Reliability Enhanced RPL for Wireless Sensor Networks

Qiang Liu[a], Hongjiang Lv[b] and Ning Yang[c]

School of Communication & Information Engineering

University of Electronic Science and Technology of China, Chengdu, China

[a]liuqiang@uestc.edu.cn, [b]191153346@qq.com, [c]yn@uestc.edu.cn

Keywords: LLN, RPL, reliability, Contiki.

Abstract. In this paper, we investigate the problem of reliability of RPL (IPv6 Routing Protocol for Low power and Lossy Networks) when applied to route traffic in Wireless Sensor Networks (WSNs). RPL is a routing protocol adapted for information routing with low power, low storage and processing sensor devices. However, there are two problems when RPL runs with ND (Neighbor Discovery) and uses the link metric such as ETX. First, loops may occur in some situation. Second, lack a mechanism to refresh the life time of a neighbor. In this paper, two mechanisms, Loop Avoidance (LA) and Neighbor Life Time Refresh (NLTR) are proposed to resolve these problems. The simulation results show that Loop Avoidance mechanism and the Neighbor Life Time Refresh mechanism can enhance the performance of the entire network significantly. Especially using these two mechanisms together, the performance of entire network is the best. These two mechanisms can effectively improve the reliability of RPL.

Introduction

A typical Wireless Sensor Networks (WSN) belong Low power and Lossy Networks (LLN) which are defined as a class of computer networks characterized by their high loss, low capacity interconnection links and low power equipment. RPL is an IPv6 distance vector routing protocol designed for such networks. RPL operates on IPv6 routing protocol and IEEE 802.15.4, which is bridged by 6LoWPAN. RPL is a gradient routing technique [1] that organizes the WSN as a Direct Acyclic Graph (DAG) rooted at the sink. It tries to minimize the cost to reach the sink from any node in the WSN using an objective function, thus obtaining a very high flexibility with respect to the operating scenario.

Although RPL is an IPv6 distance vector routing protocol for LLNs [2][3], there are still some shortcomings on reliability of routing. The first problem is that the loops may occur in some instances which results in routing error. The second is, the life time of a neighbor is not updated in time. This will generate some extra neighbor detection messages which may cause network congestion. In this paper, we consider to perfect RPL protocol to settle these two problems. Two mechanisms, Loop Avoidance (LA) and Neighbor Life Time Refresh (NLTR) are proposed respectively for above problems. In particular, using the Contiki COOJA simulator [4] available at [5][6], we evaluated the protocol reliabilities of RPL via counting packets delivery rate.

The remainder of this paper is organized as follows. In Section II, we expose the motivation of our current work, an analysis of the issues of native RPL with respect to reliability, and some related work. In Section III, we present the different mechanisms introduced by our proposed approach called AL and NLTR to improve reliability in RPL. Section IV evaluates the performance of both mechanisms using the COOJA/Contiki simulator. Finally, we conclude the paper and present directions for future work in section V.

MOTIVATION AND RELATED WORKS

In this section, at first, we describe the essential reliability issues in native RPL. Then, we introduce some related works.

Normally, the nodes in RPL network form a tree topology automatically. But in some cases, a node may select its son node as the optimized father node resulting in a loop in which each node is the father node of the other party.

On the other hand, because no the mechanism to detect the link quality of neighbor nodes, ETX mn between node X and N is generally set a default great value. This make ms far less than mn, which may cause Rank (Xs) is less than Rank (XN). When Rank (Xs) gets minimum, node X will selects S as its best parent node who was its son node.

The other issue in RPL is about neighbor detection. In the scheme of neighbor detection of RPL, an entry of a neighbor node is assigned a lifetime value when it was added into the neighbor table. If this entry is expired, the state of the neighbor changes from REACHABLE to STALE. The action of neighbor detection is triggered by transforming the neighbor's state from STALE to DELAY when a unicast message is sent to the neighbor with STALE. As the DELAY state expired, the state of neighbor node transforms from DELAY to PROBE and a NS message will be sent to this neighbor. The neighbor node will be deemed to be available if a NA is received. Otherwise, the NS will be sent continuously until reaching the maximum of retransmission. If no NA is received during this time, this neighbor node is considered unreachable and deleted from the neighbor table. In the implement of Contiki RPL, the lifetime of a neighbor node must be expired after 300 seconds. After that if there are unicast messages need to be sent, the neighbor detection will be triggered resulting in the interactive actions of NS and NA messages. This will generate some unnecessary messages which increase network load and the probability of network congestion and reduce the reliability of routing.

RELIABILITY-ENHANCED RPL

In this section, we propose two mechanisms, Loop Avoidance (LA) and Neighbor Life Time Refresh (NLTR), to solve the problems mentioned in section III.

A. Loop avoidance

The best parent node is chosen when RPL node detects the change of the link quality. In this action, all of entries in the parent table are judged and might be selected as the optimized father node. But, the son nodes of this RPL node should not be selected, otherwise the loop will occur. If these son nodes are not in the parent nodes table, there was never an occasion for loop.

So, we propose a mechanism of loop avoidance as following:

RPL node does not add all of nodes which sent DIO into the parent node table when it receives a DIO message. This action occurs if and only if the Rank of the sending node who sent DIO is not more than its own Rank.

The reason is the Rank of the son node of a RPL node is calculated based on this RPL node, so the Rank is greater than its father node's. Those DIO senders with greater Rank than its own are not added into the parent node table so that the son nodes are not added into the parent node table and the loop could be avoided.

It might also be noted that this mechanism although decreases the loop probability greatly, but it cannot promise to completely avoid all loops. A loop might occur if a RPL node selects a new parent node and does not update in time the link quality between itself and the new parent node.

B. Neighbor life time refresh

In the implement of current Contiki RPL, except for NA reception processing by neighbor accessibility detection, there is no other mechanism can refresh neighbor survival time. In a LLN, we should decrease the unnecessary overhead of protocol as soon as possible. The first reason is to save the energy cost; the second is to avoid congestion and improve the successful ratio of packet delivery.

We propose a new scheme of neighbor life time refresh. The basic idea of this scheme is to refresh the neighbor life time through the data link layer. The details are described as follows:

We define the accessibility of a neighbor as that the two-way link between a node and its neighbor node is clear. That is to say the message sent to the neighbor node can be received correctly and vice

versa. We refresh the life time of a neighbor node when a RPL node send a unicast packet and receive an ACK in data link layer from the neighbor node, which indicates that neighbor node is accessible.

PERFORMANCE EVALUATION

In this section, we simulate the two mechanisms proposed in section IV in COOJA simulator and evaluate the performance.

A. Simulation Design

We design the simulation scenario in a 300m ×300m rectangle area. The 150 nodes are distributed in this area randomly which include one root node and 149 non-root nodes as shown in Fig. 1. The non-root nodes send a unicast packet to the root node every minute after joining DODAG.
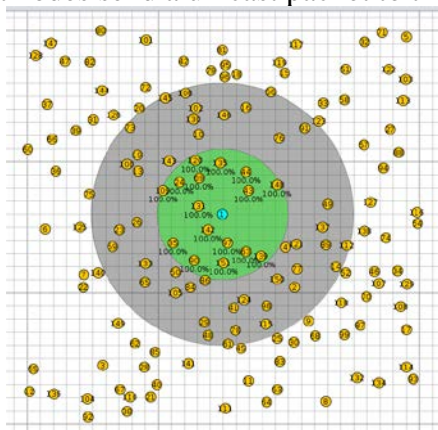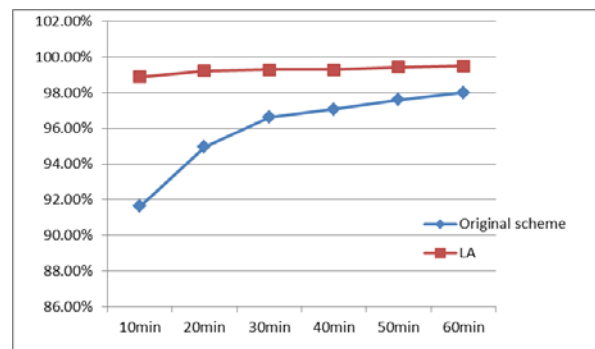

Fig.1   Nodes distribution   Fig.2 Successful delivery ratio of messages for LA

The numerical parameters are tabulated in Table I. We assume no down-ward routing in this network, i.e. there are only up-ward packets from non-root nodes to the root node. The reception successful rate is 100%.

TABLE I.   ANALYSIS PARAMETERS

Channel	Communication radius	Reception successful rate	MOP
UDGM	50 m	100%	No down-ward routing

B. Numerial Analysis

In Fig. 2, it is clear that the successful rate of packet delivery has an obviously improvement after using LA mechanism. In the original protocol, the loop occurs in the first 10 minutes so that some packets are lost and the successful rate of packet delivery is not high. As exchanging DIO many times, the loop disappears, and the successful rate of packet delivery increases gradually. However, in the scenario with LA, the loop is avoided in large part resulting in a high successful delivery rate.

Fig. 3 denotes the compare of successful delivery ratio between NLTR and original mechanism. It is clear that the successful delivery ratio with NTLR increases evidently as time increases than the original mechanism. In the original mechanism, the reason of low successful delivery ration is same as the reason in Fig.2 mentioned above. Furthermore, as the neighbor's lifetime expires, the neighbor detection messages such as NS, NA must generate between a twin nodes. This will cause the more number of protocol packets in the network and decrease the successful delivery ratio of application packets.
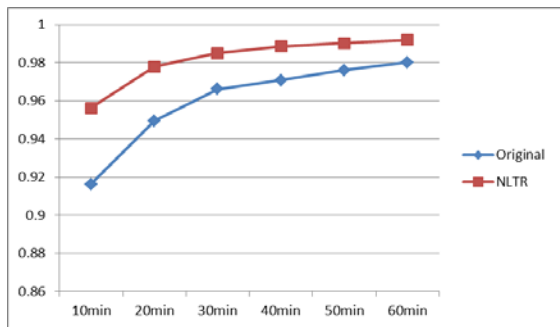
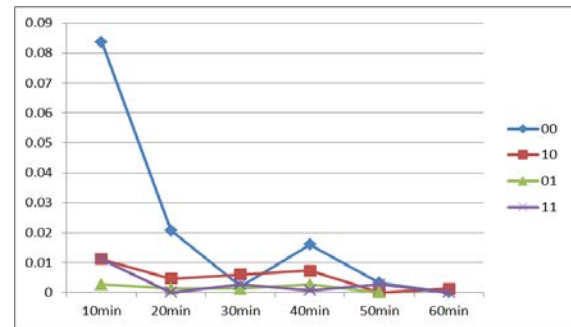Fig.3 Successful delivery ratio of messages for NLTR.　　　Fig.4 Packet loss ratio.

Fig. 4 compares the packet loss rate under the four combination cases of LA and NLTR. It denotes that the application packets are most serious at the first 10 simulation minutes. After that, the loss rate gets smooth. This due to nodes just started at the beginning of simulation, the DIO sending interval of every node is very short so that a lot of DIO message could not be sent resulting in the high loss rate. Finally, the network topology is stable gradually and the sending interval of DIO gets greater, so the loss packets become decrease.

CONCLUSION

In this paper, we investigate the principle of RPL and find existing three problems under the cases of using the neighbor detection scheme and link metric. The first is the network has some loop sometime. The second is that there is not a refreshment mechanism for the neighbor's lifetime.

In order to resolve these three problems, we propose two mechanisms, i.e. LA and NLTR. We simulate these mechanisms by Cooja platform in Contiki operation system. The simulation results show that LA and NLTR mechanisms can improve the performance of whole network obviously and no other new problems are involved. If we apply these two schemes at the same time, the network performance will be improved mostly.

ACKNOWLEDGMENT

REFERENCES

[1] N. Accettura, L. A. Grieco, G. Boggia, P. Camarda, "Performance Analysis of the RPL Routing Protocol," Proceedings of the 2011 IEEE International Conference on Mechatronics, 2011, pp. 767-772.

[2] N. T. Long, N. De Caro, W. Colitti, A. Touhafi, K. Steenhaut "Comparative Performance Study of RPL in Wireless Sensor Networks" IEEE 19th Symposium on Communications and Vehicular Technology in the Benelux (SCVT), 2012, pp. 1-6.

[3] Emilio Ancillotti, Ra_aele Bruno, Marco Conti. "RPL Routing Protocol in Advanced Metering Infrastructures-an Analysis of the Unreliability Problems", 2012, pp. 1-6.

[4] F. Osterlind, A. Dunkels, 1. Eriksson, N. Finne, and T. Voigt, "Crosslevel sensor network simulation with COOJA," in Proc. of LCN, Nov. 2006, pp. 641-648.

[5] Contiki COOJA site. [Online]. Available: http://www.sics.se/contikil.

[6] C. Boano, K. Romer, F. Osterlind, and T. Voigt. Re-alistic Simulation of Radio Interference in COOJA. In Proc. of EWSN'11, February 2011