Performance Evaluation of RPL Routing Protocol in 6lowpan

Haofei Xie , Guoqi Zhang , Delong Su, Ping Wang and Feng Zeng
Key Laboratory of Industrial Internet of Things and Networked Control, Ministry of Education,
Chongqing University of Posts and Telecommunications, Chongqing, China
zhangguoqi0101@sina.com

Abstract—**In this paper, we present the results of the comparison of three reactive routing protocols for Wireless sensor networks (WSNs), AODV, DYMO and RPL. For the purpose of performance evaluation, detailed comparisons are made with AODV and DYMO. Simulations are run to estimate the network topology change, routing overhead and average packet End-to-End Delay. We simulate RPL with COOJA based on contiki operating system. It is showed that RPL routing protocol offers adaptation to changing network topology and the performances of RPL are superior to the other two routing protocol in 6lowpan, and it has a higher stability than the AODV,DYMO.**

Keywords-comparison; RPL Routing protocol;Simulation; control overhead ;End-to-End Delay.

I. INTRODUCTION

WSN has become increasingly subject to academia and industry attention and favor. After ten years of development, there have been a large numbers of WSN protocols, especially in routing layers, such as AODV, DYMO, HiLOW protocols.etc. However, these routing protocols belong to the private protocols and are applied to specific application scenarios. In addition, the scope of their applications is narrow and it is difficult to promote due to the lack of standards.

The introduction of 6LOWPAN protocol changed this situation. In this background, the Internet Engineering Task Force (IETF) Routing Over Low power and Lossy networks (ROLL) working Group [1]was formed to develop an adapted routing solution. ROLL Working Group first evaluated the existing routing protocols, such as OSPF, DYMO, AODV, and OLSR and so on. The results showed that some of the existing routing protocols can't meet the LLN routing requirement. On the basis of routing requirements, link selection on quantitative indicators, ROLL Working Group formulates the ROLL protocol. RPL protocol is a distance vector routing protocol, nodes exchange distance vector by constructing a Directed Acyclic Graph (DAG) and the data packets route through it. DODAGs have the property that all edges are destination oriented in such a way that no cycles exist. This paper detailed analyzes the topology formation process of RPL, packet routing process and the Trickle timer, average packet End-to-End Delay and simulates RPL routing protocol to verify its performance in COOJA simulator embedded in CONTIKI OS [2] available at [3].

AODV[4] is capable of both unicast and multicast routing in Ad hoc networks. It is an on demand algorithm, meaning that it builds routes between nodes only as desired by source nodes. Dynamic On-demand MANET routing protocol (DYMO)[5] is the current engineering focus for reactive routing in the MANET. DYMO, operates very similarly to AODV, but requires only the most basic route discovery and maintenance procedures. In this study we present a performance analysis of RPL compared with AODV, DYMO.

The rest of the paper is organized as follows: Sec. II shows us an overview of the RPL protocol. Sec. III describes the performance evaluation environment and simulation reports, explains simulation results, useful to assess the validity of RPL. Finally, the last Section draws conclusions.

II. RPL ROUTING PROTOCOL OVERVIEW

A. The Format of DIO messages

RPL [6] uses the new definition of the node sending mechanism to create DODAG. RPL define three new ICMPv6 messages: DIO (DODAG Information Object), DAO (Destination Advertisement Object) and DIS (DODAG Information Solicitation).The DIO messages are sent by node, which are used to advertise DODAG messages, For instance DODAGID, OF [7], DODAG Rank, DODAG Sequence Number and other parameters. The format of the DIO Base Object is as follows in Table 1.

RPL Instance ID: 8-bit field set by the DODAG root that indicates of which RPL Instance the DODAG belongs to

DTSN: 8-bit field sent by node transmitting DIO.DTSN is used for routing downward direction maintenance

G: Used to indicate whether a DODAG is connected to the network root node, in other words, indicating whether DODAG root OF the finish

Prf: 8-bit field set by the DODAG which can design network in order to attract more nodes to join it

DODAGID):128-bit IPv6 address set by a DODAG root that uniquely identifies a DODAG.

Flags:8-bit reserved field for flags. Must be cleared when sent and will be ignored when accepted.

TABLE 1: THE FORMAT OF THE DIO BASE OBJECT

978-1-4799-3279-5 /14/$31.00 ©2014 IEEE

RPL Instance ID				Version Number	Rank	
G	0	MOP	Prf	DSTN	Flags	R
DODAGID						
Option(s) ···						

Version Number: It is controlled exclusively by the DODAG and its aim is to describe the characteristics of DODAG

Rank: 16-bit unsigned integer indicating the DODAG rank of the node sending the DIO message;

Option(s): the valid options of DIO message

B. The construction process of DODAG

Root node starts the DODAG building process by transmitting DIO messages. Neighboring nodes process DIOs and make a decision whether joining the DODAG based on the objective function and/or local policy. A node computes its Rank with respect to the root and starts advertising DIO messages to its neighbors with the updated information. As the process converges, each node in the network receives one or more DIO messages and has a preferred parent towards the sink. Hence, RPL optimizes the upward routes for multipoint-to-point traffic that accounts for most of the traffic in LLNs.

To support downward routing, RPL uses DAO control messages that give the prefix information, the route lifetime, and other information. RPL defines the storing and non-storing modes. In the non-storing mode, packets use source-routing for downward traffic. In our study, we focus on the storing mode in which each node keeps track of all accessible downlink prefixes

Simplified flow chart of the process of building DODAG shown in Figure 1, LBR sends DIO messages, node A listening LBR receives DIO will be added to the DOGAG, and replies DAO message containing its own prefix information to LBR, node A computes its Rank and starts advertising DIO messages to its neighbors with the updated information. So does node B as node A after receiving DIO from node A. The network starts, node B may receive the DIS message sent by node C, after node B adds DOGAG, node B invites node C to join DODAG, node C sends a DAO message to its parent node B, node B will integrate all information and send a DAO message to its parent node, DAO messages are progressively integrated until LBR node receives the final loopback messages including prefix information of all nodes. However, if two nodes transmit simultaneously to each other and decides to join each other, then a collision may occur. This is the reason why the DIO message received when the risk of the window will not be processed. As the random effects of the trickle timer, the later DIO messages unlikely to collide. Wherein the timer is generally used to control the flow of DIO messages, DIO messages are sent when its counter expires.

C. RPL trickle timer management

Timer management is an important part of any protocol. Most routing protocol maintain routing adjacency by sending a keep-alive message to maintain routing adjacency and sending any other necessary control packet to update routing tables without having to explicitly limit the control protocol overhead. The reason to do like this is that in traditional IP networks, the bandwidth required for the control is negligible compared to the data flow. But in the link such as LLN which is unstable and its network resources are scarce, so limiting control flow has an impact on the rapid response ability to the Internet change and the ability to maintain the synchronization. Thus for this aim, RPL utilizes DIO timer (trickle timer) called trickle algorithm [8].

Trickle algorithm uses an adaptive mechanism to control transmission rate, so that the nodes in different situations can listen to enough packets to maintain consistency. Basic idea is that when a DODAG inconsistency was detected (when a node receives a new parameter, such as the new OF, New DODAG Version Number or parent notifies a new DODAG rank, etc.) and when detecting circuit (such as when a node receives packets from his children nodes, according to its routing tables, it needs to forward this packets to the same child nodes), or a node with a new DODAGID adds to or moves to a new DODAG, it would frequently send DIO messages. When detecting DODAG inconsistencies, the node will restart its trickle timer, so that the notice of DIO messages is more frequently. With the steady of DODAG transmission frequencies, the transmission of DIO message will be reduced to control the flow. That is to say when the network changes, the node will send additional protocol control packets, and then when the network begins to stabilize, control flow rate decreases. Trickle algorithm does not require complex code and state in the network.

The creation DODAG is seen as a continuing problem for RPL, RPL will utilize trickle timer to decide when to multicast DIO messages. When detecting inconsistencies, RPL routing control messages will be sent more frequent, with the stability of the network, RPL will gradually decrease to send RPL messages.

Usually, node will restart the timer in the following cases: every time when detects DODAG inconsistencies, restarting the timer in order to increase the transmission frequency of DIO messages for quickly updating DODAG; when a new node is added to DODAG, receives a multicast DIS message from another node, moves to a new DODAG and when a node receives DIO messages reflecting DODAG changes from its parent node, when a circuit is detected (e.g., DODAG parent node receives packets which desires to be forwarded inward), when the rank of parent node changes

III. Performance Analysis of the RPL Routing Protocol

To analyze the general behavior of RPL under different conditions, we have considered a wide range of simulation settings. Sensor nodes have been randomly scattered in the sensing area and the transmission range is 50m. We have

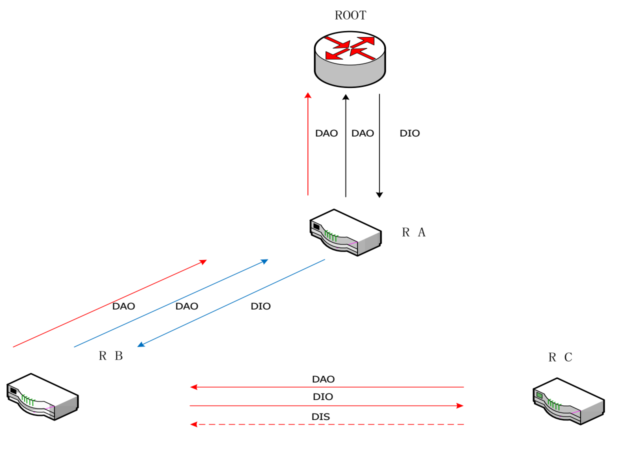

Fig.1 the construction process of DODAG

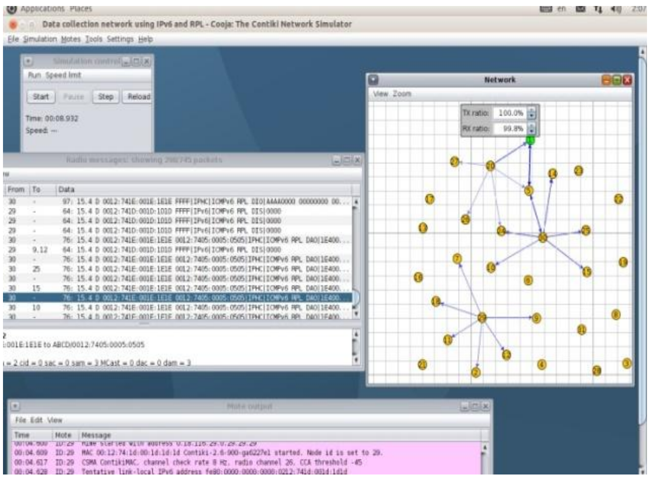

Fig.2 RPL topology in a WSN with 30 nodes

evaluated scenarios with a single sink node, assuming a squared sensing area with a side L = 200m, the number of nodes N varies in [10, 50]. For each value of N, 5 different random topologies have been simulated, thus obtaining 50 different DODAGs. Some of the topologies have been select to analyze the RPL Routing Protocol.

As shown in Fig.2, the example topologies with 30 nodes in a 6lowpan network. Through these topologies, we mainly study some characteristic simulation of RPL, including Trickle timer, Routing Overhead, Average End-to-End Delay. It's worth noting that in the relative comparison, it is necessary to set the parameters according to the paper we cite for precisely comparison.

A. The timer

To validate the Trickle timer can effectively adjust the rate of packet transmission, rapidly response to non-conformance, especially monitor the delivery interval of DIO message for all the nodes, as shown in Figure.3.With the stability of the network, the greater the transmission interval of DIO message, the less the number of messages of nodes. As during the simulation, artificially moves node 8, local repair is triggered, resulting in a fixed time delay between finding a non-conformance and the immediate reacting, what it has to done is to prevent that when all nodes detect inconsistencies at the same time reaction will cause the broadcast storm. Thus there is a fixed delay between the changes of state of the different nodes.

B. Routing Overhead

Figure 3 shows that RPL routing protocol overhead increases as the node pause time (pause time reflecting the change in the frequency of the network topology) decrease. It shows that routing protocol overhead change when network topology changes. When nodes move, there is a corresponding increase of routing overhead, the more slowly the nodes move, the less the number of routing overhead. Obviously, from the figures we can know that RPL routing overhead is higher than AODY and DYMO [9]. The reason for this situation is that when network finds a non-conformance, the Trickle timer will be triggered to carry on local routing repair which needs a large number of control packets.

C. Average End-to-End Delay

Figure 5 shows that the delay of DYMO and AODV is higher than RPL and increases with the node pause time decreasing (higher mobility of mobile nodes).In this simulation, the higher the mobility of mobile nodes, the faster the network topology changes. Obviously, from the figure we can know that AODY, DYMO protocol delay is higher than RPL

For RPL, It needs less time to obtain the necessary path so that it has low delay. The main reason for this result is that, compared with the other two protocols, RPL protocol is designed simple, and control packets of each node cost less time. So the time of establishment path also decreases.

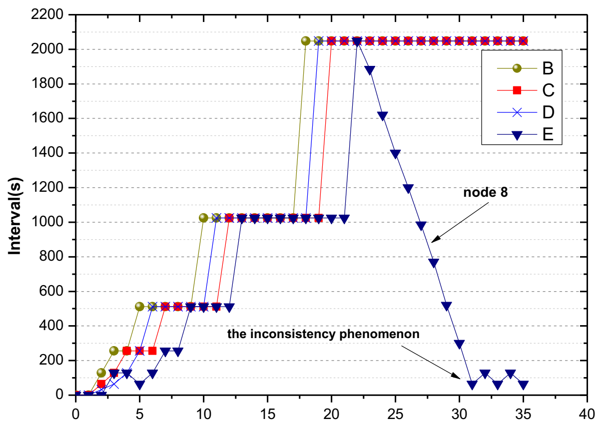

Fig.3 transmit interval changes of DIO for some nodes

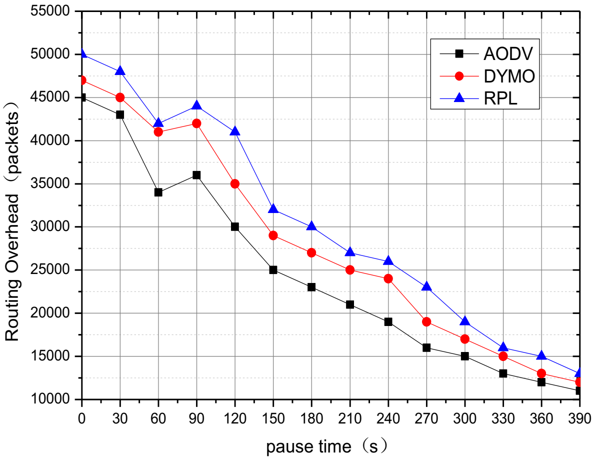

Figure.4 Simulation of routing overhead

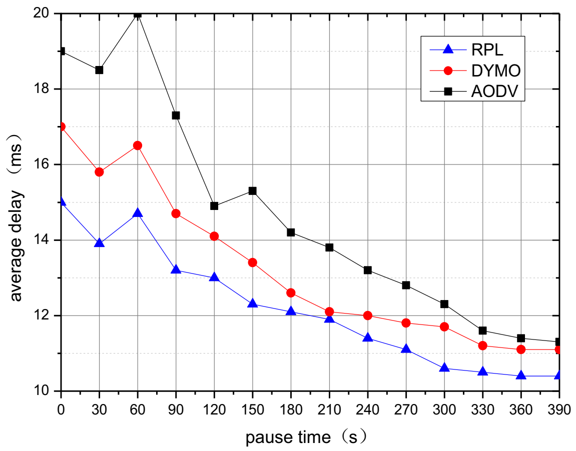

Figure.5 Simulation of the data packet average delay time

IV. CONCLUSIONS AND FUTURE WORK

In this paper, we simulated the behavior of RPL routing protocol, and compared performance with other routing protocols such as AODV, DYMO in WSNS. From the simulation study, we concluded that: （i）RPL is an efficient routing protocol and the performance of RPL is better than AODV, DYMO due to its fast network establishment, RPL can also afford critical scenarios such as rescue or military operations; （ii）its effectiveness can be further improved in terms of overhead, which can be very high. It is necessary to fulfill and study the behavior of RPL routing protocol in a very large scale (thousands of nodes) network; we can also simulate RPL from the power consumption. Future work will also compare RPL with respect to routing protocols already available for WSNs.

ACKNOWLEDGMENT

The research work was supported by National 863 project of China HQ2010AA0401028003 and China-Korea International S&T Cooperation Program under Grant No.2012DFG12240

REFERENCES

[1] ROLL working group, Feb. 2010. M. King and B. Zhu, “Gaming strategies,” in Path Planning to the West, vol. II, S. Tang and M. King, Eds. Xian: Jiaoda Press, 1998, pp. 158-176.0

[2] The Contiki Operating System, http://www. sics.se /contiki/

[3] “cooja simulator” [online]. Available:http//www.sics.se/fros/cooja.php

[4] C.E Perkins, E. Belding-Royer and S. Das, “Ad Hoc On-Demand Distance Vector (AODV) Routing” RFC3561, 2003.

[5] Chakeres and C. Perkins, “Dynamic MANET On Demand (DYMO) Routing,” draft-ietf-manet-dymo-17.txt, 2009.

[6] T. Winter and P. Thubert, RPL: IPv6 Routing Protocol for Low power and Lossy Networks, draft-ietf-roll-rpl-15 (work in progress), IETF ROLL working group, Feb. 2010.

[7] E. P. Thubert, “RPL Objective Function 0”,draft-ietf-roll-of0-03, 2010. [Online]. Available:http://tools.ietf.org/html/draft-ietf-roll-of0-03

[8] P. Levis, T. Clausen, J. Hui, O. Gnawali, and 1. Ko, The Trickle Algorithm, draft-ietf-roll-trickle-05(work in progress), IETF ROLL working group, Nov. 2010.

[9] MIAO Quan-xing and XU Lei,”DYMO Routing Protocol Research and Simulation Based on NS,”IEEE ICCASM ,2010.