

ROLL
Internet-Draft
Intended status: Standards Track
Expires: December 17, 2011

P. Thubert, Ed.
Cisco Systems
June 15, 2011

RPL Objective Function 0
draft-ietf-roll-of0-13

Abstract

The Routing Protocol for Low Power and Lossy Networks (RPL) specification defines a generic Distance Vector protocol that is adapted to a variety of networks types by the application of specific Objective Functions. An Objective Function defines how a RPL node selects and optimizes routes within a RPL Instance based on the information objects available. This document specifies a basic Objective Function that relies only on the objects that are defined in RPL and does not use any extension.

Requirements Language

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

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of [BCP 78](#) and [BCP 79](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on December 17, 2011.

Copyright Notice

Copyright (c) 2011 IETF Trust and the persons identified as the document authors. All rights reserved.

Internet-Draft

[draft-ietf-roll-of0](#)

June 2011

This document is subject to [BCP 78](#) and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1.	Introduction	3
2.	Terminology	4
3.	Objective Function 0 Overview	4
4.	OF0 Operations	5
4.1.	Computing Rank	5
4.2.	Feasible Successors Selection	7
4.2.1.	Selection Of The Preferred Parent	7
4.2.2.	Selection Of The Backup Feasible Successor	8
5.	Abstract Interface to OF0	8
6.	OF0 Operands	9
6.1.	Variables	9
6.2.	Configurable Parameters	9
6.3.	Constants	10
7.	Manageability Considerations	10
7.1.	Device Configuration	10
7.2.	Device Monitoring	11
8.	IANA Considerations	11
9.	Security Considerations	12
10.	Acknowledgements	12
11.	References	12
11.1.	Normative References	12
11.2.	Informative References	12
	Author's Address	13

1. Introduction

The Routing Protocol for LLN (RPL) [[I-D.ietf-roll-rpl](#)] specification defines a generic Distance Vector protocol that is adapted to a variety of Low Power and Lossy Networks (LLN) types by the application of specific Objective Functions. An Objective Function defines how a RPL node selects and optimizes routes within a RPL Instance based on the information objects available. This separation of Objective Functions from the core protocol specification allows RPL to be adapted to meet the different optimization criteria required by the wide range of deployments, applications and network designs.

RPL forms Directed Acyclic Graphs (DAGs) as collections of Destination Oriented DAGs (DODAGs) within instances of the protocol. Each instance is associated with a specialized Objective Function. A DODAG is periodically reconstructed as a new DODAG Version to enable a global reoptimization of the graph.

An instance of RPL running on a device uses an Objective Function to help it determine which DODAG Version it should join. The OF is also used by the RPL instance to select a number of routers within the DODAG Version to serve as parents or as feasible successors.

The RPL instance uses the OF to compute a Rank for the device. This value represents an abstract distance to the root of the DODAG within the DODAG Version. The Rank is exchanged between nodes using RPL and allows other RPL nodes to avoid loops and verify forward progression toward the destination, as specified in [[I-D.ietf-roll-rpl](#)].

The Objective Function Zero (OF0) operates on parameters that are obtained from provisioning, the RPL DODAG Configuration option and the RPL DIO base container [[I-D.ietf-roll-rpl](#)].

The Rank of a node is obtained by adding a normalized scalar, `rank_increase` ([Section 6.1](#)), to the Rank of a selected preferred

parent. The rank_increase can vary with a ratio from 1 (excellent) to 9 (worst acceptable) to represent the link properties. By default, OF0 encodes the 2-octet Rank in units of 256, and the default settings allow to encode a minimum of 28 (worst acceptable) hops and a maximum of 255 (excellent) hops.

It is important that devices deployed in a particular network or environment use the same OF to build and operate DODAGs. If they do not, it is likely that sub-optimal paths will be selected). In practice, without a common definition of an OF, RPL implementations cannot guarantee to interoperate correctly. The RPL specification [[I-D.ietf-roll-rpl](#)] does not include any OF definitions. This is

left for other documents specific to different deployments and application environments. Since there is no default OF or metric container in the RPL main specification, it might happen that, unless two given implementations follow the same guidance for a specific problem or environment, those implementations will not support a common OF with which they could interoperate.

OF0 is designed as a default OF that will allow interoperation between implementations in a wide spectrum of use cases. This is why it is not specific as to how the link properties are transformed into a rank_increase and leaves that responsibility to the implementation; rather, OF0 enforces normalized values for the rank_increase of a normal link and its acceptable range, as opposed to formulating the details of its computation. This is also why OF0 ignores metric containers.

[2.](#) Terminology

The Terminology used in this document is consistent with and incorporates that described in 'Terminology in Low power And Lossy Networks' [[I-D.ietf-roll-terminology](#)] and [[I-D.ietf-roll-rpl](#)].

The term 'feasible successor' is used to refer to a neighbor that can possibly be used as a next-hop for upwards traffic following the loop avoidance and forwarding rules that the nodes implements and that are defined in the RPL specification [[I-D.ietf-roll-rpl](#)].

3. Objective Function 0 Overview

The RPL specification describes constraints on how nodes select potential parents, called a parent set, from their neighbors. All parents are feasible successors for upward traffic (towards the root). Additionally, RPL allows the use of parents in a subsequent Version of a same DODAG as feasible successors, in which case this node acts as a leaf in the subsequent DODAG Version.

The Goal of the OF0 is for a node to join a DODAG Version that offers good enough connectivity to a specific set of nodes or to a larger routing infrastructure though there is no guarantee that the path will be optimized according to a specific metric. Thus, for the purpose of OF0, the term Grounded [[I-D.ietf-roll-rpl](#)] means that the DODAG root provides such connectivity. How that connectivity is asserted and maintained is out of scope.

Objective Function 0 is designed to find the nearest Grounded root. This can be achieved if the Rank of a node is very close to an

abstract function of its distance to the root. This need is balanced with the other need of maintaining some path diversity, which may be achieved by increasing the Rank. In the absence of a Grounded root, inner connectivity within the Low Power and Lossy Network is still desirable and floating DAGs will form, rooted at the nodes with the highest administrative preference.

OF0 selects a preferred parent and a backup feasible successor if one is available. All the upward traffic is normally routed via the preferred parent with no attempt to perform any load balancing. When the link conditions do not let an upward packet through the preferred parent, the packet is passed to the backup feasible successor.

A RPL node monitors links to a number of neighbor nodes, and can use OF0 to assign a rank_increase to each link. Though the exact method for computing the rank_increase is implementation-dependent, the computation must follow the rules that are specified in [Section 4.1](#).

4. OF0 Operations

[4.1](#). Computing Rank

An OF0 implementation first computes a variable `step_of_rank` ([Section 6.1](#)) associated with a given parent from relevant link properties and metrics. The `step_of_rank` is used to compute the amount by which to increase the rank along a particular link, as explained later in this section.

Computing a `step_of_rank` based on a static metric such as an administrative cost implies that the OF0 implementation only considers parents with good enough connectivity, and results in a Rank that is analogous to hop-count. In most LLNs, this favors paths with fewer but longer hops of poorer connectivity; it is thus RECOMMENDED to base the computation of the `step_of_rank` on dynamic link properties such as the expected transmission count metric (ETX) [[I-D.ietf-roll-routing-metrics](#)],[[DeCouto03](#)]. The Minimum Rank Objective Function with Hysteresis [[I-D.ietf-roll-minrank-hysteresis-of](#)] provides guidance on how link cost can be computed and on how hysteresis can improve Rank stability.

OF0 allows an implementation to stretch the `step_of_rank` in order to enable the selection of at least one feasible successor and thus maintain path diversity. Stretching the `step_of_rank` is NOT RECOMMENDED, because it augments the apparent distance from the node to the root, distorts the DODAG from the optimal shape and may cause instabilities due to greedy behaviors whereby depending nodes augment

their Ranks to use each other as parents in a loop. Still, an implementation MAY stretch the `step_of_rank` with at most a configurable `stretch_of_rank` ([Section 6.2](#)) of any value between 0 (no stretch) and the fixed constant `MAXIMUM_RANK_STRETCH` ([Section 6.3](#)).

An implementation MUST maintain the stretched `step_of_rank` between the fixed constants `MINIMUM_STEP_OF_RANK` and `MAXIMUM_STEP_OF_RANK` ([Section 6.3](#)). This range allows to reflect a large variation of link quality.

The gap between `MINIMUM_STEP_OF_RANK` and `MAXIMUM_RANK_STRETCH` may not be sufficient in every case to strongly distinguish links of different types or categories in order to favor, say, powered over battery-operated or wired over wireless, within a same DAG. An implementation SHOULD allow to configure a factor called `rank_factor`

([Section 6.2](#)) and to apply the factor on all links and peers to multiply the effect of the stretched `step_of_rank` in the `rank_increase` computation as further detailed below.

Additionally, an implementation MAY recognize categories of peers and links, such as different MAC types, in which case it SHOULD be able to configure a more specific `rank_factor` to those categories. The `rank_factor` MUST be set between the fixed constants `MINIMUM_RANK_FACTOR` and `MAXIMUM_RANK_FACTOR` ([Section 6.3](#)) .

The variable `rank_increase` is represented in units expressed by the variable `MinHopRankIncrease` which defaults to the fixed constant `DEFAULT_MIN_HOP_RANK_INCREASE` ([\[I-D.ietf-roll-rpl\]](#)); with that setting, the least significant octet in the RPL Rank field in the DIO Base Object is not used.

The `step_of_rank` S_p that is computed for that link is multiplied by the `rank_factor` R_f and then possibly stretched by a `stretch_of_rank` S_r . The resulting `rank_increase` is added to the Rank of preferred parent $R(P)$ to obtain that of this node $R(N)$:

$R(N) = R(P) + \text{rank_increase}$ where:

$\text{rank_increase} = (R_f * S_p + S_r) * \text{MinHopRankIncrease}$

Optionally, the administrative preference of a root MAY be configured to supersede the goal to join a Grounded DODAG. In that case, nodes will associate to the root with the highest preference available, regardless of whether that root is Grounded or not. Compared to a deployment with a multitude of Grounded roots that would result in the same multitude of DODAGs, such a configuration may result in possibly less but larger DODAGs, as many as roots configured with the highest priority in the reachable vicinity.

[4.2.](#) Feasible Successors Selection

[4.2.1.](#) Selection Of The Preferred Parent

As it scans all the candidate neighbors, OF0 keeps the parent that is the best for the following criteria (in order):

1. [\[I-D.ietf-roll-rpl\]](#) [section 8](#) spells out the generic rules for a

node to re-parent and in particular the boundaries to augment its Rank within a DODAG Version. A candidate that would not satisfy those rules MUST NOT be considered.

2. An implementation SHOULD validate a router prior to selecting it as preferred. This validation process is implementation and link type dependent, and is out of scope. A router that succeeded that validation process is preferable.
3. When multiple interfaces are available, a policy might be locally configured to order them and that policy applies first; that is a router on a higher order interface in the policy is preferable.
4. If the administrative preference of the root is configured to supersede the goal to join a Grounded DODAG, a router that offers connectivity to a more preferable root SHOULD be preferred.
5. A router that offers connectivity to a grounded DODAG Version SHOULD be preferred over one that does not.
6. A router that offers connectivity to a more preferable root SHOULD be preferred.
7. When comparing 2 parents that belong to the same DODAG, a router that offers connectivity to the most recent DODAG Version SHOULD be preferred.
8. The parent that causes the lesser resulting Rank for this node, as specified in [Section 4.1](#), SHOULD be preferred.
9. A DODAG Version for which there is an alternate parent SHOULD be preferred. This check is OPTIONAL. It is performed by computing the backup feasible successor while assuming that the router that is currently examined is finally selected as preferred parent.
10. The preferred parent that was in use already SHOULD be preferred.

11. A router that has announced a DIO message more recently SHOULD

be preferred.

These rules and their order MAY be varied by an implementation according to configured policy.

[4.2.2.](#) Selection Of The Backup Feasible Successor

When selecting a backup feasible successor, the OF performs in order the following checks:

1. When multiple interfaces are available, a router on a higher order interface is preferable.
2. The backup feasible successor MUST NOT be the preferred parent.
3. The backup feasible successor MUST be either in the same DODAG Version as this node or in an subsequent DODAG Version.
4. Along with RPL rules, a Router in the same DODAG Version as this node and with a Rank that is higher than the Rank computed for this node MUST NOT be selected as a feasible successor.
5. A router with a lesser Rank SHOULD be preferred.
6. A router that has been validated as usable by an implementation-dependant validation process SHOULD be preferred.
7. The backup feasible successor that was in use already SHOULD be preferred.

These rules and their order MAY be varied by an implementation according to configured policy.

[5.](#) Abstract Interface to OF0

Objective Function 0 interacts for its management and operations in the following ways:

Processing DIO: When a new DIO is received, the OF that corresponds to the Objective Code Point (OCP) in the DIO is triggered with the content of the DIO. OF0 is identified by OCP 0 (to be validated by IANA [Section 8](#)).

Providing DAG information: The OF0 support provides an interface that returns information about a given instance. This includes material from the DIO base header, the role (router, leaf), and the Rank of this node.

Providing a Parent List: The OF0 support provide an interface that returns the ordered list of the parents and feasible successors for a given instance to the RPL core. This includes the material that is contained in the transit option for each entry.

Triggered Updates: the OF0 support provides events to inform it that a change in DAG information or Parent List as occurred. This can be caused by an interaction with another system component such as configuration, timers, and device drivers, and the change may cause the RPL core to fire a new DIO or reset trickle timers.

[6.](#) OF0 Operands

On top of variables and constants defined in [[I-D.ietf-roll-rpl](#)], this specification introduces the following variables and constants:

[6.1.](#) Variables

OF0 uses the following variables:

`step_of_rank` (unsigned integer): an intermediate computation based on the link properties with a certain neighbor.

`rank_increase` (unsigned integer): delta between the Rank of the preferred parent and self

[6.2.](#) Configurable Parameters

OF0 can use the following optional configurable values that are used as parameters to the `rank_increase` computation:

`stretch_of_rank` (unsigned integer): the maximum augmentation to the `step-of-rank` of a preferred parent to allow the selection of an additional feasible successor. If none is configured to the device, then the `step_of_rank` is not stretched.

`rank_factor` (unsigned integer): A configurable factor that is used to multiply the effect of the link properties in the `rank_increase` computation. If none is configured, then a `rank_factor` of 1 is used.

[6.3.](#) Constants

Section 17 of [[I-D.ietf-roll-rpl](#)] defines RPL constants. OF0 fixes the values of the following constants:

DEFAULT_STEP_OF_RANK: 3

MINIMUM_STEP_OF_RANK: 1

MAXIMUM_STEP_OF_RANK: 9

DEFAULT_RANK_STRETCH: 0

MAXIMUM_RANK_STRETCH: 5

DEFAULT_RANK_FACTOR: 1

MINIMUM_RANK_FACTOR: 1

MAXIMUM_RANK_FACTOR: 4

[7.](#) Manageability Considerations

Section 18 of [[I-D.ietf-roll-rpl](#)] depicts the management of the protocol. 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.

[7.1.](#) Device Configuration

An implementation SHOULD allow to configure at least a global `rank_factor` that applies to all links. Additionally, the implementation may allow to group interfaces, links and/or neighbors and configure a more specific `rank_factor` to such groups.

An implementation MAY allow to configure a maximum `stretch_of_rank` as discussed in [Section 4.1](#). If none is configured, a value of 0 is

assumed and the `step_of_rank` is not stretched.

An OF0 implementation SHOULD support the DODAG Configuration option as specified in section 6.7.6 of [[I-D.ietf-roll-rpl](#)] and apply the parameters contained therein. When the option is used, the parameters are configured to the nodes that may become DODAG roots, and the nodes are configured to redistribute the information using the DODAG Configuration option. In particular, the value of `MinHopRankIncrease` can be distributed with that option and override

Thubert

Expires December 17, 2011

[Page 10]

Internet-Draft

[draft-ietf-roll-of0](#)

June 2011

the fixed constant of `DEFAULT_MIN_HOP_RANK_INCREASE` that is defined section 17 of [[I-D.ietf-roll-rpl](#)] with a fixed value of 256.

At build-time, the default constant values should be used, that is:

the `rank_factor` is set to the fixed constant `DEFAULT_RANK_FACTOR` ([Section 6.3](#)).

the maximum `stretch_of_rank` is set to the fixed constant `DEFAULT_RANK_STRETCH` ([Section 6.3](#)).

the `MinHopRankIncrease` is set to the fixed constant `DEFAULT_MIN_HOP_RANK_INCREASE` ([\[I-D.ietf-roll-rpl\]](#)).

The values can be overridden at anytime and apply at the next Version of the DODAG.

[7.2](#). Device Monitoring

As discussed in [Section 5](#), the OF support must be able to provide information about its operations, and trigger events when that information changes. At a minimum, the information 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 and an alternate feasible if available. For each neighbor, the Rank, the current Version Number, and the value of the Grounded flag should be indicated

8. IANA Considerations

This specification requires the assignment of an Objective Code Point (OCP) for OF0 in the Objective Code Point Registry that is requested in section 20.5. of [[I-D.ietf-roll-rpl](#)].

OCP code: The value of 0 is suggested.

Description: A basic Objective Function that relies only on the objects that are defined in [[I-D.ietf-roll-rpl](#)].

Thubert

Expires December 17, 2011

[Page 11]

Internet-Draft

[draft-ietf-roll-of0](#)

June 2011

Defining RFC: This.

9. Security Considerations

This specification inherits from [[I-D.ietf-roll-rpl](#)] for its messages and security mechanisms. Section 19 of [[I-D.ietf-roll-rpl](#)] provides Security Considerations that are valid for both documents. This document does not introduce new flows or new messages, thus requires no specific mitigation for new threats.

For RPL operations in general but applicable to the operations described in this document, [[I-D.ietf-roll-security-framework](#)] evaluates the security threats and identifies applicable countermeasures.

10. Acknowledgements

Most specific thanks to Philip Levis and Phoebus Chen for their help in finalizing this document.

Many thanks also to Adrian Farrel, Tim Winter, JP Vasseur, Julien Abeille, Mathilde Durvy, Teco Boot, Navneet Agarwal, Meral Shirazipour and Henning Rogge for in-depth review and first hand

implementers' feedback.

11. References

11.1. Normative References

[I-D.ietf-roll-rpl]

Winter, T., Thubert, P., Brandt, A., Clausen, T., Hui, J., Kelsey, R., Levis, P., Pister, K., Struik, R., and J. Vasseur, "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.

11.2. Informative References

[DeCouto03]

De Couto, Aguayo, Bicket, and Morris, "A High-Throughput Path Metric for Multi-Hop Wireless Routing", MobiCom '03 The 9th ACM International Conference on Mobile

Thubert

Expires December 17, 2011

[Page 12]

Internet-Draft

[draft-ietf-roll-of0](#)

June 2011

Computing and Networking, San Diego, California,, 2003, <http://pdos.csail.mit.edu/papers/grid:mobicom03/paper.pdf>.

[I-D.ietf-roll-minrank-hysteresis-of]

Gnawali, O. and P. Levis, "The Minimum Rank Objective Function with Hysteresis", [draft-ietf-roll-minrank-hysteresis-of-04](#) (work in progress), May 2011.

[I-D.ietf-roll-routing-metrics]

Vasseur, J., Kim, M., Pister, K., Dejean, N., and D. Barthel, "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-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-05](#) (work in progress), April 2011.

[I-D.ietf-roll-terminology]

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

Author's Address

Pascal Thubert (editor)
Cisco Systems
Village d'Entreprises Green Side
400, Avenue de Roumanille
Batiment T3
Biot - Sophia Antipolis 06410
FRANCE

Phone: +33 497 23 26 34
Email: pthubert@cisco.com