad hoc NETWORKS (MANETs) [RFC2501]. In its current form, this protocol finds shortest, defined as minimum number of hops, routes from a node to all possible destinations.

However limiting to minimum hop routes may yield what are, to the user, inferior routes. Some examples are given in Section 3. This limitation is not, however, fundamental to OLSRv2. First, the extensible message format [packetbb] used by OLSRv2 naturally permits the addition of additional information regarding links to OLSRv2 messages. Second, OLSRv2 essentially first collects topological information from the network and then forms minimum length routes. Using a definition of route length (metric) other than number of hops is a natural extension commonly used in link state protocols.

Addition of alternative route selection processes to OLSRv2 could be treated as a possible future extension. However in this case, legacy OLSRv2 nodes, which would not recognize any additional link information, would still attempt to use minimum hop routes. This would mean that, in effect, nodes differed over their valuation of links and routes. This can lead to the fundamental routing problem of "looping", and must be avoided. Thus if alternative route selection were to be considered only as a future extension, then nodes which did, and nodes which did not, implement the extension could not interoperate. This would be a significant limitation of such an extension.

This document discusses a possible extension to OLSRv2 which could be fairly straightforwardly incorporated in a revision of [OLSRv2]. The principal changes to OLSRv2 which this extension involves are:

- o Assigning metrics to links. This involves considering separate metrics for the two directions of a link, with the receiving node determining the metric in that direction. Directional metrics must be signaled in HELLO messages, and are also included in TC messages.
- o Metrics to be used in OLSRv2 are dimensionless and additive. The assignment of metrics, including their relationship to real parameters such as bandwidth, loss rate and delay, is outside the scope of OLSRv2, which simply would use these metrics in a consistent manner. However by use of a registry of metric types, used in the extended type of a single link metric TLV, nodes can use only metrics of the type that they are configured to use.

Internet-Draft

OLSRv2 Link Metrics

September 2008

- o The separation of the two functions performed by MPRs in OLSRV2, optimized flooding and reduced topology advertisement for routing, into separate sets of MPRs, denoted flooding MPRs and routing MPRs. Flooding MPRs can be calculated as MPRs currently are (but can improve the selection using metrics) while routing MPRs need a metric-aware selection algorithm, an example of which is given in this document. This guarantees the use of minimum distance routes using the chosen metric, while still using only two hop neighborhood information from HELLO messages and routing MPR selector information in TC messages.
- o Appropriate changes to protocol Information Bases, messages (new metric and modified MPR TLVs) and message processing. These are described in some detail in this document.

Internet-Draft

OLSRv2 Link Metrics

September 2008

2. Applicability

The objective of this document is to serve as a basis for discussion as to whether such a revision of [OLSRv2], to form part of the basic OLSRv2 specification, is desirable, or even essential for the widespread adoption of OLSRv2 that is its objective. None of the changes proposed in this document affect any of the other constituent parts of OLSRv2, in particular they do not affect [NHDP], since as some uses of that protocol will not need metrics, they should not have metrics imposed on them.

The addition of metrics in this way to OLSRv2 would form a mandatory part of the specification. An implementation that is to interwork with all other implementations of OLSRv2, subject to any administrative configuration of choice of metric type, MUST fully implement the use of link metrics.

Internet-Draft

OLSRv2 Link Metrics

September 2008

3. Motivational Scenarios

The basic situation that suggests the desirability of use of routes other than minimum hop routes is shown in Figure 1.

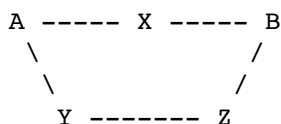


Figure 1

The minimum hop route from A to B is via X. However if the links A to X and X to B are poor (e.g. having low bandwidth or being unreliable) but the links A to Y, Y to Z and Z to B are better (e.g. having reliable high bandwidth) then the route A to B via Y and Z may be preferred.

There are other situations where even if links do not show immediately obvious benefits to users, their use should be discouraged. Consider a network with many short range links, and a few long range links. Use of minimum hop routes will immediately lead to heavy use of the long range links. This will be particularly undesirable if those links achieve their longer range through reduced bandwidth or through being less reliable. However, even if the long range links have the same characteristics as the short range links, it may be better to reserve usage of the long range links for when this usage is particularly valuable - for example when the use of one long range link saves several short range links (rather than the single link that is all that is needed to be saved for a minimum hop route).

A related case is that of a privileged relay node. An example is an aerial node in an otherwise ground based network. The aerial node may have a link to many, or even all, other nodes. That would lead to all nodes attempting to send all their traffic (other than to immediate neighbors and some two hop neighbors) via the aerial node. It may however be important to reserve that capacity for cases where the aerial node is actually essential, such as if the ground based portion of the network is disconnected.

Other cases may involve attempts to avoid areas of congestion, to route around insecure nodes (by preference, but prepared to use them if there is no other alternative) and nodes attempting to discourage their use as relays due to, for example, limited battery power. OLSRV2 does have another mechanism to aid in this, a node's willingness to act as an MPR. However there are cases where that cannot help, but where use of non-minimum hop routes could.

Internet-Draft

OLSRv2 Link Metrics

September 2008

Similarly note that OLSRV2's optional use of link quality (through its use of [NHDP]) is not a solution to these problems, use of link quality as so specified allows a node to decline to use a link, not only on its own, but on all nodes' behalf. It does not allow the use of a link, for example, if all else fails. Note that it is not suggested that use of non-minimum hop routes will replace these mechanisms, there is a place for each used separately or together.

It should also be noted that the loop-free property of OLSRV2, and of this modification, apply strictly only in the static state. When the network topology is changing, and with possibly lossy messages, it is possible for transient loops to form, but with update rates appropriate to the rate of topology change these are sufficiently rare. Changing link metrics is a form of network topology change, and should be limited to a rate slower than the message information update rate (defined by the parameters HELLO_INTERVAL, HELLO_MIN_INTERVAL, TC_INTERVAL and TC_MIN_INTERVAL).

Internet-Draft

OLSRv2 Link Metrics

September 2008

4. Link Metrics

Using the approach suggested here, link metrics will be:

- o As used by OLSRV2, dimensionless. However they may, directly or indirectly, correspond to specific physical information (such as delay, loss rate or bandwidth) but this knowledge will not be used by OLSRV2. Instead, generating the metric value will be the responsibility of a mechanism external to OLSRV2.
- o Additive, so that the metric of a route is the sum of the metrics of the links forming that route. Note that this requires a metric where a low value of a link metric indicates a "good" link and a high value of a link metric indicates a "bad" link, where the former will be preferred to the latter.
- o Directional, the metric from node A to node B need not be the same as the metric from node B to node A, even when using the same interfaces.

These aspects of link and route metrics are discussed in the following sections.

4.1. Link Metric Types

There are various physical characteristics that may be used to define a link metric. Some examples, which also illustrate some characteristics of metrics that result, are:

- o Delay is a straightforward metric, as it is naturally additive, the delay of a multi-link route is the sum of the delays of the links. (This does not directly take into account delays due to nodes, rather than links, but these can be divided among incoming and outgoing links.) However given a limited range of link metric value (as must be used) more than one type of delay metric may be required, representing different ranges of delay value.
- o Probability of loss on a link is, as long as probabilities of loss are small and independent, approximately additive. (A slightly more accurate approach is a negatively scaled logarithm of the probability of not losing a packet.) If losses are not independent then this will be pessimistic. Again, more than one range of values (or, equivalently, more than one scaling of the logarithms) may be needed.
- o Bandwidth is not additive, it even has the wrong characteristic of being good when high, bad when low; thus a mapping that inverts its ordering must be applied to it. Such a mapping can, at best,

Internet-Draft

OLSRv2 Link Metrics

September 2008

only produce a metric that it is acceptable to treat as additive. Consider, for example, a preference for a route that maximizes the minimum bandwidth link on the route, and then prefers a route with the fewest links of each bandwidth from the lowest. If links may be of three discrete bandwidths, "high", "medium" and "low", then this preference can be achieved, on the assumption that no route will have more than 10 links, with metric values of 1, 10 and 100 for the three bandwidths. If routes can have more than 10 links, the range of metrics must be increased; this indicates a preference for a wide "dynamic range" of link metric values. Depending on the ratios of the numerical values of the three bandwidths, the same effect may be achieved by using a scaling of an inverse power of the numerical values of the bandwidths. For example if the three bandwidths were 2, 5 and 10 Mbit/s, then a possible mapping would be the fourth power of 10 Mbit/s divided by the bandwidth, giving metric values of 625, 16 and 1 (good for up to 16 links in a route). This mapping can be extended to a system with more bandwidth values, for example giving a 4 Mbit/s bandwidth a metric value of about 39. This may lose the capability to produce an absolutely maximum minimum bandwidth route, but will usually produce either that, or something close (and at times maybe better, is a route of three 5 Mbit/s links really better than one of a single 4 Mbit/s link?) Specific metrics will need to define the mapping (e.g. a power and bandwidth scaling).

There are also many other possible metrics, including physical layer information (such as signal to noise ratio, and error control statistics) and information such as interface queue statistics.

In a well-designed network, all nodes will use the same metric type. It will not produce good routes if, for example, some link metrics are based on bandwidth and some on path loss (except to the extent that these may be correlated).

How to achieve this is an administrative matter, outside the scope of OLSRV2. In fact even the actual physical meanings of the metrics will be outside the scope of OLSRV2. This is because new metrics may be added in the future, for example as bandwidths increase, or even adding new concepts (perhaps a metric may be based on financial cost). Each such type will have a metric type number (whose range is considered later). Initially a single link metric type zero will be defined as indicating a dimensionless metric with no predefined physical meaning.

An OLSRV2 node will then be instructed which single link metric type to use and recognize, without knowing whether it represents delay, probability of loss, bandwidth, cost or any other quantity. This

Internet-Draft

OLSRv2 Link Metrics

September 2008

recognized link metric type number will be a node parameter, and subject to change in case of reconfiguration, or possibly the use of a protocol (outside the scope of OLSRV2) permitting a process of link metric type agreement between nodes.

The use of link metric type numbers also suggests the possibility of use of multiple link metric types and multiple network topologies. This is a possible future extension to OLSRV2, but is not included in this proposal. To allow for that future possibility, the sending of more than one metric, of different physical types, which should not be done for reasons of efficiency, will however not be forbidden, but other types than that configured will be ignored.

The following three sections assume a chosen single link metric type, of unspecified physical nature. The selection of that type is described in Section 4.5.

4.2. Directional Link Metrics

OLSRv2 uses only "symmetric" (bidirectional) links, which may pass traffic in either direction. A key decision is whether these links should each be assigned a single metric, used in both directions, or a metric in each direction, noting that:

- o Links can have different characteristics in each direction, use of directional link metrics recognizes this.
- o In many (possibly most) cases, the two ends of a link will naturally form different views as to what the link metric should be. To use a single link metric requires a coordination between the two that can be avoided if using directional links. Note that if using a single metric, it would be essential that the two ends agree as to its value, otherwise it is possible for looping to occur. This problem does not occur for directional metrics.

Based on these considerations, directional metrics are preferred. Each node must thus be responsible for defining the metric in one direction only. This could be in either direction, i.e. that a node is responsible for either incoming or outgoing link metrics, as long as the choice is universal. The former (incoming) case is used because, in general, receiving nodes have more information available to determine link metrics (for example received signal strength, interference levels and error control coding statistics).

Note that, using directional metrics, if node A defines the metric of the link from node B to node A, then node B must use node A's definition of that metric on that link in that direction. (Node B could, if appropriate, use a bad mismatch between directional metrics

Internet-Draft

OLSRv2 Link Metrics

September 2008

as a reason to discontinue use of this link, using the link quality mechanism in [NHDP].)

4.3. Reporting Link Metrics

Links, and hence link metrics, will be reported in HELLO messages and TC messages. A node must report the incoming directional metric in its HELLO messages in order that this is available at the other end of the link. This will mean that, for a bidirectional link, both ends of the link will know both metrics.

To fit the OLSRv2 model, node A must be responsible for advertising the metric from node A to node B in TC messages. (This is the opposite of HELLO messages, which advertise the metric from node B to node A in node A's HELLO messages.) This can be seen by considering a route connecting single OLSRv2 interface nodes P to Q to R to S. Node P receives its only information about the link from R to S in the TC messages transmitted by node R, which is an MPR of node S (assuming that only MPR selectors are reported in TC messages). Node S may not even transmit TC messages (if no nodes have selected it as an MPR and it has no attached networks to report). So any information about the metric of the link from R to S must also be included in the TC messages sent by node R, hence node R is responsible for reporting the metric for the link from R to S.

In this example, node P also receives information about the link between Q and R in the HELLO messages sent by node Q. Without the presence of link metrics, this link may be used by OLSRv2 for two hop routing to node R using just HELLO messages sent by node Q. Assuming that this property (which accelerates local route formation) is to be retained, node P must receive the metric of the link from Q to R in HELLO messages sent by node Q. This indicates that node Q must be responsible for reporting the metric for the outgoing link from Q to R. This is an addition to the incoming link metric information that a HELLO message must report.

This leaves two possible design choices:

- o HELLO messages can report only incoming link metrics. This is required only for links on this OLSRv2 interface, i.e. with a LINK_STATUS TLV, and only when indicating HEARD or SYMMETRIC. This prevents the use of two-hop routes informed only by HELLO messages, and would be a change to OLSRv2.
- o HELLO messages report both incoming and outgoing link metrics. The former is required only for links on this OLSRv2 interface, i.e. with a LINK_STATUS TLV, and only when indicating HEARD or SYMMETRIC. The latter is required for all OLSRv2 interfaces, i.e.

Internet-Draft

OLSRv2 Link Metrics

September 2008

including those with an OTHER_NEIGHB TLV as well as a LINK_STATUS TLV, but only when each is SYMMETRIC. If a node has more than one possible outgoing link metric value (on different OLSRV2 interfaces), it must use the smallest (even if using a LINK_STATUS TLV, and this smallest metric is for another OLSRV2 interface). In some cases the incoming and outgoing metrics will be equal, and signaling that allows just one metric to be sent in such cases will improve efficiency.

Accelerated two-hop route formation is a feature of OLSRV2 it would be unfortunate to lose, and hence the latter approach is adopted. In addition, Section 5.1 offers an additional reason for reporting outgoing link metrics, without which metrics can affect only routing, not flooding.

4.4. Defining Link Metrics

When a node reports a neighbor node in a HELLO message it may do so for the first time with LINK_STATUS == HEARD. The receiving node may immediately consider the link to be symmetric and use it.

As the node is responsible for defining and reporting incoming link metrics it must evaluate that metric, and attach that link metric to the appropriate address (which will have LINK_STATUS == HEARD) in the next HELLO message reporting that address on that OLSRV2 interface. There will be no outgoing link metric available to report.

This procedure requires a node to immediately decide on a link metric once it has heard a neighbor on an OLSRV2 interface for the first time. This is because, on receiving a HELLO message from this node, that neighbor will (unless link quality indicates otherwise) immediately consider the link to be symmetric and use it. This may, depending on the physical nature of the link metric, be too early for an ideal decision as to that metric, however a choice must be made (even if only that a default value is used). The metric may later be refined based on further observation of HELLO messages, other message transmissions between the nodes (it may be appropriate to use unicast packets to test the link) or other observations of the environment. It will probably be best to over-estimate the metric if initially uncertain as to its value, to discourage, rather than over-encourage, its use.

4.5. Link Metric TLVs

Metric values will naturally be reported using a new address block TLV, here named LINK_METRIC. The different types of metric, both physical and directional, will require the use of a TLV extended type to represent the type of the metric. To allow for efficient

Internet-Draft

OLSRv2 Link Metrics

September 2008

reporting of link metrics which are the same in both directions (which may be the case either by chance, or consistently if using, for example, a metric based on a limited number of bandwidth values) two bits, the least significant, will be allocated for direction (least significant bit 7 for incoming, bit 6 for outgoing) with two zeros used to indicate bidirectional, and two ones reserved. (Using two zeros rather than two ones allows a possible efficiency gain as is described below.) The remaining most significant six bits will be the link metric type, defined by the node parameter LINK_METRIC_TYPE, which must be in the range 0 to 63, inclusive.

Link metric types, and their physical meaning and mapping, will be allocated by IANA. Link metric types 56 to 63 will be for private/experimental use, and 1 to 55 will be allocated by expert review. Link metric type 0 will be defined by OLSRV2 as described below (thus also allowing an interoperable implementation of OLSRV2 with no further link metric type definitions). This defines a process for the allocation of all of the type extensions of the LINK_METRIC TLV, except those with two least significant bits both one, which will be reserved (even in the private/experimental range).

The value field of the LINK_METRIC TLV, which may be multivalued, will be as described in the following section. There will also be a default metric value, and a LINK_METRIC TLV with that value may be omitted, and if a link metric is required, but no LINK_METRIC TLV of the appropriate type is present, then that default value will be assumed.

The need for an extension of the TLV type costs an extra octet each time it is used. The following approach allows the extra octet to sometimes be omitted (always in TC messages). It also allows there to be only a single reference to the link metric type used in each message.

A message TLV, of type LINK_METRIC_EXTENSION is also defined. A message may not include more than one such TLV. It takes a single octet value, which represents the default LINK_METRIC type extension. Its most significant six bits must be the node parameter LINK_METRIC_TYPE. It may have its two least significant bits equal to 0, 1 or 2 (the latter will typically be used in a TC message, any may be used in a HELLO message but usually one of the first two will be preferred). If there is no LINK_METRIC_EXTENSION TLV then one with value zero is assumed.

Any LINK_METRIC TLV with no type extension is treated as having a type extension equal to the value of the LINK_METRIC_EXTENSION TLV in that message. Any LINK_METRIC TLV with a type extension whose most significant six bits are all zero replaces them with the most

Internet-Draft

OLSRv2 Link Metrics

September 2008

significant six bits of the value of the LINK_METRIC_EXTENSION TLV.

The use of the LINK_METRIC_EXTENSION TLV may be illustrated by assuming that that physical link type N is to be used. Then in a HELLO message, where all LINK_METRIC TLVs should have type extensions 4N, 4N+1 or 4N+2, these TLVs can instead have type extensions 0, 1 and 2, and the first of these (i.e. for all links with the same metric in each direction) can be omitted. In a TC message, where all LINK_METRIC TLVs should have type 4N+2, a single LINK_METRIC_EXTENSION TLV can have value 4N+2, and all LINK_METRIC TLV type extensions can be omitted.

For a network which does not use link metrics, simply omitting a LINK_METRIC_EXTENSION TLV and all LINK_METRIC TLVs uses only default values of a dimensionless metric, i.e. is equivalent to using hop count, with no additional overhead. However a node in such a network MUST still recognize and use link metrics in the event that other nodes use values other than the default values.

4.6. Link Metric Values

In keeping with the requirement that OLSRv2 can be unaware of the details of metric values (which may be defined in the future) a single link metric value definition is required. (It would be possible to have two options by taking a bit from the type extension, but this would cut down the number of available types to 32, and this is not recommended.)

As previously noted, a reasonably wide dynamic range of link metrics is desirable. On the other hand, link metrics that occupy no more than one octet are also desirable for message size reasons. One approach that includes both requirements is already in use in OLSRv2, for time values, as described in [timetlv]. This specifies a value using a mantissa and exponent, together occupying 8 bits. For link metric purposes a 4 bit mantissa and a 4 bit exponent is suggested here. ([timetlv] uses a 3 bit mantissa and a 5 bit exponent, offering increased range but reduced precision.) This would be used so that the transmitted octet $16*b + a$ represents the value $(1 + a/16) * 2^b$. This would then represent values from a minimum of 1 to a maximum of 63488. However this also allows fractional metrics, so for convenience it is suggested that the metric value range used is considered to be from a minimum of 16 to a maximum of $16 * 63488 (= 1015808)$, i.e. that $16*b + a$ represents the value $(16 + a) * 2^b$. Note that this rescaling has no effect on message contents or performance. The limiting values of the metric will be defined as the constants MINIMUM_METRIC (16) and MAXIMUM_METRIC (1015808) to allow their more convenient use. (It is recommended that all mappings from real parameters to link metric values are specified

Internet-Draft

OLSRv2 Link Metrics

September 2008

using these constants by name.)

As noted above, in order that metric use can be most efficient, a default value is needed. This also should be type-independent. It is suggested that this is in the centre of the above range logarithmically; the closest representable value is 4096 (a == 0, b == 8). This will be defined as the constant `DEFAULT_METRIC`. It is also suggested that route metric summation should be exact. Since a route cannot have more than 255 links, 28 bit (or more, in practice probably 32 bit) arithmetic can be used.

Internet-Draft

OLSRv2 Link Metrics

September 2008

5. MPRs with Link Metrics

MPRs are used for two purposes in OLSRV2. In both cases it is MPR selectors that are actually used, MPR selectors being determined from MPRs advertised in HELLO messages.

- o Optimized flooding. Note that this is currently managed by this node's Relay Set, whose minimum contents are the set of the OLSRV2 interface addresses of this node's MPR selectors which are connected to the relevant OLSRV2 interface of this node. A change to this approach is included in this document.
- o Routing. Note that routing is based on the Topology Set, which is based on received TC messages, whose contents are set from the Advertised Neighbor Set, whose minimum contents are the OLSRV2 interface addresses of the MPR selectors of this node.

Metrics interact with these two uses of MPRs differently, which are considered separately in the following two sections. The relationship between the two sets of MPRs is considered in Section 5.3.

5.1. Flooding MPRs

MPR selection for flooding can ignore metrics. Selection using any algorithm that ignores metrics, including any allowed by [OLSRv2], will produce a flooding solution that works.

However, that does not mean that metrics cannot be usefully considered in selecting MPRs for flooding. Consider the network in Figure 2, where numbers are metrics of links away from node A, by shortest routes. (Simple metric values are used for clarity, rather than using the range MINIMUM_METRIC to MAXIMUM_METRIC; the values could be replaced by scaled values in that range.)

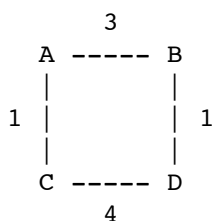


Figure 2

Which is the better MPR selection by node A: B or C? If the metric represents probability of message loss, then clearly choosing B maximizes the probability of a message sent by A reaching D. This is

Internet-Draft

OLSRv2 Link Metrics

September 2008

despite that C has a lower metric in its connection to A than B does. (Similar arguments about a preference for B can be made if, for example, the metric represents bandwidth or delay rather than probability of loss.)

However, this does not automatically mean that, as in the example above, only the second hop should be considered. If this example is modified to that in Figure 3:

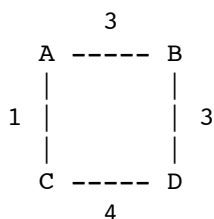


Figure 3

then it is possible that, for A, selecting C as a flooding MPR is preferable to selecting B. If the metrics represent scaled values of delay, or the probability of loss, then selecting C is clearly better. This indicates that the sum of metrics is an appropriate measure to use to choose between B and C.

However, this is a particularly simple example. Usually it is not a simple choice between two nodes as an MPR, each only adding one node coverage. A more general process, when considering which node to next add as an MPR, should incorporate the metric to that node, and the metric from that node to each symmetric strict 2-hop neighbor, as well as the number of newly covered symmetric strict 2-hop neighbors as well as the other factors used in the example algorithm in [OLSRv2].

Note that, as in [OLSRv2], each node can make its own independent choice of MPRs, and MPR selection algorithms, and still interoperate. A possible algorithm, representing a modification of the current algorithm in [OLSRv2] (and reducing to it when all metrics are equal, i.e. using minimum hop routing) is suggested in Section 6.11.

Note that the references above to the direction of the metrics is correct: for flooding, directional metrics outward from a node are appropriate, i.e. metrics in the direction of the flooding. This is an additional reason for including outward metrics in HELLO messages, as otherwise a metric-aware flooding MPR selection is not possible.

Internet-Draft

OLSRv2 Link Metrics

September 2008

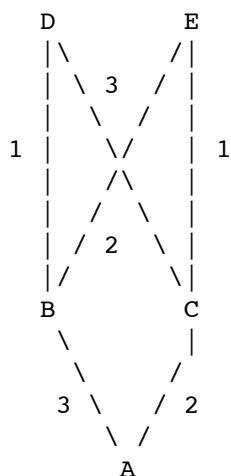


Figure 6

It is shown in Appendix A that selecting MPRs according to this definition, and advertising only such links (plus knowledge of local links from HELLO messages), will result in selection of shortest routes, even if all links are considered in the definition of shortest route.

However the definition noted above as sufficient for MPR selection is not necessary. For example, consider the network in Figure 7. (The metrics from B to C and C to B are both assumed to be 2.)

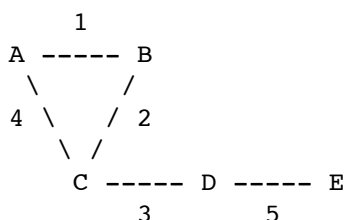


Figure 7

Using the above definition, A must pick both B and C as MPRs, in order to cover the symmetric strict 2-hop neighbors C and D, respectively. (C is a symmetric strict 2-hop neighbor because the route length via B is shorter than the 1-hop link.)

However, A only needs to pick B as an MPR, because the only reason to pick C as an MPR would be so that C can advertise the link to A for routing - to be used by, for example, E. But A knows that no other node should use the link C to A in a shortest route, because routing via B is shorter. So if there is no need to advertise the link from

Internet-Draft

OLSRv2 Link Metrics

September 2008

C to A, then there is no reason for A to select C as an MPR.

This process of "thinning out" the MPR selection uses just local information from HELLO messages. Using any minimum distance algorithm, the node identifies shortest routes, whether one, two or more hops, from all nodes in its symmetric strict 2-hop neighborhood. It then selects as MPRs all symmetric strict 1-hop neighbors that are the last node (before the selecting node itself) on any such route. Where there is more than one shortest distance route from a node, only one such route is required. Alternative routes may be selected so as to minimize the number of last nodes - this is the equivalent to the selection of a minimal set of MPRs in the non-metric case. (An example of how to perform this in practice is given in Appendix B.)

Note that, compared to the first proposed approach, this only removes MPRs whose selection can be directly seen to be unnecessary. Consequently if (as is shown in Appendix A) the first approach creates minimum distance routes, then so does this revised process.

Note that the examples in Figure 5 and Figure 6 show that use of link metrics may require a node to select more MPRs, and even select MPRs when otherwise it would not when not using metrics. This may result in more, and larger, messages being generated, and forwarded more often. Thus the use of link metrics is not without cost, even excluding the cost of link metric signaling. There is however no cost (in message size or number of messages) if all link metrics are default valued and no link metric TLV is used.

These examples consider only single OLSRv2 interface nodes. However if nodes have more than one OLSRv2 interface, then the process is unchanged, other than that if there is more than one known metric between two nodes (on different OLSRv2 interfaces), then the smallest must be used. There is no need to calculate MPRs per OLSRv2 interface for routing. That requirement results from the consideration of flooding and the need to avoid certain "deadlock" conditions, which are not relevant to routing.

5.3. Relationship Between MPR Sets

It would be convenient if the two sets of MPRs were the same. This can be the case if all metrics are equal (whether to the default value or any other value), but in general, for "good" sets of MPRs they are not. (A reasonable definition of this is that there is no common minimal set of MPRs.) If metrics are asymmetrically valued (the two sets of MPRs use opposite direction metrics), or nodes have multiple interfaces (where routing MPRs can ignore this, but flooding MPRs cannot) this is particularly unlikely. However even using a

Internet-Draft

OLSRv2 Link Metrics

September 2008

symmetrically valued metric with a single OLSRv2 interface on each node, the sets are not equal, nor is one always a subset of the other. To show this, consider these examples, where all lettered nodes are assumed equally willing to be MPRs, and numbers are bidirectional metrics for links.

In Figure 8, for flooding, A does not require any MPRs. For routing, A must select B as an MPR.

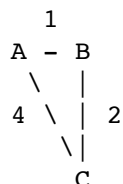


Figure 8

In Figure 9, for routing, A must select C and D as MPRs. For flooding a minimal set of MPRs for A is to just select B. In this example the set of routing MPRs will serve as a set of flooding MPRs, but a non-minimal one (although one that might be better, depending on the relative importance of number of MPRs and flooding link metrics).

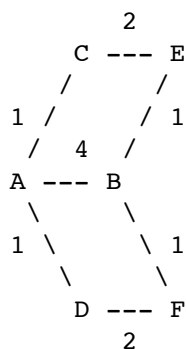


Figure 9

However, this is not always the case. In Figure 10, A's set of routing MPRs must contain B, it need not contain C. For MPR flooding, A need not select B, but it must select C. (In this case, flooding with A selecting B rather than C as an MPR will reach D, but in three hops rather than the minimum two that MPR flooding guarantees.)

Internet-Draft

OLSRv2 Link Metrics

September 2008

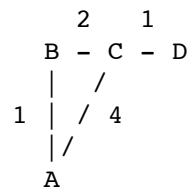


Figure 10

Internet-Draft

OLSRv2 Link Metrics

September 2008

6. Implementation

Implementation of metrics in OLSRV2 requires the following additions to [OLSRv2]:

- o Definition of the constant minimum, maximum and default metric values `MINIMUM_METRIC`, `MAXIMUM_METRIC` and `DEFAULT_METRIC`, and the mapping between metric values and single octet representation.
- o Definition of the node parameter `LINK_METRIC_TYPE`.
- o Addition of link metric information to the Local Information Base, the Interface Information Base and the Topology Information Base.
- o Modifications to the Interface Information Base, Node Information Base and Processing and Forwarding Information Base to reflect the two types of MPRs to be used.
- o A `LINK_METRIC` address block TLV to represent metrics, to handle incoming and outgoing/agreed cases and alternative link metric types.
- o A `LINK_METRIC_EXTENSION` message TLV to allow a single representation of the link metric type in a message.
- o A modification of the TLV to represent MPRs, to handle routing and flooding cases.
- o HELLO message generation to add metrics and both MPR types.
- o HELLO message processing to use metrics and both MPR types.
- o Separate routing and flooding MPR calculations and update of the Neighbor Set.
- o TC message generation to add metrics.
- o TC message processing to use metrics.
- o Routing Set updates to use metrics.

These changes are summarized in the following sections. Updates to the constraints that apply to the Information Bases are summarized in Appendix D.

Internet-Draft

OLSRv2 Link Metrics

September 2008

6.1. Parameters and Constants

The constant minimum, maximum and default metric values are defined by:

- o MINIMUM_METRIC = 16
- o MAXIMUM_METRIC = 1015808
- o DEFAULT_METRIC = 4096

The node parameter LINK_METRIC_TYPE may take any value from 0 to 63 inclusive. If this node parameter is changed, then all protocol sets which contain link metric information (i.e. all those updated in the following sections) must have all of their contents immediately removed, except that Link Tuples which are not pending should instead be updated by:

- o L_HEARD_time = EXPIRED
- o L_SYM_time = EXPIRED

The usual consequences of a Link Tuple no longer being symmetric, if it was, and of timeout of being heard, must be applied. The former of these will include, in all cases, not just this one:

- o L_mpr_selector = false

and the latter will include, in all cases, not just this one:

- o L_in_metric = unspecified
- o L_out_metric = unspecified

6.2. Local Information Base

Each Local Attached Network Tuple, defined in [OLSRv2] will need one additional element:

AL_metric is the metric of the link to the attached network with address AL_net_addr from this node;

This could replace the existing AL_dist element, however in order that the R_dist elements in a Routing Set can be set correctly (as there may be an external use for these) the AL_dist element has been retained, and hence also the hop count value in the GATEWAY TLV. Attached networks have not been discussed in this document up to this point, but they will behave very similarly to as currently defined in

Internet-Draft

OLSRv2 Link Metrics

September 2008

[OLSRv2], with appropriate use of this metric.

6.3. Interface Information Base

Each Link Tuple, defined in [OLSRv2] by reference to [NHDP], will need three additional elements:

`L_in_metric` is the metric of the link from the OLSRv2 interface with addresses `L_neighbor_iface_addr_list` to this OLSRv2 interface;

`L_out_metric` is the metric of the link to the OLSRv2 interface with addresses `L_neighbor_iface_addr_list` from this OLSRv2 interface;

`L_mpr_selector` is a boolean flag, describing if this neighbor has selected this node as a flooding MPR, i.e. is a flooding MPR selector of this node.

`L_in_metric` will be specified by a process outside the OLSRv2 specification, similarly to `L_quality`. When a Link Tuple is created, the default value of `L_in_metric` (used if not over-ridden) is `DEFAULT_METRIC`. When `L_HEARD_time` expires then `L_in_metric` should be set as unspecified.

`L_out_metric` will be defined by this protocol. When a Link Tuple is created, the default value of `L_out_metric` will be set as unspecified. Setting `L_out_metric` will require the corresponding `N_metric` to be updated by:

- o If there is no old `N_metric`, or if the new `L_out_metric` is less than the old `N_metric`, then set the new `N_metric` to the new `L_out_metric`.
- o Otherwise, if the old `L_out_metric` is equal to the old `N_metric`, and the new `L_out_metric` is greater than the old `N_metric`, then set the new `N_metric` to the minimum of all corresponding `L_out_metric` values, including the new `L_out_metric`.

The recording of flooding MPR selectors in the Link Set is part of a process that includes deleting the Relay Set from the Processing and Forwarding Information Base, and making relaying decisions specified only by the flooding MPR selector.

Each 2-Hop Tuple, defined in [OLSRv2] by reference to [NHDP], will need two additional elements:

Internet-Draft

OLSRv2 Link Metrics

September 2008

N2_in_metric is the metric of the link from the node with address N2_2hop_iface_addr to the node with OLSRV2 interface addresses N2_neighbor_iface_addr_list, being the lowest known such metric received on that OLSRV2 interface;

N2_out_metric is the metric of the link to the node with address N2_2hop_iface_addr from the node with OLSRV2 interface addresses N2_neighbor_iface_addr_list, being the lowest known such metric received on that OLSRV2 interface;

6.4. Node Information Base

Each Neighbor Tuple, defined in [OLSRv2] by reference to [NHDP], will need four additional or modified elements:

N_metric is the minimum metric of any link from this node to this neighbor, the minimum of any corresponding L_out_metric;

N_routing_mpr is a boolean flag, describing if this neighbor is selected as a routing MPR by this node;

N_flooding_mpr is a boolean flag, describing if this neighbor is selected as a flooding MPR by this node;

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

Note that flooding MPR selector status is recorded in the Link Sets, not in the Neighbor Set. N_routing_mpr and N_flooding_mpr replace N_mpr.

6.5. Topology Information Base

The Advertised Neighbor Set will consist of Advertised Neighbor Tuples. In addition to A_neighbor_iface_addr these will contain one additional element:

A_metric is the metric from this node to the node with OLSRV2 interface address A_neighbor_iface_addr.

Modifying any A_metric will update the ANSN. A_metric values are set from the corresponding N_metric values, and must be changed whenever those values are changed (as well as this, the Advertised Neighbor Set being changed when changes to N_mpr_selector values occur).

Each Topology Tuple, defined in [OLSRv2], will need one additional element:

Internet-Draft

OLSRv2 Link Metrics

September 2008

T_metric is the metric of the link from the node with originator address T_orig_addr to the OLSRV2 interface with address T_dest_iface_addr.

Each Attached Network Tuple, defined in [OLSRv2], will need one additional element:

AN_metric is the metric of the link from the node with originator address AN_orig_addr to the attached network with address AN_net_addr.

The existing AN_dist element is retained, as for AL_dist in the Local Attached Network Tuple.

Each Routing Tuple, defined in [OLSRv2], will need one additional element:

R_metric is the metric of the route to the destination with address R_dest_addr.

The R_dist element has been retained as well as adding R_metric. It is outside the scope of OLSRV2 to specify how R_dist and/or R_metric may be used when the Routing Set is used to update the IP routing table or for any other purpose.

6.6. Processing and Information Base

The Relay Sets are removed, as noted in Section 6.3.

6.7. Metric Representation

Both HELLO messages and TC messages will need to associate metric values with neighbor addresses that they report. These metric values will have a type defined by node parameter LINK_METRIC_TYPE. This in turn will define the most significant six bits of a TLV type extension, where the least significant two bits are 1 for an incoming metric, or 2 for an outgoing metric. If metrics are to be allocated in both directions, and they are equal, then a single representation of that metric, with least significant two bits 0 may be used.

The metric, and the type extension, may be represented in any one of the following ways:

- o If the metric value equals the constant DEFAULT_METRIC, then no representation is needed as long as there is a message LINK_METRIC_EXTENSION TLV with its value equal to the type extension defined above, unless this is zero in which case the LINK_METRIC_EXTENSION TLV may be omitted.

Internet-Draft

OLSRv2 Link Metrics

September 2008

- o The metric value may be represented by associating the address with a LINK_METRIC TLV with type extension as described above.
- o The metric value may be represented by associating the address with a LINK_METRIC TLV with type extension with least significant two bits as described above, and most significant six bits all zero, provided that there is a message LINK_METRIC_EXTENSION TLV with most significant six bits of its value equal to node parameter LINK_METRIC_TYPE, unless this is zero when the LINK_METRIC_EXTENSION TLV may be omitted.
- o The metric value may be represented by associating the address with a LINK_METRIC TLV with no type extension, provided that there is a message LINK_METRIC_EXTENSION TLV with its value equal to the type extension defined above, unless this is zero when the LINK_METRIC_EXTENSION TLV may be omitted.

However the intended use is more simply one of either:

- o When a metric is required, use a LINK_METRIC TLV with the appropriate type extension. A LINK_METRIC_EXTENSION TLV is not required.
- o Include a LINK_METRIC_EXTENSION TLV with value equal to the smallest type extension that would be required by a LINK_METRIC TLV using the first approach. When a metric is required, use a LINK_METRIC TLV with only the least significant two bits of the type extension used, except if these are equal to those in the value of the LINK_METRIC_EXTENSION TLV then omit the LINK_METRIC TLV type extension.

In both cases the metric may be omitted if equal in value to DEFAULT_METRIC. A LINK_METRIC_EXTENSION TLV whose value is zero may be omitted.

In all cases, association with a LINK_METRIC TLV may be with a TLV covering a single or multiple addresses, and in the latter case with a single or multiple values.

6.8. MPR Representation

The current single TLV which reports MPR status will need to report both routing and flooding MPR status. As the current TLV, it will report this for all relevant addresses of the node; however for flooding MPRs it does so only for addresses which have a symmetric link on the reporting OLSRV2 interface.

Rather than using separate TLVs, it is suggested that two extended

Internet-Draft

OLSRv2 Link Metrics

September 2008

types are used to represent these two types, and a third extended type is used to indicate both. The most efficient type extension, zero, could be used to represent both when a LINK_STATUS TLV with Type == SYMMETRIC is present, but to represent only routing MPR status when only an OTHER_NEIGHB TLV with Type == SYMMETRIC is present.

6.9. HELLO Message Generation

The following additional reporting by a HELLO message is required. Link metric association is as previously described.

- o Each included address from an L_neighbor_iface_addr_list with an associated LINK_STATUS TLV with Value == HEARD or Value == SYMMETRIC must have an associated incoming link metric of the appropriate type with value L_in_metric.
- o Each included address from an N_neighbor_iface_addr_list with an associated LINK_STATUS or OTHER_NEIGHB TLV with Value == SYMMETRIC must have an associated outgoing link metric of the appropriate type with value N_metric.
- o Each included address from an L_neighbor_iface_addr_list with an associated LINK_STATUS TLV with Value == SYMMETRIC must have an associated MPR TLV indicating flooding MPR status if and only if the corresponding N_flooding_mpr == true.
- o Each included address from an N_neighbor_iface_addr_list with an associated LINK_STATUS or OTHER_NEIGHB TLV with Value == SYMMETRIC must have an associated MPR TLV indicating routing MPR status if and only if the corresponding N_routing_mpr == true.

Routing and flooding MPR indications can be combined when appropriate.

6.10. HELLO Message Processing

Processing a HELLO message has the following extra steps:

- o When adding or updating a Link Tuple when the HELLO message includes an address of the receiving OLSRV2 interface with a LINK_STATUS TLV:
 - * If the reported status is HEARD or SYMMETRIC, then the appropriate L_out_metric must be set to the value of any incoming (to the sending node) link metric of the appropriate type associated with this address using a LINK_METRIC TLV. If there is no such TLV then L_out_metric is set to

Internet-Draft

OLSRv2 Link Metrics

September 2008

DEFAULT_METRIC.

- * If the reported status is LOST then L_out_metric is set as unspecified.

The corresponding N_metric must also be updated if necessary.

- o All 2-Hop Tuples that are added or updated by the HELLO message also have their N2_in_metric updated to the value of any associated incoming (to the sending node) link metric value of the appropriate type associated with this address using a LINK_METRIC TLV. If there is no such TLV then N2_in_metric is set to DEFAULT_METRIC.
- o All 2-Hop Tuples that are added or updated by the HELLO message also have their N2_out_metric updated to the value of any associated outgoing (to the sending node) link metric value of the appropriate type associated with this address using a LINK_STATUS TLV. If there is no such TLV then N2_out_metric is set to DEFAULT_METRIC.
- o When adding or updating a Link Tuple, if the HELLO message includes an address of the receiving OLSRv2 interface with a LINK_STATUS TLV with value SYMMETRIC, then the presence or absence of an associated MPR TLV indicating flooding TLV status will set or clear the appropriate L_mpr_selector.
- o When adding or updating a Neighbor Tuple, if the HELLO message includes an address of the receiving OLSRv2 interface with a LINK_STATUS or OTHER_NEIGHB TLV with value SYMMETRIC, then the presence or absence of an associated MPR TLV indicating routing TLV status will set or clear the appropriate N_mpr_selector.

6.11. MPR Calculation and Neighbor Set Update

For routing MPRs, a possible algorithm is given in Appendix B. This sets or clears N_routing_mpr in all Neighbor Tuples with N_symmetric == true.

For flooding MPRs, the existing per OLSRv2 interface algorithm can be used unchanged. In particular its first stage (adding necessary MPRs) and third stage (removing unnecessary MPRs) are appropriate unchanged. Its second stage, which prioritizes possible added MPRs, can have link metrics (L_out_metric + N2_out_metric) added as a consideration in that prioritization. One suggestion is that after picking candidate new MPRs which maximize the new coverage of two hop neighbors, ties can be broken (before tie breaking based on maximizing the total coverage of two hop neighbors, new and old) by

Internet-Draft

OLSRv2 Link Metrics

September 2008

minimizing the sum of `L_out_metric` + `N2_out_metric` for each candidate MPR, across all newly covered two hop neighbors. Whatever algorithm is used, it sets or clears `N_flooding_mpr` instead of the current `N_mpr`.

In addition to the modified algorithms, a modification of the circumstances in which they are needed (i.e. when the neighborhood has changed sufficiently) is also required, and is different in each case. That for flooding MPRs adds changes to `L_out_metric` and/or `N2_out_metric` values. As use of these is optional, so is the recalculation, furthermore cases may be restricted to when the metrics increase for MPRs or decrease for non-MPRs. That for routing MPRs adds changes to `L_in_metric` and/or `N2_in_metric` values, and is compulsory to maintain shortest routes.

6.12. TC Message Generation

The following additional contents of a TC message are required. Link metric association is as previously described.

- o Each included `A_neighbor_iface_addr` must have an associated outgoing link metric of the appropriate type with value `A_metric`.
- o Each included `AL_net_addr` must have an associated outgoing link metric of the appropriate type with value `AL_metric`.

6.13. TC Message Processing

Processing a TC message has the following extra steps:

- o When adding or updating a Topology Tuple, set `T_metric` to the value of any associated `LINK_METRIC` TLV, or to `DEFAULT_METRIC` if none.
- o When adding or updating an Attached Network Tuple, set `AN_metric` to the value of any associated `LINK_METRIC` TLV, or to `DEFAULT_METRIC` if none.

6.14. Routing Set Calculation

Routing Set calculation using the Network Topology Graph is unchanged, except that the arcs in the latter have metrics rather than hop counts:

- o For a local symmetric link use `L_out_metric`.
- o For a 2-hop symmetric link use `N2_out_metric`.

Internet-Draft

OLSRv2 Link Metrics

September 2008

- o For an advertised symmetric link use T_metric.
- o For a symmetric 1-hop neighbor address use N_metric.
- o For an attached network address use AN_metric.

An example algorithm, modified from that in [OLSRv2], is given in Appendix C.

Internet-Draft

OLSRv2 Link Metrics

September 2008

7. IANA Considerations

This document presents no IANA considerations. Addition of metrics to [OLSRv2] will add to that document's IANA considerations.

Internet-Draft

OLSRv2 Link Metrics

September 2008

8. Security Considerations

This document does not specify any security considerations.

Internet-Draft

OLSRv2 Link Metrics

September 2008

9. References

9.1. Normative References

- [OLSRv2] Clausen, T., Dearlove, C., and P. Jacquet, "The Optimized Link State Routing Protocol version 2", draft-ietf-manet-olsrv2-07.txt (work in progress), July 2008.
- [packetbb] Clausen, T., Dearlove, C., Dean, J., and C. Adjih, "Generalized MANET Packet/Message Format", draft-ietf-manet-packetbb-15.txt (work in progress), September 2008.

9.2. Informative References

- [RFC2501] Macker, J. and S. Corson, "Mobile Ad hoc Networking (MANET): Routing Protocol Performance Issues and Evaluation Considerations", RFC 2501, January 1999.
- [NHDP] Clausen, T., Dearlove, C., and J. Dean, "MANET Neighborhood Discovery Protocol (NHDP)", draft-ietf-manet-nhdp-07.txt (work in progress), July 2008.
- [timetlv] Clausen, T. and C. Dearlove, "Representing multi-value time in MANETs", draft-ietf-manet-timetlv-07.txt (work in progress), September 2008.

Internet-Draft

OLSRv2 Link Metrics

September 2008

Appendix A. MPR Routing Property

In order that nodes can find and use shortest routes in a network while using the minimum reduced topology supported by OLSRV2 (that a node only advertises its MPR selectors in TC messages), MPR selection must result in the property that there are shortest routes with all MPR intermediate nodes.

More formally, the required property is that for any distinct nodes X and Z there is a shortest path from X to Z , $X - Y_1 - Y_2 - \dots - Y_m - Z$ such that Y_1 is an MPR of Y_2 , \dots Y_m is an MPR of Z . Call such a path a routable path, and call this property the routable path property.

This appendix shows that the simplest previously described redefinition of MPR selection with link metrics has the consequence that the routable path property is satisfied. It assumes that nodes with willingness `WILL_NEVER` have been removed. It assumes a network of directed links each with a positive metric. It considers links to be between nodes, independent of interfaces.

Although this appendix is concerned with routes with minimum total metric, not hop count, it proceeds by induction on the number of hops. Although it considers minimum metric routes with a bounded number of hops, it then allows that number of hops to increase so that overall minimum metric paths, regardless of the number of hops, are considered.

The required definition for a node X selecting MPRs is that for each distinct node Z from which there is a two hop path, there is a shorter, or equally short, path which is either $Z - Y - X$ where Y is an MPR of X , or is the direct link $Z - X$. Note that the existence of locally known, shorter, but more than two hop paths, which can be used to reduce the numbers of MPRs, is not considered here.

The routable path property is a corollary of the property that for all positive integers n , and all distinct nodes X and Z , if there is any path from X to Z of n hops or fewer, then there is a shortest path, from among those of n hops or fewer, that is a routable path. This may be called the n -hop routable path property.

The n -hop routable path property is trivial for $n = 1$, and is the definition of the MPRs of Z for $n = 2$.

Proceeding by induction, assuming the n -hop routable path property is true for $n = k$, consider the case that $n = k+1$.

Suppose that $X - V_1 - V_2 - \dots - V_k - Z$ is a shortest $k+1$ hop path

Internet-Draft

OLSRv2 Link Metrics

September 2008

from X to Z. We construct a path which has no more than $k+1$ hops, has the same or shorter length (hence has the same, shortest, length considering only paths of up to $k+1$ hops, by assumption) and is a routable path.

First consider whether V_k is an MPR of Z. If it is not then consider the two hop path $V_{k-1} - V_k - Z$. This can be replaced either by a direct link $V_{k-1} - Z$ or by a two hop path $V_{k-1} - W_k - Z$ where W_k is an MPR of Z, such that the metric from V_{k-1} to Z by the replacement path is no longer. In the former case (replacement one hop link) this now produces a path of length k , and the previous inductive step may be applied. In the latter case we have replaced V_k by W_k , where W_k is an MPR of Z. Thus we need only consider the case that V_k is an MPR of Z.

We now apply the previous inductive step to the path $X - V_1 - \dots - V_{k-1} - V_k$, replacing it by an equal length path $X - W_1 - \dots - W_{m-1} - V_k$, where $m \leq k$, where this path is a routable path. Then because V_k is an MPR of Z, the path $X - W_1 - \dots - W_{m-1} - V_k - Z$ is a routable path, and demonstrates the n -hop routable path property for $n = k+1$.

This thus shows that, with the revised MPR definition, for any distinct nodes X and Z, there is a routable path (using OLSRV2's MPR-based reduced topology) from X to Z, i.e. that this modification of OLSRV2 still finds minimum length paths.

Internet-Draft

OLSRv2 Link Metrics

September 2008

Appendix B. Routing MPR Calculation

This is a possible algorithm for calculating routing MPRs. At the start of the calculation set `N_routing_mpr = false` in all Neighbor Tuples.

This calculation is not per OLSRV2 interface, but is for all OLSRV2 interfaces together. Thus the union of all of the node's Link Sets and the union of all of the node's 2-Hop Sets are considered in what follows.

For convenience assign each Link Tuple a unique identity `L_id`, and each Neighbor Tuple a unique identity `N_id`. These are transient, used during this calculation only.

Note that each 2-Hop Tuple has a unique corresponding Link Tuple and each Link Tuple has a unique corresponding Neighbor Tuple. Thus for each 2-Hop Tuple we can determine corresponding values of `L_id`, `N_id`, `L_in_metric` and `N_willingness`.

Define a Local Topology Tuple (used only during MPR calculation), which represents a route from a final node to this node, to include:

`LT_next_id` is the identity of the Neighbor Tuple corresponding to the nearest node to this node on the route;

`LT_last_id` is the identity of a Link Tuple corresponding to the furthest node on the route from this node, other than the final node, for an OLSRV2 interface on which that furthest node has a symmetric link to this node;

`LT_final_iface_addr` is an address of the final node;

`LT_last_metric` is the metric of the part of the route from the last node to this node;

`LT_final_metric` is the metric of the link from the final node to the last node;

`LT_number_hops` is the number of hops on the route from the final node to this node.

All such final nodes can reach this node in two hops, the first hop being to the last node other than the final node, but the preferred route may use more hops with a lower metric. When the route uses two hops, the last and next nodes are the same (the node between this node and the final node). As for 2-Hop Tuples, a separate Local Topology Tuple is used for each address of each final node.

Internet-Draft

OLSRv2 Link Metrics

September 2008

It is assumed here that between routes with equal metric, a route with fewest hops is preferred.

Then, for each 2-Hop Tuple whose corresponding `N_willingness` is not equal to `WILL_NEVER`, create a Local Topology Tuple with:

- o `LT_next_id` = corresponding `N_id`
- o `LT_last_id` = corresponding `L_id`
- o `LT_final_iface_addr` = `N2_2hop_iface_addr`
- o `LT_last_metric` = corresponding `L_in_metric`
- o `LT_final_metric` = `N2_in_metric`
- o `LT_number_hops` = 2

Now, while there are any two Local Topology Tuples (Tuple A and Tuple B) such that:

- o A's `LT_final_iface_addr` is in the `L_neighbor_iface_addr_list` corresponding to B's `LT_last_id`, and
- o A's `LT_last_metric` + `LT_final_metric` < B's `LT_last_metric`

update Tuple B by:

- o `LT_next_id` = A's `LT_next_id`
- o `LT_last_metric` = A's `LT_last_metric` + `LT_final_metric`
- o `LT_number_hops` = A's `LT_number_hops` + 1

This replaces Tuple B's route from this node to its last node by Tuple A's route from this node to its final node.

Once that process is finished, remove all Local Topology Tuples such that either:

- o there is a Link Tuple with `LT_final_iface_addr` in `L_neighbor_iface_addr_list`; AND
- o `L_in_metric` <= `LT_final_metric`

or:

Internet-Draft

OLSRv2 Link Metrics

September 2008

- o there is another Local Topology Tuple with the same LT_final_iface_addr; AND
 - * a smaller value of LT_last_metric + LT_final_metric; OR
 - * an equal value of LT_last_metric + LT_final_metric and a smaller value of LT_number_hops.

This removes Local Topology Tuples where either there is a Link Tuple offering a better one hop route, or another Local Topology Tuple offering a better route, from the final node.

For each remaining Local Topology Tuple define that the Neighbor Tuple with identity LT_next_id covers the 2-hop neighbor address LT_final_iface_addr.

A valid set of routing MPRs is any subset of these Neighbor Tuples which collectively cover all of these LT_final_iface_addr. Set the corresponding N_routing_mpr = true.

While any subset with this property is valid, a heuristic for a "good" subset is required. The current heuristic has three main steps: add necessary neighbors, add additional neighbors in a prioritized order until coverage is complete, remove unneeded neighbors (possibly in order of ascending willingness). There is no reason to modify this. The middle step currently uses the following priority order: greatest willingness, maximum new coverage, maximum coverage, if an MPR selector, any. This will still work (the MPR selector step should be routing MPR selector). It may be considered that metrics could be used. However in principle this is not necessary, as metrics have already been taken into account in this construction. (This differs from flooding MPRs, where considering metrics in this step is appropriate as they are not used up to this point.)

Internet-Draft

OLSRv2 Link Metrics

September 2008

Appendix C. Example Algorithm for Calculating the Routing Set

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

The following terminology is used:

- o A Neighbor Tuple corresponds to a Link Tuple if `N_neighbor_iface_addr_list` contains `L_neighbor_iface_addr_list`.
- o A Neighbor Tuple corresponds to a 2-Hop Tuple if `N_neighbor_iface_addr_list` contains `N2_neighbor_iface_addr_list`.
- o A Neighbor Tuple corresponds to a Routing Tuple if `N_neighbor_iface_addr_list` contains `R_next_iface_addr`.
- o A Neighbor Tuple is preferred to another Neighbor Tuple if either the former has greater `N_willingness` than the latter, or if they have equal `N_willingness`, the former has `N_mpr_selector == true`, and the latter has `N_mpr_selector == false`.

C.1. Add Local Symmetric Links

1. For each Local Interface Tuple:
 1. Select an address (the "local address") in `I_local_iface_addr_list`.
 2. For each Link Tuple for this local interface with `L_status == SYMMETRIC` and `L_out_metric == N_metric` of the corresponding Neighbor Tuple:
 1. For each address (the "current address") in `L_neighbor_iface_addr_list`, if there is no Routing Tuple with `R_dest_addr == current address`, then add a Routing Tuple with:
 - `R_dest_addr = current address;`
 - `R_next_iface_addr = current address;`
 - `R_dist = 1;`
 - `R_metric = N_metric;`

Internet-Draft

OLSRv2 Link Metrics

September 2008

- R_local_iface_addr = local address.

2. For each Neighbor Tuple which corresponds to a Routing Tuple (the "previous Tuple"):
 1. For each address (the "current address") in N_neighbor_iface_addr_list, if there is no Routing Tuple with R_dest_addr == current address, then add a Routing Tuple with:
 - + R_dest_addr = current address;
 - + R_next_iface_addr = R_dest_addr of the previous Tuple;
 - + R_dist = 1;
 - + R_metric = N_metric.
 - + R_local_iface_addr = R_local_iface_addr of the previous Tuple.

C.2. Add Remote Symmetric Links

The following procedure, which adds Routing Tuples for destination nodes h+1 hops away, MUST be executed for each value of h, starting with h = 1 and incrementing by 1 for each iteration. The execution MUST stop when no new Routing Tuples are added in an iteration.

1. For each Topology Tuple, if:
 - * For the Advertising Remote Node Tuple with AR_orig_addr == T_orig_addr, there is an address in the AR_iface_addr_list which is equal to the R_dest_addr of a Routing Tuple (the "previous Tuple") whose R_dist == h; AND
 - * One of:
 - + T_dest_iface_addr is not equal to R_dest_addr of any Routing Tuple; OR
 - + The Routing Tuple with R_dest_addr == T_dest_iface_addr has R_metric > (R_metric of the previous Tuple) + T_metric; OR
 - + The Routing Tuple with R_dest_addr == T_dest_iface_addr has R_metric == (R_metric of the previous Tuple) + T_metric, R_dist == h+1, and the Neighbor Tuple corresponding to the previous Tuple is preferred to the Neighbor Tuple corresponding to this Routing Tuple.

Internet-Draft

OLSRv2 Link Metrics

September 2008

then add a Routing Tuple (replacing any existing Routing Tuple with $R_dest_addr == T_dest_iface_addr$) with:

- * $R_dest_addr = T_dest_iface_addr$;
- * $R_next_iface_addr = R_next_iface_addr$ of the previous Tuple;
- * $R_dist = h+1$;
- * $R_metric = (R_metric \text{ of the previous Tuple}) + T_metric$;
- * $R_local_iface_addr = R_local_iface_addr$ of the previous Tuple.

2. After the above iteration has completed, if $h == 1$, then for each 2-Hop Tuple and corresponding Neighbor Tuple, if:

- * $N_willingness$ is not equal to $WILL_NEVER$; AND
- * There is a Routing Tuple (the "previous Tuple") with $R_dist == 1$ to which the Neighbor Tuple corresponds,; AND
- * One of:
 - + $N2_2hop_iface_addr$ is not equal to R_dest_addr of any Routing Tuple; OR
 - + The Routing Tuple with $R_dest_addr == N2_2hop_iface_addr$ has $R_metric > N_metric + N2_out_metric$; OR
 - + The Routing Tuple with $R_dest_addr == N2_2hop_iface_addr$ has $R_metric == N_metric + N2_out_metric$, $R_dist == 2$ and the Neighbor Tuple corresponding to the 2-Hop Tuple is preferred to the Neighbor Tuple corresponding to the Routing Tuple,

then add a Routing Tuple (replacing any existing Routing Tuple with $R_dest_addr == N2_2hop_iface_addr$) with:

- * $R_dest_addr = N2_2hop_iface_addr$;
- * $R_next_iface_addr = R_next_iface_addr$ of the previous Tuple;
- * $R_dist = 2$;
- * $R_metric = N_metric + N2_out_metric$;
- * $R_local_iface_addr = R_local_iface_addr$ of the previous Tuple.

Internet-Draft

OLSRv2 Link Metrics

September 2008

C.3. Add Attached Networks

1. For each Attached Network Tuple, if:

- * For the Advertising Remote Node Tuple with AR_orig_addr == AN_orig_addr there is an address in the AR_iface_addr_list which is equal to the R_dest_addr of a Routing Tuple (the "previous Tuple"); AND
- * One of:
 - + There is no Routing Tuple with R_dest_addr == AN_net_addr;
OR
 - + The Routing Tuple with R_dest_addr == AN_net_addr has R_metric > (R_metric of the previous Tuple) + AN_metric; OR
 - + The Routing Tuple with R_dest_addr == AN_net_addr has R_metric == (R_metric of the previous Tuple) + AN_metric, and R_dist > (R_dist of the previous Tuple) + AN_metric,

then add a new Routing Tuple (replacing any existing Routing Tuple with R_dest_addr == AN_net_addr) with:

- * R_dest_addr = AN_net_addr;
- * R_next_iface_addr = R_next_iface_addr of the previous Tuple;
- * R_dist = (R_dist of the previous Tuple) + AN_dist;
- * R_metric = (R_metric of the previous Tuple) + AN_metric;
- * R_local_iface_addr = R_local_iface_addr of the previous Tuple.

Internet-Draft

OLSRv2 Link Metrics

September 2008

Appendix D. Constraints

The constraints specified in [OLSRv2] must be updated to match modifications to the Information Bases. These constraint modifications are as described in this appendix.

In each Local Attached Network Tuple:

- o AL_metric MUST be representable as the value of a LINK_METRIC TLV (hence MUST NOT be less than MINIMUM_METRIC and MUST NOT be greater than MAXIMUM_METRIC).

In each Link Tuple:

- o If L_status is not LOST then L_in_metric MUST be representable as the value of a LINK_METRIC TLV (hence MUST NOT be less than MINIMUM_METRIC and MUST NOT be greater than MAXIMUM_METRIC).
- o If L_status is LOST then L_in_metric MUST be considered to be unspecified.
- o If L_status is SYMMETRIC then L_out_metric MUST be representable as the value of a LINK_METRIC TLV (hence MUST NOT be less than MINIMUM_METRIC and MUST NOT be greater than MAXIMUM_METRIC).
- o If L_status is not SYMMETRIC then L_out_metric MUST be considered to be unspecified.
- o If L_mpr_selector == true then N_symmetric MUST be true.

In each Neighbor Tuple constraints involving N_mpr apply to both N_flooding_mpr and N_routing_mpr, and:

- o If N_symmetric == true then N_metric MUST equal the minimum value of all L_out_metric values that are not unspecified and whose corresponding L_neighbor_iface_addr_list is contained in this N_neighbor_iface_addr_list.
- o If N_symmetric == false then N_metric MUST be considered to be unspecified.

In each 2-Hop Tuple:

- o N2_in_metric MUST be representable as the value of a LINK_METRIC TLV (hence MUST NOT be less than MINIMUM_METRIC and MUST NOT be greater than MAXIMUM_METRIC).

Internet-Draft

OLSRv2 Link Metrics

September 2008

- o N2_out_metric MUST be representable as the value of a LINK_METRIC TLV (hence MUST NOT be less than MINIMUM_METRIC and MUST NOT be greater than MAXIMUM_METRIC).

In each Advertised Neighbor Tuple:

- o A_metric MUST equal N_metric for the Neighbor Tuple whose N_neighbor_iface_addr_list contains this A_neighbor_iface_addr.

In each Topology Tuple:

- o T_metric MUST be representable as the value of a LINK_METRIC TLV (hence MUST NOT be less than MINIMUM_METRIC and MUST NOT be greater than MAXIMUM_METRIC).

In each Attached Network Tuple:

- o AN_metric MUST be representable as the value of a LINK_METRIC TLV (hence MUST NOT be less than MINIMUM_METRIC and MUST NOT be greater than MAXIMUM_METRIC).

Internet-Draft

OLSRv2 Link Metrics

September 2008

Appendix E. Acknowledgements

The authors would like to thank Alan Cullen (BAE Systems) for review and comments, and Brian Adamson (NRL), Justin Dean (NRL), Charles Perkins (WiChorus) and Stan Ratliff (Cisco) for discussions.

Internet-Draft

OLSRv2 Link Metrics

September 2008

Authors' Addresses

Christopher Dearlove
BAE Systems Advanced Technology Centre

Phone: +44 1245 242194
EMail: chris.dearlove@baesystems.com
URI: <http://www.baesystems.com/>

Thomas Heide Clausen
LIX, Ecole Polytechnique, France

Phone: +33 6 6058 9349
EMail: T.Clausen@computer.org
URI: <http://www.ThomasClausen.org/>

Philippe Jacquet
INRIA, France

Phone: +33 1 3963 5263
EMail: Philippe.Jacquet@inria.fr
URI: <http://hipercom.inria.fr/>

Internet-Draft

OLSRv2 Link Metrics

September 2008

Full Copyright Statement

Copyright (C) The IETF Trust (2008).

This document is subject to the rights, licenses and restrictions contained in BCP 78, and except as set forth therein, the authors retain all their rights.

This document and the information contained herein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY, THE IETF TRUST AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Intellectual Property

The IETF takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights. Information on the procedures with respect to rights in RFC documents can be found in BCP 78 and BCP 79.

Copies of IPR disclosures made to the IETF Secretariat and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF on-line IPR repository at <http://www.ietf.org/ipr>.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this standard. Please address the information to the IETF at ietf-ipr@ietf.org.

