

Researching and Hardware Implementation of RPL Routing Protocol Based on the Contiki Operating System

Jinghan Wang and Hongxin Li

Abstract—This paper focuses on solving the problem of Internet IP. Then we studied the RPL Routing Protocol which is suitable for this requirement, through simulation verified it. In the simulation we used 20 nodes to test, forming the DAG, RPL's control messages were verified. In order to verify the RPL, we collected the DIO, DAO and DIS messages and observed its networking process in the hardware experiments. At last, it verified the feasibility and reliability of the RPL.

Index Terms—IoT WSN, RPL routing protocol, simulation, hardware implementation.

I. INTRODUCTION

The Internet of things (IoT) is very popular now, how to introduce IP address in the Internet of things is a challenge. Wireless Sensor Networks (WSNs) are comprised of a large amount of sensor nodes, which have the function of collection, processing and transmission. Therefore these sensor nodes can fulfill requirements in specific application scenarios. It will revolutionize the human life in the upcoming years, providing at the same time new business opportunities and covering many important domains such as: smart-cities, environmental monitoring and health care [1].

Usually, the overall network load is low in WSN applications, such as the communication reliability and to the efficient use of node batteries. In particular, the routing is a crucial problem in WSNs [2], due to possible link failures, low data rates, and limited energy reserves. In fact, sophisticated techniques are required for setting up and maintaining reliable paths, as well as for promptly detecting link failures, without wasting energy and communication resources [3]. Because of the uniqueness of WSNs, traditional IP routing protocols, such as OSPF, IS-IS, AODV, OLSR, are not able to satisfy the typical requirements of multipoint-to-point WSN applications [4].

For this reason, The Internet Engineering Task Force (IETF) working group, namely Routing Over Low-power and Lossy Networks (ROLL) [5], has proposed a novel IPv6 Routing Protocol for Low power and Lossy Networks, that is RPL [6]. As stated in the draft standard [6], RPL is a gradient routing technique [7] 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.

Given the application of RPL in the real world, it is

necessary to analyze its basic network performance. using the Contiki COOJA simulator available at ,we study the DAG and the parameter of DAG how send in nodes.

The rest of the paper is organized as follows: Section II provides an overview on the RPL protocol. Section III describes the simulation methodology. Section IV through hardware programming and debugging verifies reality of RPL topology. Finally, the last Section draws conclusions.

II. RPL OVERVIEW

The Working Group focuses on standardizing RPL which is a gradient-based routing protocol for WSNs with bidirectional links. It can support a wide variety of different link layers, including ones that are constrained, potentially lossy, or typically utilized in conjunction with host or router devices with very limited resources [6].

In RPL, given a set of sinks, a DAG (i.e., a gradient) is defined as how link costs and node properties have to be combined in order to compute path costs. Link costs and node information can include available energy resources, hop count, throughput, latency, reliability, and so on [8]. In other terms, RPL minimizes the costs to reach any sink (from any sensor) by means of an objective function, RPL separates packet processing and forwarding from the routing optimization objective. In RPL, it uses four identifiers to define and maintain a topology:

- RPL Instance ID, identifying one of the possible RPL instances running on the same WSN. Each instance may serve different and antagonistic constraints or performance criteria.
- DODAGID, specifying one DODAG within one RPL instance. One RPL instance may have several DODAGs.
- DODAG Version Number, which is incremented each time the DODAG is rebuilt.
- Rank, defining individual node positions with respect to the DODAG root. Its exact calculation is left to the objective function, rank must monotonically decrease as the DODAG is followed towards the DODAG destination.

RPL provides a mechanism to disseminate information over the dynamically-formed network topology to enable minimal configuration in the nodes and to allow them to operate mostly autonomously. It uses three key words to define one DAG.

- DIO (DAG Information Option), it contains information about the rank, the objective function, IDs, and so on. They are broadcasted periodically by each node to create the DODAG.

Manuscript received January 14, 2014; revised May 16, 2014.

The authors are with Lanzhou University, Lanzhou, China (e-mail: 838174216r@qq.com, hongxinli@lzu.edu.cn).

- DAO (Destination Advertisement Object), it is used to transfer data, to announce the distance between nodes and sink.
- DIS (DODAG Informational Solicitation), nodes use it to ask DIO if they want to join in a DAG. Every node has only one sink.

The root of the DODAG starts sending DIO messages with rank equal to 1. Upon DIO reception, nodes update their rank and the cost to the sink. Each node can select its parent based on several possible rules, such as objective function, path cost, rank, and so on. A DODAG root can issue a global repair operation by creating a newer version of the DODAG. RPL also supports mechanisms, which may be used for local repair within the same DODAG version, e.g., upon loop detection.

When a node joins the WSN, it waits for a DIO message in order to discover possible parents. Optionally, a new node can multicast a DODAG Informational Solicitation (DIS) to ask for a DIO.

Finally, it is worth to note that nodes information can also be propagated upwards along the DAO. These messages are used for handling downward and peer-to-peer traffic (i.e., from sinks to the sensors and from sensor to sensor). These messages are triggered by the reception of a DIO message, or in global and local repair operations. After receiving a DAO message, each node forwards it to its parent on the expiration of a Delay DAO timer, which is implementation-dependent, according to [6].

To avoid redundancies and to control the signaling overhead, the trickle algorithm [9] triggers, for each node, a new DIO message only when the overall amount of control packets already sent in the neighborhood of that node is small enough. In the trickle algorithm, the time is split in an endless sequence of intervals with size I . A node can transmit a new DIO message at a random instant t in the second half of each interval if, since the beginning of that interval, the number of signaling messages (which have been heard) is smaller than a given threshold ORO, i.e., the redundancy constant. The size I is not fixed, but it is varied over the time in the range $[I_m, 2M \times I_m]$. In particular, starting from the minimum size I_m , I is doubled, at the end of each interval, up to a maximum number of times M . When an inconsistent state is detected (e.g., there is the detection of a loop or, the trickle timer is reset; that is, I is set to the value I_m).

III. SIMULATION SETUP

Sensor nodes use the open source operating system Contiki. Contiki is specially designed for low-power and memory-constrained devices. It contains several lightweight network mechanisms: the uIP TCP/IP stack [10], the Rime stack and the uIPv6 stack [11]. In this study, the uIPv6 stack is used, which provides IPv6 networking and contains RPL routing protocol. To study the topology of RPL under different conditions, we have considered a wide range of simulation settings. In particular, we have evaluated (using the Contiki COOJA simulator [12]) scenarios with a single sink node, assuming a squared sensing area with a side $L=100$ m. We chose the Sky mote and also considered a transmitting range $R_{tx}=50$ m for all nodes, including the

sink. Sensor nodes have been randomly scattered in the sensing area, subject to the constraint of obtaining a connected physical communication graph.

Fig. 1 and Fig. 2 show example topologies obtained for a WSN with 20 nodes. 3 of the rank value, it is the sink in the WSN, within the scope of its communication, forming the DAG. Other nodes, the scope of sink' communication, send DIS packets to ask DIO. At the same time, the sink and other nodes will send DIO message periodically, for example node 4.

Time	Mote	Message
00:01.203	ID:13	aaaa::1
00:01.206	ID:13	fe80::212:740d:d:d0d
00:01.206	ID:3	Starting 'UDP server process' 'collect common process'
00:01.207	ID:3	I am sink!
00:01.209	ID:3	UDP server started
00:01.212	ID:13	Created a server connection with remote address :: local/remote port 5688/8775
00:01.212	ID:3	created a new RPL dag
00:01.217	ID:3	Server IPv6 addresses: aaaa::212:7403:3:303

Fig. 1. ID:3 form a sink of a WSN.

The Contiki has been used as IPv6 header compression at the network layer. RPL is obviously used as routing protocol. Trickle timer parameters have been set as, $I_m= 4.096s$, $M= 8$. At the transport layer, UDP has been used. Node ranks have been defined using the Expected Transmission Count (ETX) as suggested in [13].

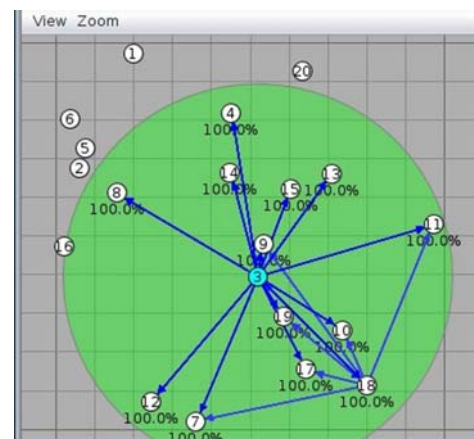


Fig. 2. Topology of a WSN with 20 nodes in COOJA.

IV. HARDWARE IMPLEMENTATION

In order to verify the implementation of RPL protocol, we through the hardware programming and debugging to test and verify. The following was what we have done.

A. Building the Experiment Platform

Experiment used SM2530 wireless sensor module, composed of 80c51 and cc2530. Nodes communication by wireless way and they use batteries. Through the serial line Internet Protocol SLIP for the exchange of data. In this experiment, we randomly placed four SM2530 modules in different places (in the space of $4m \times 15m$), checking the network networking process and showing its topology structure.

B. Analysis of Experimental Data

To observing the network structure (DAG), the sink node with USB powered is connected to the computer, the other three nodes with batteries were placed different positions, through the WSN Monitor showed its topology structure. Open the switch of four nodes, after about 5 minutes; we

can see its structure. Fig. 3 shows a simple DAG.

The red node is the sink, taking the dark blue node for example, the 0xEB77 is IPv6 address of a node which is the last 2 byte, the rest are not displayed. And 10:02:17 is the time, indicating the WSN Monitor software a recent to receive what time the node data. 38°C is not the temperature here, 38 indicates the serial number of the packets sent. Each node has serial number count, its serial number increasing. If each upload packet can reach the WSN Monitor, the number is continuous; if found discontinuous, then there is packet loss, which can judge the stand or fall of the network and environment, bringing convenience to debug and diagnosis network.

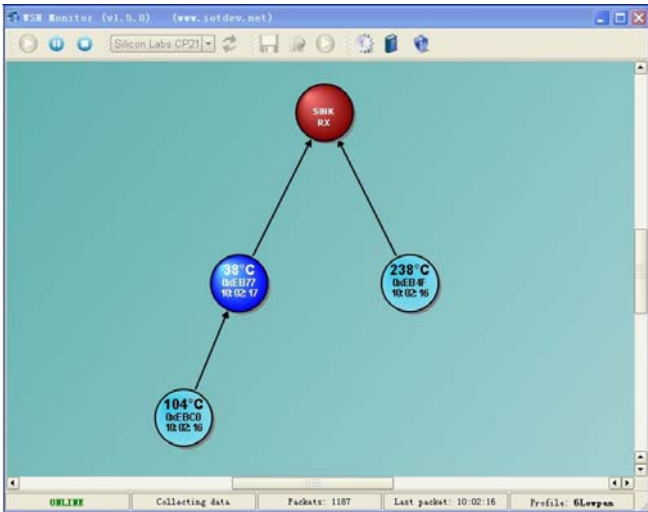


Fig. 3. Topology of a WSN with 3 real nodes.

After observing the structure, we further analyzed the collected data, different from the former, four nodes used batteries, still can form the network topology (the form is mainly to observe topology). Though the Wire shark software we can analyze RPL's control messages. Fig. 4 shows.

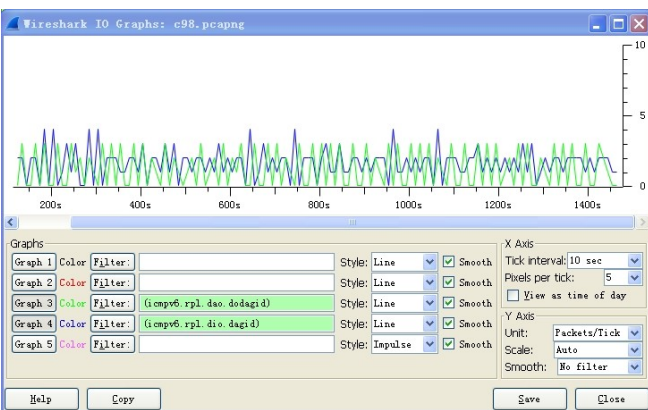


Fig. 4. The statistics of data.

In the experiment, four nodes constantly exchange data, from node to node and node to sink and RPL with ICMPv6 protocol controls the message, through the UDP communication sending data. In Fig. 4 the line represents the child nodes return the DAO to its parent node; the blue line represents the four nodes in periodically send DIO to its neighbor nodes. Receiving the DIO messages when opened the node, the nodes would not send DIS messages.

We can see the messages of DIO and DAO in Fig. 5 and Fig. 6.

```

2406 1429.50000000 fe80::212:4b00:2f5:ebc0 ff02::1a ICMPv6
2407 1432.140625000 fe80::212:4b00:2f5:eb4f ff02::1a ICMPv6
2414 1435.312500000 fe80::212:4b00:2f5:eb4c ff02::1a ICMPv6
2415 1435.312500000 fe80::212:4b00:2f5:eb4c ff02::1a ICMPv6
...
Frame 2407: 97 bytes on wire (776 bits), 95 bytes captured (760 bits) on interface 0
IEEE 802.15.4 Data, Dst: Broadcast, Src: TexasIns_00:02:f5:eb:4f
6LOWPAN
Internet Protocol Version 6, Src: fe80::212:4b00:2f5:eb4f (fe80::212:4b00:2f5:eb4f), dst: ff02::1a
Internet Control Message Protocol v6
Type: RPL Control (155)
Code: 1 (DODAG Information Object)
Checksum: 0x43c3 [correct]
RPLInstanceID: 30
Version: 240
Rank: 384
Flags: 0x10
Destination Advertisement Trigger Sequence Number (DTSN): 123
Flags: 0x00
Reserved: 00
DODAGID: aaaa::1 (aaaa::1)
    
```

Fig. 5. DIO message.

The sink node (fe80:: 212: 4b00: 2f5:eb4f) send DIO message to its neighbor nodes, determining whether a node want to join.

```

2351 1395.140625000 fe80::212:4b00:2f5:eb77 fe80::212:4b00:2f5:eb4c ICMPv6
2391 1422.218750000 fe80::212:4b00:2f5:eb77 fe80::212:4b00:2f5:eb4c ICMPv6
2393 1423.031250000 fe80::212:4b00:2f5:ebc0 fe80::212:4b00:2f5:eb4c ICMPv6
2398 1425.263625000 fe80::212:4b00:2f5:eb4f fe80::212:4b00:2f5:eb4c ICMPv6
2419 1439.343750000 fe80::212:4b00:2f5:eb4f fe80::212:4b00:2f5:eb4c ICMPv6
2423 1439.718750000 fe80::212:4b00:2f5:ebc0 fe80::212:4b00:2f5:eb4c ICMPv6
2425 1440.234375000 fe80::212:4b00:2f5:eb77 fe80::212:4b00:2f5:eb4c ICMPv6
...
Frame 2351: 76 bytes on wire (608 bits), 74 bytes captured (592 bits) on interface 0
IEEE 802.15.4 Data, Dst: TexasIns_00:02:f5:eb:4c, Src: TexasIns_00:02:f5:eb:77
6LOWPAN
Internet Protocol Version 6, Src: fe80::212:4b00:2f5:eb77 (fe80::212:4b00:2f5:eb77), dst: ff02::1a
Internet Control Message Protocol v6
Type: RPL control (155)
Code: 2 (Destination Advertisement Object)
Checksum: 0x34az [correct]
RPLInstanceID: 30
Flags: 0x40
Reserved: 00
DAO Sequence: 111
DODAGID: aaaa::1 (aaaa::1)
ICMPv6 RPL option (RPL Target aaaa::212:4b00:2f5:eb77/128)
ICMPv6 RPL option (Transit Information)
    
```

Fig. 6. DAO message.

This message is that parent node (fe80:: 212: 4b00: 2f5eb77) send to child node (fe80::212:4b00:2f5:eb4c), consistent with DAG figure.

V. CONCLUSION

This paper designed and realized RPL networking function of the agreement. Introducing RPL to IoT can realize IP address. Through the experiment, we collected data and analyzed them based on the parameters of RPL. Experimental results show that four independent node can achieve a normal networking process, accurate and reliable data transmission, this process is completely conforms to the theory of RPL.

REFERENCES

- [1] M. Dohler, D. Barthel, R. Maraninchi, L. Mounier, S. Aubert, C. Dugas, A. Buhig, R. Pagnat, M. Renaudin, A. Duda, M. Heusse, and R. Valois, "The ARESA project: facilitating research, development and commercialization of WSNs," in *Proc. IEEE Conf. on Sensor, Mesh and Ad Hoc Commun and Networks*, 2007, pp. 590-599.
- [2] Q. Lampin, D. Barthel, and F. Valois, "Efficient route redundancy indag-based wireless sensor networks," *IEEE Wireless Commun. and Networking*, WCNC, 2010.
- [3] T. Watteyne, A. Molinaro, M. G. Richich, and M. Dohler, "From MANET to IETF ROLL standardization: a paradigm shift in WSN routing protocols," *IEEE Commun. Surveys and Tutorials*, pp. 688-707, 2011.
- [4] J. Tripathi, J. C. D. Oliveira, and J. Vasseur, "A performance evaluation study of RPL: Routing protocol for low power and lossy networks," in *Proc. the Conference on Information Sciences and Systems*, 2010, pp.1-10.
- [5] IETF WG. Routing Over Low Power and Lossy Networks. [Online]. Available: <http://www.tools.ietf.org/wg/roll/>.

- [6] RPL: IPv6 Routing Protocol for Low Power and Lossy Networks, 2010.
- [7] T. Watteyne, K. Pister, D. Barthel, M. Dohler, and A. Blum, "Implementation of gradient routing in wireless sensor networks," *IEEE Globecom*, pp. 1-6, 2009.
- [8] J. Vasseur and M. Kim, *Routing Metrics used for Path Calculation in Low Power and Lossy Networks*, 2010.
- [9] P. Levis, T. Clausen, J. Hui, and O. Gnawali, *The Trickle Algorithm*, IETF RFC 6206, 2010.
- [10] A. Dunkels. (2010). Full TCP/IP for 8-Bit Architectures. [Online] Available: <http://www.doi.acm.org/10.1145/1066116.1066118>
- [11] J. Zhao and R. Govindan. (2013). Understanding Packet Delivery Performance in Dense Wireless Sensor Networks. [Online]. Available: <http://www.doi.acm.org/10.1145/958491.958493>.
- [12] F. Osterlind, A. Dunkels, I. Eriksson, N. Finne, and T. Voigt, "Crosslevel sensor network simulation with COOJA," in *Proc. the LCN*, Nov. 2006, pp. 641-646.
- [13] J. Vasseur and M. Kim, *Routing Metrics used for Path Calculation in Low Power and Lossy Networks*, Nov. 2010.



Jinghan Wang was born in 1990. She is from Information and Communication Engineering Laboratory, Lanzhou University. Her research interests include wireless sensor network of RPL Routing Protocol, and carries on its improvement. She has attended the seminar of the direction of power line carrier in 2013. she takes participated in the Longi Silicon project, involved in the research process of pulling of monocrystalline silicon.

Hongxin Li is an associate professor and a master tutor. His research interests include industrial automation and industrial communication network. He had received the national nature science funds. He has published many articles in journal meeting and is responsible for campus multiple projects, He is responsible for Czochralski silicon single crystal of Xi'an LONGi Silicon Materials Corp. He is the leader of the project and as a leader, he launched a wireless sensor routing for the application of the research direction in the direction of power line carrier.