Networking Working Group
Internet-Draft
Intended status: Standards Track
Expires: September 10, 2012

O. Gnawali
P. Levis
Stanford University

March 9, 2012

The Minimum Rank with Hysteresis Objective Function
draft-ietf-roll-minrank-hysteresis-of-07

Abstract

The Routing Protocol for Low Power and Lossy Networks (RPL) uses objective functions to construct routes that optimize or constrain the routes it selects and uses. This specification describes the Minimum Rank Objective Function with Hysteresis (MRHOF), an objective function that selects routes that minimize a metric, while using hysteresis to reduce churn in response to small metric changes. MRHOF works with metrics that are additive along a route, and the metric it uses is determined by the metrics RPL Destination Information Object (DIO) messages advertise.

Table of Contents

1. Introduction

An objective function specifies how RPL [I-D.ietf-roll-rpl] selects paths. For example, if an RPL instance uses an objective function that minimizes hop-count, RPL will select paths with minimum hop count. RPL requires that all nodes in a network use a common OF; relaxing this requirement may be a subject of future study.

The nodes running RPL might use a number of metrics to describe a link or a node [I-D.ietf-roll-routing-metrics] and make these metrics available for route selection. RPL advertises metrics in RPL Destination Information Object (DIO) messages with a Metric Container suboption. An objective function can use these metrics to choose routes.

To decouple the details of an individual metric or objective function from forwarding and routing, RPL describes routes through a value called Rank. Rank, roughly speaking, corresponds to the distance associated with a route. RPL defines how nodes decide on paths based on Rank and advertise their Rank. An objective function defines how nodes calculate Rank, based on the Rank of its potential parents, metrics, and other network properties.

This specification describes MRHOF, an objective function for RPL. MRHOF uses hysteresis while selecting the path with the smallest metric value. The metric that MRHOF uses is determined by the metrics in the DIO Metric Container. For example, the use of MRHOF with the latency metric allows RPL to find stable minimum-latency paths from the nodes to a root in the DAG instance. The use of MRHOF with the ETX metric allows RPL to find the stable minimum-ETX paths from the nodes to a root in the DAG instance. In the absence of a metric in the DIO Metric Container or the lack of a DIO Metric Container, MRHOF defaults to using ETX to compute Rank, as described in Section 3.5.

Because MRHOF seeks to minimize path costs as described by metrics, it can only be used with additive metrics. MRHOF ignores metrics t

2. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

This terminology used in this document is consistent with the terminologies described in [I-D.ietf-roll-terminology],

[I-D.ietf-roll-rpl], and [I-D.ietf-roll-routing-metrics].

This document introduces three terms:

Selected metric: The metric chosen by the network operator to use for path selection. This metric can be any additive metric listed in [I-D.ietf-roll-routing-metrics].

Path cost: Path cost quantifies a property of an end-to-end path. Path cost is obtained by summing up the selected metric of the links or nodes along the path. Path cost can be used by RPL to compare different paths.

Worst parent: The node in the parent set with the largest path cost.

3. The Minimum Rank Objective Function with Hysteresis

The Minimum Rank with Hysteresis Objective Function, MRHOF, is designed to find the paths with the smallest path cost while preventing excessive churn in the network. It does so by finding the minimum cost path and switching to that path only if it is shorter (in terms of path cost) than the current path by at least a given threshold.

MRHOF may be used with any additive metric listed in [I-D.ietf-roll-routing-metrics] as long the routing objective is to minimize the given routing metric. Nodes MUST support at least one of these metrics: node energy, hop count, latency, link quality level, and ETX. Nodes SHOULD support the ETX metric. MRHOF does not support non-additive metrics.

3.1. Computing the Path cost

Root nodes (Grounded or Floating) set the variable cur_min_path_cost to MIN_PATH_COST.

If a non-root node does not have metrics to compute the path cost through any of the candidate neighbors, it MUST join one of the candidate neighbors as a RPL Leaf.

Otherwise, nodes compute the path cost for each candidate neighbor reachable on an interface. The path cost of a neighbor represents the cost of the path, in terms of the selected metric, from a node to the root of the DODAG through that neighbor. A non-root node computes a neighbor's path cost by adding two components:

1. If the selected metric is a link metric, the selected metric for the link to a candidate neighbor. If the selected metric is a node metric, the selected metric for the node.

2. The value of the selected metric in the metric container in the DIO sent by that neighbor.

A node SHOULD compute the path cost for the path through each candidate neighbor reachable through an interface. If a node cannot compute the path cost for the path through a candidate neighbor, the node MUST NOT select the candidate neighbor as its preferred parent, except if it cannot compute the path cost through any neighbor, in which case it may join as a leaf as described above.

If the selected metric is a link metric and the metric of the link to a neighbor is not available, the path cost for the path through that neighbor SHOULD be set to MAX_PATH_COST. This cost value will prevent this path from being considered for path selection.

If the selected metric is a node metric, and the metric is not available, the path cost through all the neighbors SHOULD be set to MAX_PATH_COST.

The path cost corresponding to a neighbor SHOULD be re-computed each time:

1. The selected metric of the link to the candidate neighbor is updated.

2. If the selected metric is a node metric and the metric is updated.

3. A node receives a new metric advertisement from the candidate neighbor.

This computation MAY also be performed periodically. Too much delay in updating the path cost after the metric is updated or a new metric advertisement is received can lead to stale information.

3.2. Parent Selection

After computing the path cost for all the candidate neighbors reachable through an interface for the current DODAG iteration, a node selects the preferred parent. This process is called parent selection. To allow hysteresis, parent selection maintains a variable, cur_min_path_cost, which is the path cost of the current preferred parent.

3.2.1. When Parent Selection Runs

A MRHOF implementation SHOULD perform Parent Selection each time:

1. The path cost for an existing candidate neighbor, including the preferred parent, changes. This condition can be checked immediately after the path cost is computed.

2. A new candidate neighbor is inserted into the neighbor table.

The parent selection MAY be deferred until a later time. Deferring the parent selection can delay the use of better paths available in the network.

3.2.2. Parent Selection Algorithm

If the selected metric for a link is greater than MAX_LINK_METRIC, the node SHOULD exclude that link from consideration during parent selection.

A node MUST select the candidate neighbor with the lowest path cost as its preferred parent, except as indicated below:

1. A node MAY declare itself as a Floating root, and hence have no preferred parent, depending on system configuration.

2. If cur_min_path_cost is greater than MAX_PATH_COST, the node MAY declare itself as a Floating root.

3. If the smallest path cost for paths through the candidate neighbors is smaller than cur_min_path_cost by less than PARENT_SWITCH_THRESHOLD, the node MAY continue to use the current preferred parent. This is the hysteresis component of MRHOF.

4. If ALLOW_FLOATING_ROOT is 0 and no neighbors are discovered, the node does not have a preferred parent, and MUST set cur_min_path_cost to MAX_PATH_COST.

If there are multiple neighbors which share the smallest path cost, a node MAY use a different objective function to select which of these neighbors should be considered to have the lowest cost.

A node MAY include up to PARENT_SET_SIZE-1 additional candidate neighbors in its parent set. The cost of path through the nodes in the parent set is smaller than or equal to the cost of the paths through any of the nodes that are not in the parent set. If the cost of the path through the preferred parent and the worst parent is too large, a node MAY keep a smaller parent set than PARENT_SET_SIZE.

Once the preferred parent is selected, the node sets its cur_min_path_cost variable to the path cost corresponding to the preferred parent. The value of the cur_min_path_cost is carried in the metric container corresponding to the selected metric when DIO messages are sent.

3.3. Computing Rank

DAG roots set their Rank to MinHopRankIncrease.

Once a non-root node selects its parent set, it can use the following table to covert the path cost of a parent (written as Cost in the table) to a Rank value:

Node/link Metric	Rank
Node Energy	255 - Cost
Hop-Count	Cost
Latency	Cost/65536
Link Quality Level	Cost
ETX	Cost

Table 1: Conversion of metric to rank.

Rank is undefined for these node/link metrics: node state and attributes, throughput, and link color. If the rank is undefined, the node must join one of the neighbors as a RPL Leaf node according to [I-D.ietf-roll-rpl].

MRHOF uses this Rank value to compute the Rank it associates with the path through each meqber of the parent set. The Rank associated with a path through a member of the parent set is the maximum of two values. The first is corresponding Rank value calculated with the table above, the second is the that node's advertised Rank plus MinHopRankIncrease.

A node sets its Rank to the maximum of three values:

1. The Rank calculated for the path through the preferred parent

2. The Rank of the member of the parent set with the highest advertised Rank plus one

3. The largest calculated Rank among paths through the parent set, minus MaxRankIncrease

The first case is the Rank associated with the path through the preferred parent. The second case covers requirement 4 of Rank advertisements in Section 8.2.1 of [I-D.ietf-roll-rpl]. The third case ensures that a node does not advertise a Rank which then precludes it from using members of its parent set.

Note that the third case means that a node advertises a conservative Rank value based on members of its parent set, which might be significantly higher than the Rank calculated for the path through the preferred parent. Accordingly, picking a parent set whose paths have a large range of Ranks will likely result in sub-optimal routing: nodes will not choose good paths because they are advertised as much worse than they actually are. The exact selection of a parent set is an implementation decision.

3.4. Advertising the Path Cost

Once the preferred parent is selected, the node sets its cur_min_path_cost variable to the path cost corresponding to its preferred parent. It then calculates the metric it will advertise in its metric container. This value is the path cost of the member of the parent set with the highest path cost. Thus, while cur_min_path_cost is the cost through the preferred parent, a node advertises the highest cost path from the node to the root through a member of the parent set. The value of the highest cost path is carried in the metric container corresponding to the selected metric when DIO messages are sent.

If ETX is the selected metric, a node SHOULD NOT advertise it in a metric container. Instead, a node MUST advertise an approximation of its ETX in its advertised Rank value, following the rules described in Section 3.3. If a node receives a DIO with a metric container holding an ETX metric, MRHOF MUST ignore the ETX metric value in its Rank calculations.

DODAG Roots advertise a metric value which computes to a cost of MIN_PATH_COST.

3.5. Working Without Metric Containers

In the absence of metric container, MRHOF uses ETX as its metric. It locally computes the ETX of links to its neighbors and adds this value to their advertised Rank to compute the associated Rank of routes. Once parent selection and rank computation is performed using the ETX metric, the node advertises a Rank equal to the ETX cost and MUST NOT include a metric container in its DIO messages.

4. Using MRHOF for Metric Maximization

MRHOF cannot be directly used for parent selection using metrics which require finding paths with maximum value of the selected metric, such as path reliability. It is possible to convert such a metric maximization problem to a metric minimization problem for some metrics and use MRHOF provided:

There is a fixed and well-known maximum metric value corresponding to the best path. This is the path cost for the DAG root. For example, the logarithm of the best link reliability has a value of 0.

The metrics in the maximization problem are all negative. The logarithm of the link reliability is always negative.

For metrics meeting the above conditions, the problem of maximizing the metric value is equivalent to minimizing the modified metric value, e.g., logarithm of link reliability. MRHOF is not required to work with these metrics.

5. MRHOF Variables and Parameters

MRHOF uses the following variable:

cur_min_path_cost: The cost of the path from a node through its preferred parent to the root computed at the last parent selection.

MRHOF uses the following parameters:

MAX_LINK_METRIC: Maximum allowed value for the selected link metric for each link on the path.

MAX_PATH_COST: Maximum allowed value for the path metric of a selected path.

MIN_PATH_COST: The minimum allowed value for the path metric of the selected path.

PARENT_SWITCH_THRESHOLD: The difference between the cost of the path through the preferred parent and the minimum cost path in order to trigger the selection of a new preferred parent.

PARENT_SET_SIZE: The number of candidate parents, including the preferred parent, in the parent set.

ALLOW_FLOATING_ROOT: If set to 1, allows a node to become a floating root.

The parameter values are assigned depending on the selected metric. The best values for these parameters should be experimentally determined. The working group has long experience routing with the ETX metric. Based on those experiences, these values are RECOMMENDED:

MAX_LINK_METRIC: 10. Disallow links with greater than 10 expected transmission count on the selected path.

MAX_PATH_COST: 100. Disallow paths with greater than 100 expected transmission count.

MIN_PATH_COST: 0. At root, the expected transmission count is 0.

PARENT_SWITCH_THRESHOLD: 172. Switch to a new path only if it is expected to require at least 1.5 fewer transmissions than the current path. As ETX is encoded as number of transmissions times 128, this is a threshold of 172.

PARENT_SET_SIZE: 3. If the preferred parent is not available, two candidate parents are still available without triggering a new round of route discovery.

ALLOW_FLOATING_ROOT: 0. Do not allow a node to become a floating root.

6. Manageability

Section 18 of [I-D.ietf-roll-rpl] depicts the management of RPL. This specification inherits from that section and its subsections, with the exception that metrics as specified in [I-D.ietf-roll-routing-metrics] are not used and do not require management.

6.1. Device Configuration

An implementation SHOULD allow the following parameters to be configured at installation time: MAX_LINK_METRIC, MAX_PATH_COST, MIN_PATH_COST, PARENT_SWITCH_THRESHOLD, PARENT_SET_SIZE, and ALLOW_FLOATING_ROOT. An implementation MAY allow these parameters to be configured dynamically at run time once a network has been deployed.

A MRHOF implementation SHOULD support the DODAG Configuration option

as described in [I-D.ietf-roll-rpl] and apply the parameters it specifies. Care should be taken in the relationship between the MRHOF PARENT_SWITCH_THRESHOLD parameter and the RPL MaxRankIncrease parameter. For example, if MaxRankIncrease is smaller than PARENT_SWITCH_THRESHOLD, a RPL node using MRHOF could enter a situation where its current preferred parent causes the nodes Rank to increase more than MaxRankIncrease but MRHOF does not change preferred parents: this could cause the node to leave the routing topology even though there may be other members of the parent set which would allow the node's Rank to remain within MaxRankIncrease.

Unless configured otherwise, a MRHOF implementation SHOULD use the default parameters as specified in Section 5.

Because of the partially-coupled relationship between Rank and metric values, networks using MRHOF require care in setting MinHopRankIncrease. A large MinHopRankIncrease will cause MRHOF to be unable to select paths with different hop counts but similar metric values. If MinHopRankIncrease is large enough that its increment is greater than that caused by link cost, then metrics will be used to select a preferred parent but the advertised Rank will be a simple hopcount. This behavior might be desirable, but it also might be unintended: care is recommended.

6.2. Device Monitoring

A MRHOF implementation should provide an interface for monitoring its operation. At a minimum, the information provided should include:

DAG information as specified in Section 6.3.1 of [I-D.ietf-roll-rpl], and including the DODAGID, the RPLInstanceID, the Mode of Operation, the Rank of this node, the current Version Number, and the value of the Grounded flag.

A list of neighbors indicating the preferred parent. The list should indicate, for each neighbor, the Rank, the current Version Number, the value of the Grounded flag, and associated metrics.

7. Acknowledgements

Thanks to Antonio Grilo, Nicolas Tsiftes, Matteo Paris, JP Vasseur, and Phoebus Chen for their comments.

8. IANA Considerations

This specification requires a value allocated from the "Objective

Code Point (OCP)" sub-registry of the "Routing Protocol for Low Power and Lossy Networks (RPL)" registry. A value of 1 is suggested.

9. Security Considerations

This specification makes simple extensions to RPL and so is vulnerable to and benefits from the security issues and mechanisms described in [I-D.ietf-roll-rpl] and [I-D.ietf-roll-security-framework]. This document does not introduce new flows or new messages, thus requires no specific mitigation for new threats.

MRHOF depends on information exchanged in a number RPL protocol elements. If those elements were compromised, then an implementation of MRHOF might generate the wrong path for a packet, resulting in it being misrouted. Therefore, deployments are RECOMMENDED to use RPL security mechanisms if there is a risk that routing information might be modified or spoofed.

10. References

10.1. Normative References

[I-D.ietf-roll-routing-metrics]
Barthel, D., Vasseur, J., Pister, K., Kim, M., and N. Dejean, "Routing Metrics used for Path Calculation in Low Power and Lossy Networks", draft-ietf-roll-routing-metrics-19 (work in progress), March 2011.

[I-D.ietf-roll-rpl]
Brandt, A., Vasseur, J., Hui, J., Pister, K., Thubert, P., Levis, P., Struik, R., Kelsey, R., Clausen, T., and T. Winter, "RPL: IPv6 Routing Protocol for Low power and Lossy Networks", draft-ietf-roll-rpl-19 (work in progress), March 2011.

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

10.2. Informative References

[I-D.ietf-roll-security-framework]
Tsao, T., Alexander, R., Dohler, M., Daza, V., and A. Lozano, "A Security Framework for Routing over Low Power and Lossy Networks", draft-ietf-roll-security-framework-07

(work in progress), January 2012.

[I-D.ietf-roll-terminology]
Vasseur, J., "Terminology in Low power And Lossy Networks", draft-ietf-roll-terminology-05 (work in progress), March 2011.

Authors' Addresses

Omprakash Gnawali
Stanford University
S255 Clark Center, 318 Campus Drive
Stanford, CA 94305
USA

Phone: +1 650 725 6086
Email: gnawali@cs.stanford.edu

Philip Levis
Stanford University
412 Gates Hall, Stanford University
Stanford, CA 94305
USA

Email: pal@cs.stanford.edu