                Recommendations for Efficient Implementation of RPL
                    draft-gnawali-roll-rpl-recommendations-01

Abstract

   RPL is a flexible routing protocol applicable to a wide range of Low
   Power and Lossy Networks. To enable this wide applicability, RPL
   provides many configuration options and gives implementers choices on
   how to implement various components of RPL. Drawing on our
   experiences, we distill the design choices and configuration
   parameters that lead to efficient RPL implementations and operations.

Status of this Memo

   This Internet-Draft is submitted to IETF in full conformance with the
   provisions of BCP 78 and 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/1id-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 September 14, 2011.

Table of Contents

1. Introduction

   RPL [I-D.ietf-roll-rpl] is a routing protocol that is applicable in a
   wide range of settings in networks characterized by low power and
   lossy links (LLN). Because RPL is designed to work in a wide range
   of settings, it offers many configuration parameters and choices in
   how different mechanisms are implemented. This flexibility is
   essential to ensure the wide applicability of this protocol.

   One can take advantage of this flexibility to implement and configure
   RPL in the most efficient way for a given network. However, it is
   easy to inadvertently configure RPL to work inefficiently in the
   network. These design choices must be made carefully drawing on
   implementation and operational experiences.

   In this document, we describe aspects of configuration and mechanisms
   that impact the performance of RPL. We hope these descriptions serve
   as guidelines and best practices for RPL implementers and enables
   them to understand why certain design and configuration choices are
   favored over others.

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 does not introduce new terms.

3. Set the Minimum Trickle Interval with Care

   The minimum Trickle interval determines the fastest rate at which RPL
   will send DIOs. It is not useful to have multiple DIOs in the
   transmit queue at a given node. The information in the older DIOs is
   likely already stale when the new DIO is generated. In systems that
   cannot cancel the packets that are already in the queue, it is
   advisable to set the minimum interval to be much larger than the
   minimum link layer packet time.

4. Use Large Maximum Trickle Interval

   The maximum Trickle interval determines the slowest rate at which RPL
   will send DIOs. It is recommended that the maximum interval is set
   to several hours. A large interval does not necessarily make RPL
   less agile or the routing information stale. Trickle will operate at
   a rate between the minimum and maximum interval depending on the
   dynamics in the network.

5. Use Small Trickle Redundancy Constant

   If a node receives more DIOs than the redundancy constant, it does
   not transmit, i.e., suppresses, its DIO. The rationale for this
   suppression is that the additional DIOs do not help discover new or
   better paths if certain number of DIOs have already been transmitted
   in the neighborhood of a node. In general, the smaller this number
   the more efficient the route discovery. Setting this value too small
   can lead to network partitioning as many nodes will suppress their
   DIOs and will not be discovered. A constant of 3-5 has been found
   adequate in deployments.

6. Poison Route Sparingly

   It is often not necessary for a node to poison a route explicitly by
   advertising a rank of INFINITY. With datapath validation, it is easy
   to detect a loop and coupled with adaptive beaconing, the routes can
   be repaired quickly without additional explicit mechanism for route
   poisoning. Poisoning the route does not prevent loops because the
   control packet can get dropped on the lossy link.

7. Preserve Neighbor Information

   The neighborhood information is useful even when a node detects that
   it has lost a route. It is recommended that the nodes not flush the
   entire or subset of the neighbor table even when a node loses its
   route or detects a loop. It is sufficient to mark the nodes in the
   table with the updated information that resulted in route loss or
   loops, e.g., marking the particular parent with a rank of INFINITY.

8. Slow-Down Datapath Traffic During Path Inconsistency

   When a node detects that a path is inconsistent through datapath
   validation, it tasks the control plane to repair the topology and
   make it consistent. During this time, although the route is

available, it is advisable that the data packets are sent at lower
rates to reduce contention with the control packets. This slow-down
can increase data packet latency or lead to queue overflow.

9. Choose Better Path Cost Over Route Stability

With bursty links, a link metric designed to reflect link quality
accurately can change rapidly. Other link metrics may also change
rapidly. As a result, the path cost computed using these agile
metrics can change rapidly. Selecting the best path then implies
frequent parent changes. Route flapping is not detrimental to the
performance of many network protocols such as sensor data collection
over UDP. Hence, oftentimes, it is better to optimize for path cost
than for path stability.

10. Acknowledgements

Thanks to Ulrich Herberg and Mukul Goyal for valuable comments.

11. IANA Considerations

None.

12. Security Considerations

Security considerations to be developed in accordance to the output
of the WG.

13. References

13.1. Normative References

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

13.2. Informative References

[I-D.ietf-roll-routing-metrics]
           Vasseur, J. and D. Networks, "Routing Metrics used for
           Path Calculation in Low Power and Lossy Networks",
           draft-ietf-roll-routing-metrics-01 (work in progress),
           October 2009.

[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-18 (work in
          progress), February 2011.

[I-D.ietf-roll-terminology]
          Vasseur, J., "Terminology in Low power And Lossy
          Networks", draft-ietf-roll-terminology-01 (work in
          progress), May 2009.

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
   358 Gates Hall, Stanford University
   Stanford, CA 94305
   USA

   Email: pal@cs.stanford.edu