ROLL	P. Thubert, Ed.
Internet-Draft	Cisco Systems
Intended status: Standards Track	May 17, 2011
Expires: November 18, 2011	

RPL Objective Function 0
draft-ietf-roll-of0-12

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 a specific Objective Function. 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

Copyright Notice

Table of Contents

1. Introduction

The Routing Protocol for Low Power and Lossy Networks *[I-D.ietf-roll-rpl]* was designed as a generic core that is agnostic to metrics and that is adapted to a given problem using Objective Functions (OF). This separation of Objective Functions from the core protocol specification allows RPL to adapt to meet the different optimization criteria required by the wide range of use cases.

RPL forms Destination Oriented Directed Acyclic Graphs (DODAGs) within instances of the protocol. Each instance is associated with a specialized Objective Function. A DODAG is periodically reconstructed in a new Version to enable a global reoptimization of the graph.
An Objective Function selects the DODAG Version that a device joins within an instance, and a number of neighbor routers within that DODAG Version as parents or feasible successors. The OF generates the Rank of the device, that represents an abstract distance to the root within the DODAG. In turn, the Rank is used by the generic RPL core to avoid loops and verify forward progression towards a destination, as specified in [I-D.ietf-roll-rpl].
The Objective Function 0 (OF0) operates on parameters that are obtained from provisionning, 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, 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. As a result, OF0 with default settings allows to encode a minimum of 28 (worst acceptable) hops and a maximum of 255 (excellent) hops.
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 upgoing 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. For the purpose of OF0, Grounded thus means that the 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 represents closely 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, LLN inner connectivity 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.

OF0 assigns a rank_increase to each link to another node that it monitors. 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 step_of_rank associated with a given parent from relevant link properties and metrics.

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 analoguous to hop-count. In most LLNs, this favors paths with less 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) [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.

An implementation MAY allow to stretch the step_of_rank with a stretch_of_rank up to no more than MAXIMUM_RANK_STRETCH in order to enable the selection of at least one feasible successor and thus maintain path diversity. The use of a stretch_of_rank is not recommended as 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 behaviours whereby depending nodes augment

their Ranks to use each other as parents in a loop. An implementation MUST maintain the stretched step_of_rank between MINIMUM_STEP_OF_RANK and MAXIMUM_STEP_OF_RANK, which 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 a configurable factor called rank_factor and to apply the factor on all links and peers.
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 SHOULD be set between MINIMUM_RANK_FACTOR and MAXIMUM_RANK_FACTOR.
The rank_increase is expressed in units of MinHopRankIncrease, which defaults to DEFAULT_MIN_HOP_RANK_INCREASE; with that setting, the least significant octet in the RPL Rank is not used.
The step_of_rank Sp that is computed for that link is multiplied by the rank_factor Rf and then possibly stretched by a stretch_of_rank Sr. 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) + rank_increase where:
rank_increase = (Rf*Sp + Sr) * 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 a 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 reparent 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.

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.

5. Abstract Interface with RPL core

Objective Function 0 interacts with the RPL core in the following ways:

Processing DIO: When a new DIO is received, the RPL core calls the OF that corresponds to the Objective Code Point OCP) in the DIO . OF0 corresponds to the OCP 0 (to be validated by IANA).

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.

Calling back: The RPL core provides a call back interface for the OF0 support 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

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 parameters:

stretch_of_rank (unsigned integer): an optional augmentation to the step-of-rank of the preferred parent to allow the selection of additional parents.

rank_factor (unsigned integer): A configurable factor that is used to multiply the effect of the link properties in the rank_increase computation.

6.3. Constants

OF0 fixes the following constants:

DEFAULT_MIN_HOP_RANK_INCREASE: 256

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. IANA Considerations

This specification requires the assignment of a RPL Objective Code Point (OCP) for OF0. The value of 0 is suggested.

8. Security Considerations

Security Considerations for OCP/OF are to be developed in accordance with recommendations laid out in, for example, [I-D.tsao-roll-security-framework].

9. Acknowledgements

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

Many thanks also to Tim Winter, JP Vasseur, Julien Abeille, Mathilde Durvy, Teco Boot, Navneet Agarwal and Henning Rogge for in-depth review and first hand implementer's feedback.

10. References

10.1. Normative References

[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-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", Internet-Draft draft-ietf-roll-rpl-19, March 2011.
[I-D.ietf-roll-terminology]	Vasseur, J, "Terminology in Low power And Lossy Networks", Internet-Draft draft-ietf-roll-terminology-06, September 2011.
[I-D.tsao-roll-security-framework]	Tsao, T, Alexander, R, Daza, V and A Lozano, "A Security Framework for Routing over Low Power and Lossy Networks", Internet-Draft draft-tsao-roll-security-framework-02, March 2010.
[I-D.ietf-roll-minrank-hysteresis-of]	Gnawali, O and P Levis, "The Minimum Rank Objective Function with Hysteresis", Internet-Draft draft-ietf-roll-minrank-hysteresis-of-04, May 2011.
[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 Computing and Networking, San Diego, California,, 2003.

Author's Address

Pascal Thubert editor Thubert 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