ICIC Express Letters
Volume **11**, Number **5**, May **2017**

ICIC International ©2017 ISSN 1881-803X
pp. **927–938**

F-RPL: AN OPTIMIZED RPL ROUTING PROTOCOL

Bing Hu and Zhixin Sun*

College of Internet of Things
Nanjing University of Posts and Telecommunications
No. 66, New Mofan Road, Nanjing 210003, P. R. China
*Corresponding author: sunzx@njupt.edu.cn

Received November 2016; accepted February 2017

Abstract. *To solve the problem that RPL routing protocol cannot complete routing discovery and link quality update fairly and efficiently in the process of establishing and maintaining the network topology, an optimized RPL routing protocol (F-RPL) is proposed in this paper. F-RPL improves the Trickle algorithm by introducing a new variable, which is used to record the number of times the node is suppressed to send. In order to realize the fairness of Trickle algorithm, the priority of the next transmission is decided according to the value of the variable. Meanwhile, an active neighbor detection mechanism is proposed to improve the routing quality of the network. Simulation results show that the F-RPL protocol has a significant improvement in business packets delivery rate, network formation time and routing reliability when compared with RPL.*
Keywords: Wireless sensor networks (WSNs), Routing protocol for LLNs (RPL), Neighbor detection

1. **Introduction.** Over the past few years, the application fields of wireless sensor networks (WSNs) involve military, environmental protection, equipment manufacturing, agricultural monitoring and other industries. Since intelligent nodes of a variety of access networks present explosive growth, the concept of Internet of Things (IoT) is becoming more and more clear. For the access of massive intelligent nodes, IPv6 technology can bring a large amount of address space for WSNs, while providing efficient and convenient network discovery and address automatic configuration. 6LoWPAN (IPv6 over Low-power Wireless Personal Area Network) workgroup is founded by IETF in November 2004, which aims to solve the problems brought by the application of IPv6 technology in WSNs.

Wireless sensor network is a typical Low-power and Lossy Network (LLN). The connection between sensors is through a variety of low-power links, such as IEEE 802.15.4, Bluetooth, low-power WI-FI. Since low-power network has the characteristic that network data is easy to lose, it has specific requirements for its routing. In order to solve the routing problem in 6LoWPAN, IETF founded the ROLL (Routing over Low-power and Lossy Networks) workgroup in 2008. Afterwards, the workgroup put forward RPL (Routing Protocol for LLNs) protocol [1].

RPL routing protocol is designed for lossy links and energy constrained equipment. It is a distance vector routing protocol, which is based on the destination of the distance to determine the best path. The RPL routing protocol is analyzed in detail in [2], and the performance of the protocol is tested based on the Contiki operating system. Since the original scheme of RPL is designed for static networks, and it does not support the dynamic characteristics of network nodes, [3] proposes Co-RPL to support the dynamic characteristics of networks. In order to improve the routing performance of RPL, a new broadcast mechanism initiated by the root node is proposed [4], which makes full use of the tree network topology composed by RPL. In [5], a way to extend IEEE 802.15.4 MAC protocol is proposed. Based on the modified MAC protocol, RPL can support multiple paths to forward packets. In order to support the automatic configuration and multi-channel utilization of WSNs based on RPL, connectivity detection and channel scan mechanism are proposed in [6].

One of the core design principles of RPL is to minimize the routing control overhead and signaling data while reducing the energy consumption and improve the reliability of the protocol. For this purpose, RPL uses the Trickle algorithm mechanism to reduce the message interaction and the network burden, and to control the packet transmission frequency adaptively. If the parameters of Trickle algorithm are not set appropriate, the message inhibition mechanism of Trickle will lead to suboptimal path, especially in the network with uneven network density distribution. And it is mainly caused by the unfairness of Trickle algorithm mechanism [7]. From a mathematical point of view, a probability model is proposed to improve the fairness of the node to transmit data [8]. In [9], an adaptive Trickle mechanism is presented to reduce the energy consumption of nodes. However, Trickle algorithm still have the problems of unbalance load and suboptimal path, which are caused by the unfair transmission probability of nodes.

Because the link in RPL network must be bidirectional, in some special scenarios, some links may exhibit asymmetric properties. Therefore, before selecting a node as the parent, it is necessary to verify the reachability of the node. The standard RFC6550 document of RPL is recommended to use the Neighbor Unreachability Detection (NUD) mechanism, which is defined in [10], to complete this function. In the standard neighbor detection mechanism, two conditions are needed to trigger the neighbor reachability detection: the status of the neighbor is STALE and there has been unicast packet that is sent to the neighbor. However, unicast packet is only sent and received between RPL's parent node and child node, so the RPL node can only get the link information of its optimal parent node and child nodes. This will lead to a lower perception of the other nodes around, so that it cannot find a better parent node in fact.

Therefore, in order to solve the problem that RPL routing protocol cannot complete routing discovery and link quality update fairly and efficiently in the process of establishing and maintaining the network topology, an optimized RPL routing protocol (F-RPL) is proposed in this paper, and the paper has the following contributions. 1) In order to realize the fairness of Trickle algorithm to the most extent, a new variable is introduced to record the number of times the node is suppressed to send, and the priority of the next transmission is decided according to the value of variable. 2) An active neighbor detection mechanism is proposed to improve the routing quality of the network.

The remainder of the paper is structured as follows. The concept and process of RPL are illustrated in Section 2. The detailed technical scheme of F-RPL routing protocol is presented in Section 3. Afterwards, in Section 4, simulation analysis of F-RPL routing protocol is carried out. Finally, conclusions are drawn in Section 5.

2. System Model.

RPL is based on IPv6 routing protocol, which is specially designed for LLN. It is a distance vector protocol and runs on the IEEE 802.15.4 PHY and MAC layer. It is primarily designed for data aggregation type scene where nodes intermittently send messages to the sink node. RPL establishes a Destination-Oriented Directed Acyclic Graph (DODAG) with only one root (i.e., out-degree is zero) through Objective Function (OF), Routing Metric and Routing Constraint. This section will focus on the constructive process of DODAG and the limitations of RPL routing protocol.

2.1. Constructive process of DODAG.

The nodes in the network are built into a tree topology by RPL, and each of the tree is called a DODAG. Each DODAG has exactly one root node (called DODAG root). RPL routing protocol uses four identifiers to establish and maintain DODAG, which are RPLInstanceID, DODAGID, DODAGVersionNumber and Rank. RPLInstanceID and DODAGID are used to identify a DODAG in the network, because there may be a number of different instance and DODAG or a node may belong to more than one instance. DODAGVersionNumber identifies the new and old state of the network. DODAG can check and repair the network according to this value. The value of Rank indicates the location of a node in the network, which is the measurable value of the distance to the root node.

The constructive process of DODAG is shown in Figure 1. Select the Expected Transmission count (ETX), which is the number of times required to transmit a packet successfully, as a routing metric. In order to create a DODAG network, root node periodically broadcasts DODAG Information Object (DIO) control packet to annunciate the existence of the network. DIO packet contains a series of configuration and network information. The node who wants to join this DODAG must use the same parameters. After receiving the DIO packet, the nodes need to calculate their Rank information in the network according to the OF and join the network. Based on the calculated Rank value, the RPL node selects an optimal parent node from the neighbor nodes to obtain the optimal uplink path to the root node, and considers the parent node as the default next hop. To establish the whole network topology as soon as possible, after calculating its location, the node needs to broadcast the latest DIO packet. At the same time, the nodes that do not join the network send the DODAG Information Solicitation (DIS) request packet to obtain the network information as soon as possible. Root node, as the network manager, has the right to trigger the entire network updates, which are including updating the network information field carried in DIO or maintaining and repairing the DODAG network. It is easy to find that DIO packet is unlike the Hello packet of the traditional mobile ad hoc networks. RPL node does not verify whether itself can communicate with the sender of DIO directly, nor measures the link adjacency with its neighbors through packets.

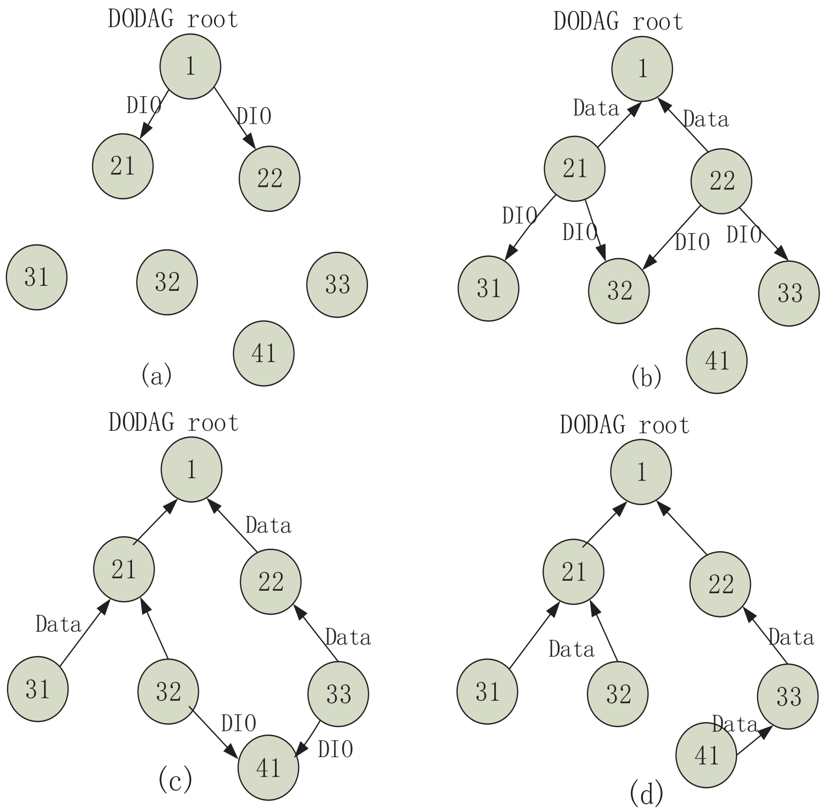

Figure 1. Uplink routing constructive process of DODAG

2.2. **Limitation of RPL routing protocol.** In order to reduce the energy consumption at design time, RPL routing protocol drastically restricts the quality of routing. Therefore, RPL routing protocol cannot complete routing discovery and selection fairly and efficiently in the process of establishing and maintaining the network topology, which leads to the selection of suboptimal path. Especially the problems existing in Trickle timer mechanism and neighbor detection mechanism affect the reliability of routing and the performance of network, which lead to the increase of network energy consumption and bandwidth consumption. Next, improved schemes will be proposed to solve the problems that exist in Trickle timer mechanism and neighbor detection mechanism.

3. **F-RPL.**

3.1. **Optimization strategy of Trickle timer.** Timer mechanism is the key component of RPL protocol. The timer used for controlling the transmission of DIO control packet is based on the Trickle algorithm. Trickle algorithm uses an adaptive strategy to control the packet transmission frequency effectively. When the network changes suddenly, the time interval for transmitting DIO control packet reduces to the preset minimum appropriate value rapidly, so that the node will send more control packets according to the timer to find, select or repair the network. Afterwards, when the DODAG network tends to a stable state, the transmission time interval will increase under a certain algorithm. Therefore, Trickle algorithm can adapt to the network state through adjusting the transmission frequency of control packet dynamically to reduce the energy consumption and network bandwidth effectively.

Trickle algorithm defines three global variables: min interval ($I_{\min}$), max interval ($I_{\max}$) and redundant constant (k). In addition, Trickle algorithm also defines 3 variables to maintain the current state of the node: the current interval (I), the current time value (t) and redundant event counter (c).

In the DIO timing Trickle algorithm of RPL, the node initializes the parameters Trickle algorithm, $I_{\min}$, $I_{\max}$ and k, according to the obtained DIO packets. Set $c = 0$, $I = I_{\min}$ and t is a random value in the interval $[I/2, I]$. Whenever a node receives a DIO packet from the parent node in the DODAG network, it compares whether the DIO packet is redundant. If it is, the redundancy counter c will add 1. At the sending time t, the value of c will be compared with the RPL constant k. If c is great than or equal to k, Trickle will suppress the transmission of the packet; otherwise it is allowed to transmit the packet. Once the current interval I has expired, I will double automatically, which indicates that a new DIO control interval is generated. At this point, counter c is set to 0, t re-selects a new random value from the sending interval. The current interval I is doubled in accordance with this law until reaching the maximum value $I_{\max}$. The algorithm can ensure that each node has the same transmission probability in a long time. However, limited to various lossy characteristics of LLN link, the network is likely to change in this period of time. In view of the problems of uneven load and suboptimal path caused by the unfair transmission probability of nodes in the Trickle algorithm, we propose an optimized Trickle algorithm to ensure that all nodes have more equitable management mechanism in a short time, which is more beneficial to find the available path.

In order to achieve the fair mechanism of Trickle algorithm to the greatest extent, we determine the priority of the next transmission according to the number of times the node is suppressed. The longer the node does not send packets, the higher the priority is of the next time to send the packet. Therefore, we specify that each node maintains a variable sum, which is used to record the number of time intervals that the node has not sent DIO packets continuously. At time t, if a node fails to send a DIO packet, sum adds 1; if it sends a DIO packet during the current interval, sum is reset to 0. Each node is to obtain its send priority of each round through maintaining the value of sum. Whenever a new time interval starts, sending time t is selected from the $[I/2^{sum+1}, I]$ randomly. In addition, when a node fails to send the DIO packet for two consecutive time intervals, the node must send the DIO packet at a random time t in the third interval; at the same time, the value of sum is reset to 0. The pseudo-code of the optimized Trickle algorithm is shown in Algorithm 1.

Algorithm 1 Optimized Trickle algorithm

function Trickle-Initialization()
 $I = I_{\min}$, $I_{\max} = I_{\min} \cdot 2^8$, $k = 3$
 $sum = 0$
function DIOIntervalBegins()
 $c = 0$
 $t = random(I/2^{sum+1}, I)$
function ConsistentTransmissionReceived()
 $c = c + 1$
function TimerExpires()
 if($k \geq c || sum == 2$) then
 Transmit DIO
 $sum = 0$
 else
 $sum = sum + 1$
 end if
function DIOIntervalEnds()
 $c = 0$
 if InconsistentTransmissionReceived
 $I = I_{\min}$
 $sum = sum + 1$
 else
 $I = I \times 2$
 if $I_{\max} \leq I$ then
 $I = I_{\max}$
 end if
 end if

3.2. **Optimization strategy of neighbor discovery mechanism.** Neighbor detection is an important function of Neighbor Discovery Protocols (NDP), and it is also the basis of RPL routing protocol. RPL uses the Neighbor Unreachability Detection (NUD) of NDP to complete the detection of link quality and neighbor connectivity. NUD defines various states of the neighbor: 1) INCOMPLETE state, which indicates that the neighbor state cannot be obtained and the neighbor link layer address resolution is not completed; 2) REACHABLE state, which indicates that the neighbor node is reachable and has the ability of communication; 3) STALE state. This state shows that the neighbor information is not clear and needs to be detected again; 4) DELAY state, which indicates that it is about to detect the neighbor node; 5) PROBE state, which indicates that it is being executed.

The neighbor detection mechanism of NDP is shown in Figure 2. Each neighbor of the node has a survival time. When the timer time is up, the state of neighbor will change from REACHABLE to STALE. In this situation, only when the node has a packet transmission demand, the NUD will be triggered. Then, the neighbor state turns into DELAY state, and this state will be controlled by the timer for some time. Subsequently, enter the PROBE state. Afterwards, the node tries to send Neighbor Solicitation (NS) packet to

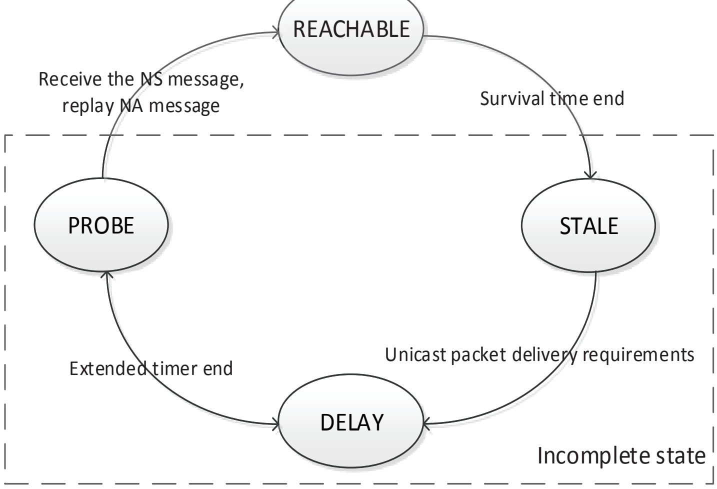

Figure 2. State transition diagram of the NUD

the neighbor node. If the neighbor node receives the NS packet and replies to the Neighbor Advertisement (NA) packet, it is proved that the neighbor is still reachable. Subsequently, the node will reset the neighbor to Reachable state and refresh the neighbors survival time. If the neighbor node does not reply to the NA packet and the NS retransmission timer expires, the node sends NS packet again. This step is executed until the number of times for sending NS packet reaches the upper limit, and the neighbor node is disconnected from the node. If the neighbor node that is disconnected is the node' parent, the node needs to select a new optimal parent node from the reachable neighbor nodes to maintain the connectivity of the network. Assuming that there are many neighbor nodes that are not reachable, it indicates that the network has some problems and network repair is required.

In the aforementioned neighbor detection mechanism, there need two conditions to trigger the NUD: the state of the neighbor is STALE and there has unicast packet that is sent to the neighbor. And the unicast packets can only be received and transmitted between the parent and child. Therefore, RPL node can only obtain the link information of its optimal parent node and its children, which leads to low awareness of other nodes around, so that it cannot find a better parent node.

In order to detect the link state information of the surrounding neighbor nodes, we remove the option that "there has unicast packet that is sent to the neighbor" and send multicast NS packets to the surrounding neighbors to detect the reachability of the neighbor. At the same time, combined with the optimized Trickle algorithm, it can perceive the dynamic change of the network state. If the sending interval of DIO control packet is very small, it indicates that the network appears to change. At this moment, it also needs to perceive the link state of the surrounding neighbors to obtain a better parent node. Therefore, when a node receives a unicast DIO control packet from a neighbor, the neighbor also triggers the NUD to determine the reachability of uplink routing. Once the neighbor's survival time expires, as long as it satisfies one of the conditions: the state of the neighbor is the STALE state or receive a unicast DIO control packet, the NUD will be triggered. This detection method is called active detection mechanism, and its neighbor state transition diagram is shown in Figure 3.

4. **Simulation Analysis.** Contiki is an embedded operating system (OS) which supports 6LoWPAN protocol stacks. Cooja simulation tool is selected in the simulation experiment, which is a network simulation software that is specially designed for Contiki OS. It is able to develop the routing protocol for low power network on the Contiki OS. Set the simulation area of 6LoWPAN network is a square area with 100m^2. In the network, there

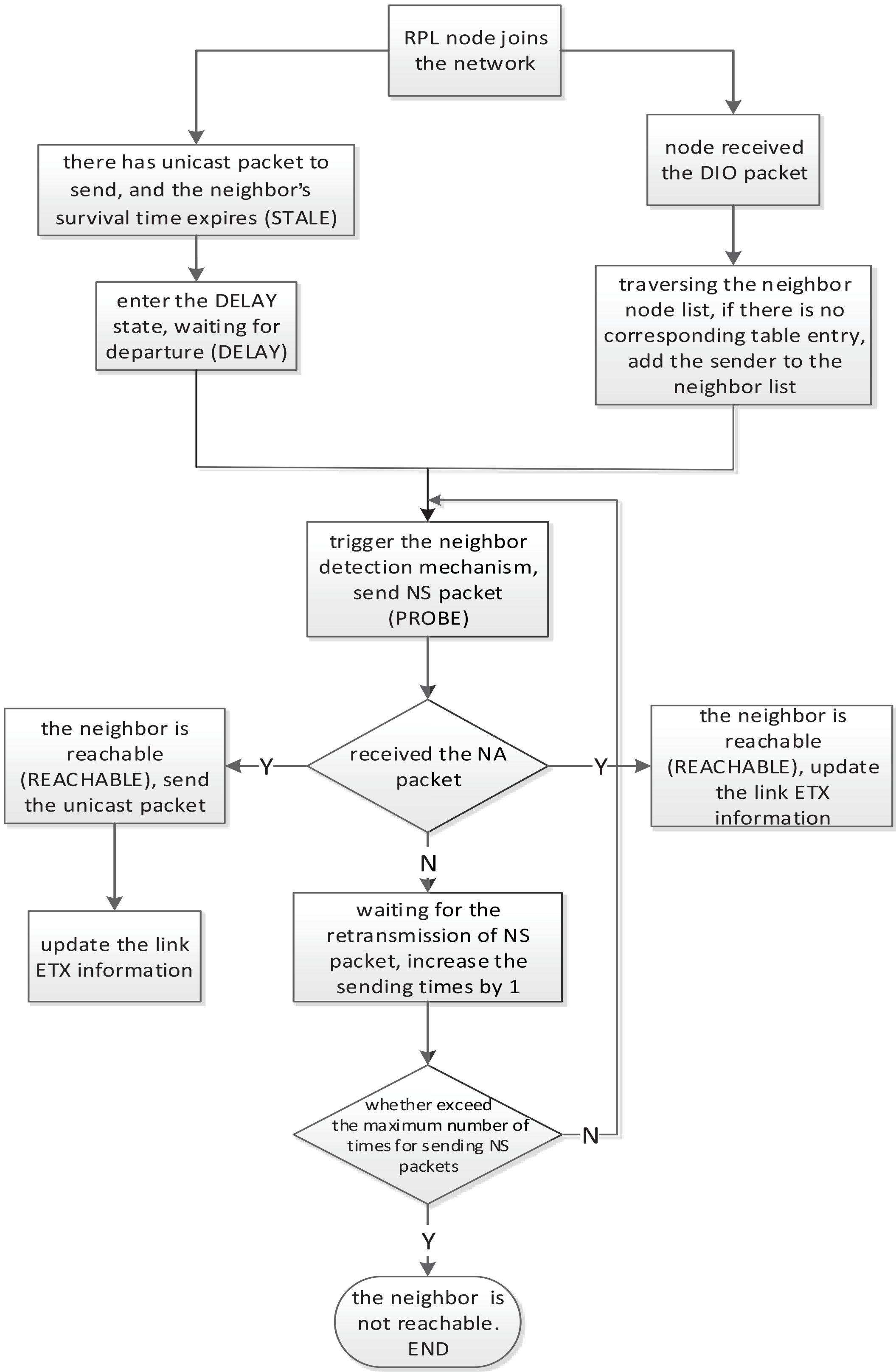

Figure 3. Optimized neighbor detection mechanism

is a border router LBR to manage the network and several common routing nodes. The parameters of Trickle algorithm are set as follows: $I_{\min} = 2^8$ms, $I_{\max} = 2^{8+8}$ms and redundant constant $k = 3$. In order to restore the real application environment, select the Unit Disk Graph Radio Medium (UDGM) as the wireless communication channel model. Set the reliable communication range of nodes is 40m and the interference range is 80m. The contents of the simulation test are as follows.

1) **Successful business packets delivery rate.** It is the ratio of the number of packets received successfully by the destination node to the total number of packets sent by the source node.
2) **Average energy consumption of nodes.** The average energy consumption of node reflects the network survivability and stability.
3) **Average initial formation time of the network.** The initial formation time of the network is defined as the time that all nodes in the network are joined by DODAG and can send business packets. This moment, the topology structure of the network is not necessarily optimal.
4) **Average link quality of the network.** This index is defined as the average measure value of all the links of a path in the network.

Figure 4 shows an example of the simulation scene graph set up by the Cooja simulator.

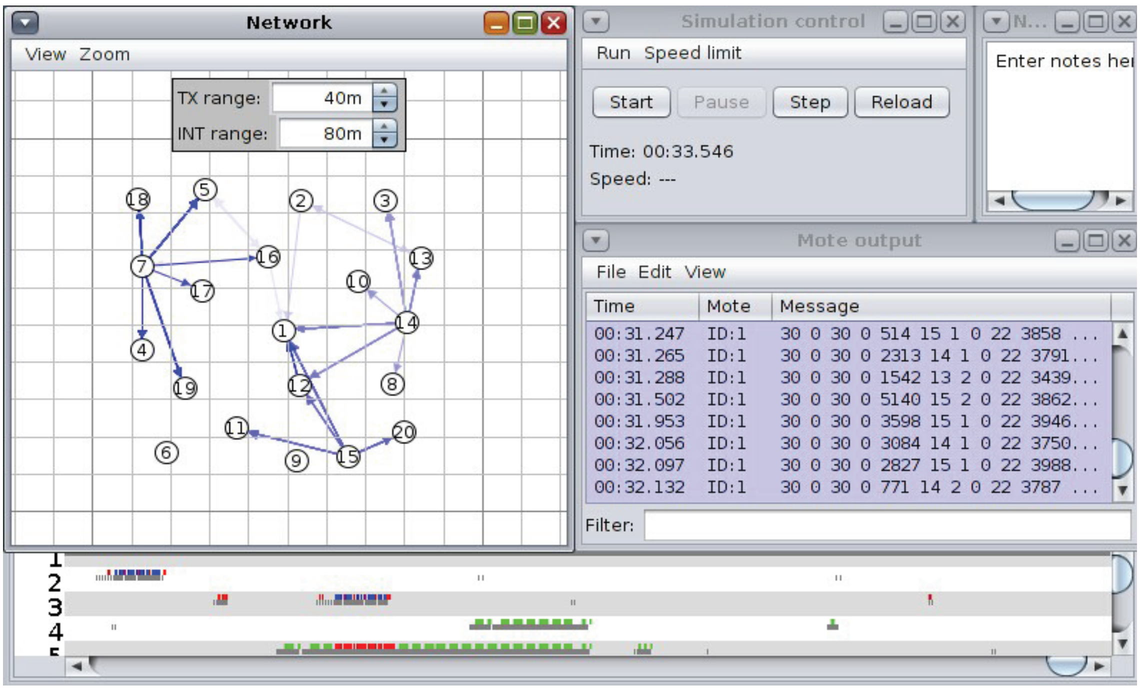

FIGURE 4. Cooja simulation scene graph

Scene 1. Test the effect of RPL protocol on the network performance under different node densities.

Set the number of nodes in the network is 8, 12, 16, 20, 24 and the time interval for sending a business packet is 10s.

It can be seen directly from Figures 5 and 6 that F-RPL can form an efficient routing topology faster in the network initialization phase and can ensure the effective delivery of business packets and reduce the packet loss rate. Benefiting from the optimized Trickle algorithm, nodes can be more efficient to spread and find more routing information, so as to achieve a better routing to improve the routing quality of the network. However, as shown in Figure 7, since the optimized Trickle algorithm needs more control packets in the initial stage of network formation, the energy consumption is slightly higher than that of the original scheme.

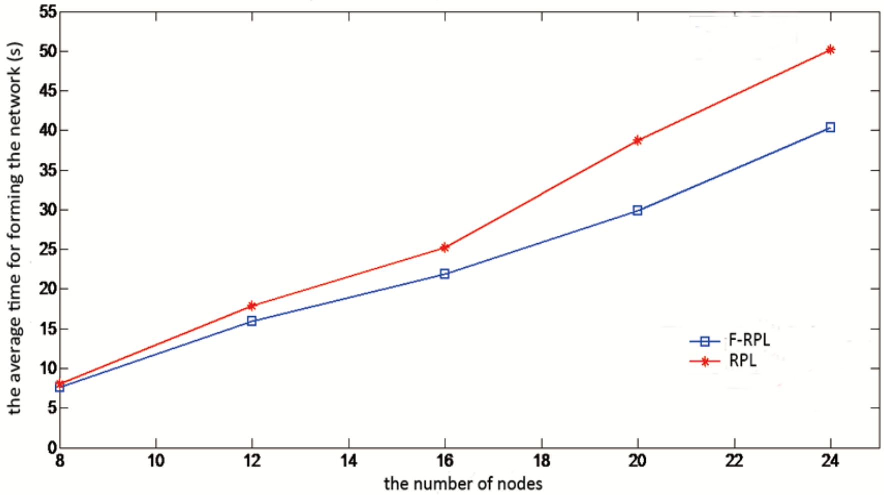

FIGURE 5. The time required for all nodes to join the DODAG

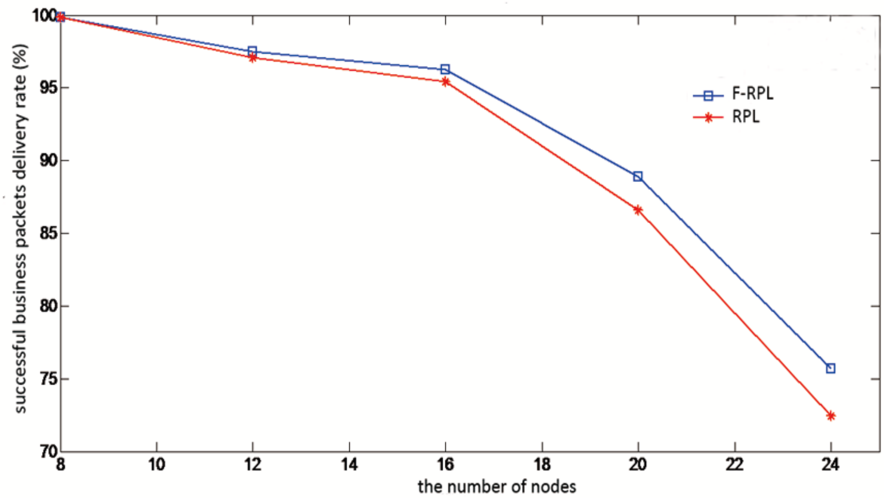

FIGURE 6. The delivery rate of the business packets during the simulation time of 3min

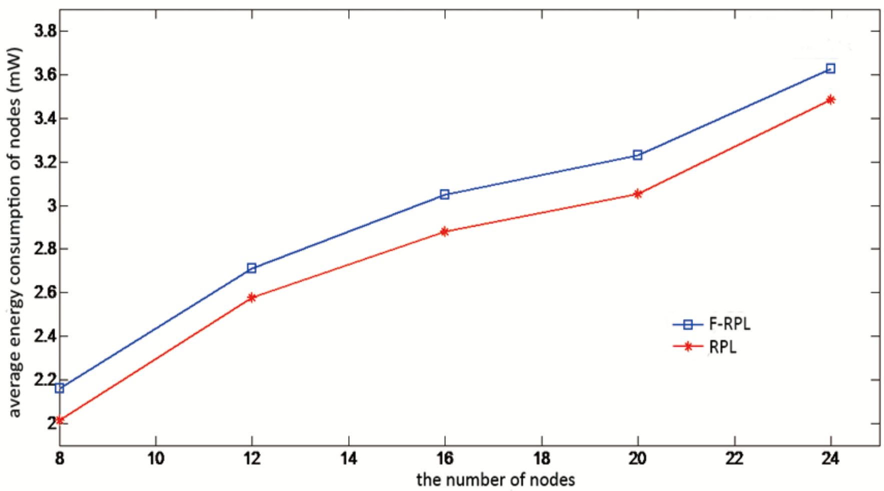

FIGURE 7. The average energy consumption of nodes per unit time during the simulation time of 3min

Scene 2. Test the network performance and routing quality under different simulation time.

Assume that there are 20 nodes randomly distributed in the range of 100m × 100m and LBR is located near the middle of the network. All nodes are connected to LBR via DODAG and the time interval for sending a business packet is 20s. When the network topology is formed, the network performances, which include successful business packets delivery rate, average energy consumption of nodes per unit time and average link quality of the network, are recorded in different simulation time.

As shown in Figures 8-10, in the process of establishing the DODAG topology tends to stability, the successful business packets delivery rate and average link quality of the network of F-RPL are improved when compared with RPL. However, the energy consumption of F-RPL is higher than that of RPL. With the gradual stabilization of the network, the differences of the network performance indexes of both schemes are reduced gradually. This is because the original RPL protocol also gradually formed a more optimal and more stable topology. In conclusion, F-RPL can form the DODAG topology faster and better at the expense of increasing a small amount of energy, and the successful

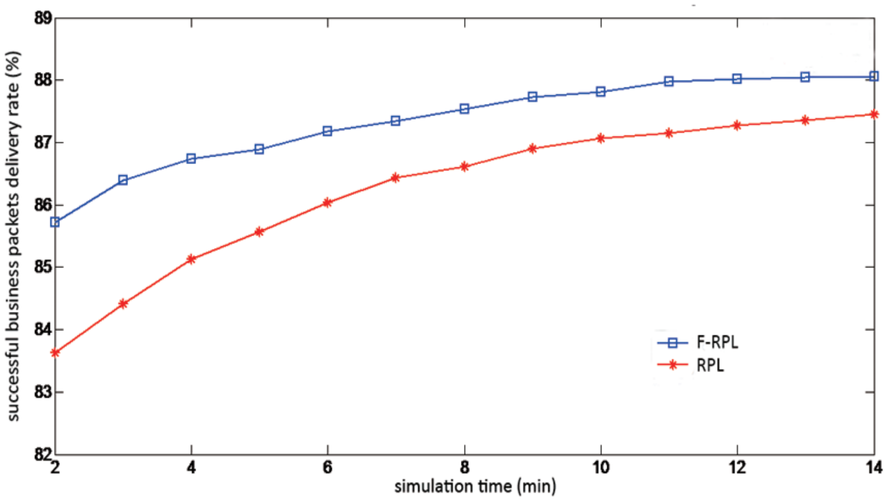

FIGURE 8. Average delivery rate of business packets under different simulation time

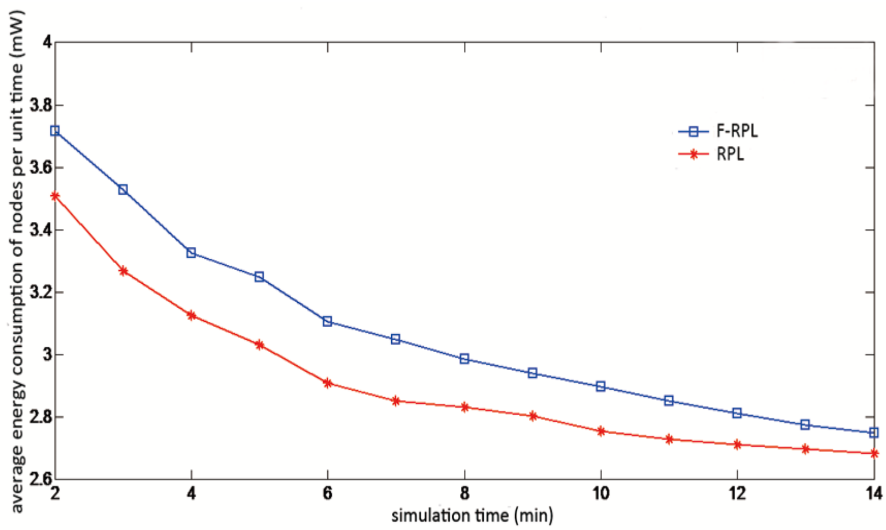

FIGURE 9. Average energy consumption of nodes per unit time under different simulation time

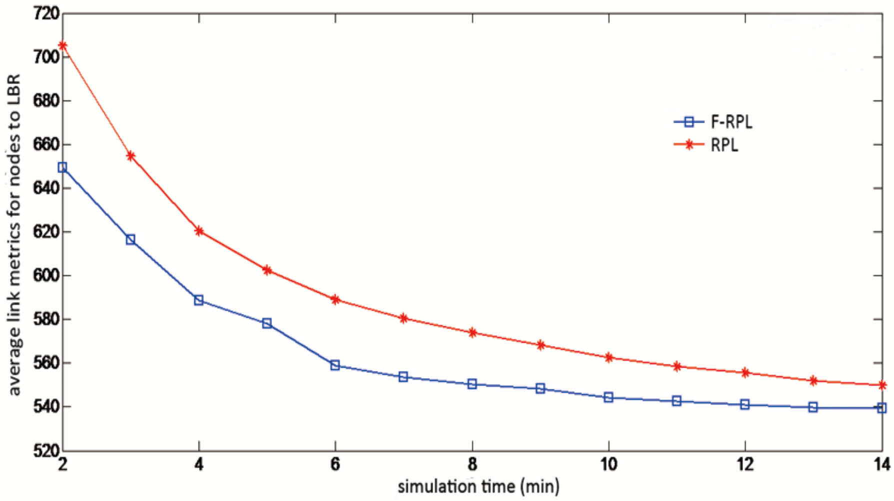

FIGURE 10. Average link metrics for nodes to LBR under different simulation time

business packets delivery rate, average link quality of the network and the time interval of forming a stable network topology are all improved when compared with RPL. It is adaptive for the wireless network with frequent network topology changes.

5. **Conclusions.** This paper proposes an optimized PRL routing protocol, F-RPL, by analyzing the working principle of RPL routing protocol and optimizing the Trickle algorithm of RPL timer management mechanism and the neighbor detection mechanism. F-RPL can control the broadcast of DIO packets more reasonable and fair and find the changes of neighbors faster, thus reducing the formation of suboptimal paths and optimizing the quality of DODAG network routing. We use Cooja simulation platform to compare and analyze the network performance of RPL and F-RPL. Simulation results show that F-RPL improves the DODAG network formation time and the quality of network routing under the premise of increasing a small amount of energy consumption so as to improve the successful business packets delivery rate and reduce the network packets loss.

Acknowledgment. This work was supported by the National Natural Science Foundation of China (Nos. 61170276, 61373135).

REFERENCES

[1] T. Winter, P. Thubert, A. Brandt et al., RPL: IPv6 routing protocol for low power and lossy network, *IETF Internet Draft Draftietf-Roll-Rpl-19*, pp.122-140, 2012.

[2] D. Ha and O. Gnawali, Performance of RPL under wireless interference, *IEEE Communication Magazine*, vol.51, no.12, pp.137-143, 2013.

[3] O. Gaddour, A. Koubaa, R. Rangarajan et al., Co-RPL routing for mobile low power wireless sensor networks using corona mechaniam, *Proc. of the 9th IEEE International Symposium on Industrial Embedded Systems*, pp.200-209, 2014.

[4] T. Clausen and U. Herberg, Comparative study of RPL-enabled optimized broadcast in wireless sensor networks, *Proc. of the 6th International Conference on Intelligent Sensors, Sensor Networks and Information Processing*, pp.7-12, 2010.

[5] B. Pavković, F. Theoleyre and A. Duda, Multipath opportunistic RPL routing over IEEE 802.15.4, *Proc. of the 14th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, pp.179-186, 2011.

[6] P. Kulkarni, S. Gormus, Z. Fan and B. Motz, A self-organising mesh networking solution based on enhanced RPL for smart metering communications, *IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks*, pp.1-6, 2011.

[7] C. Vallati and E. Mingozzi, Trickle-F: Fair broadcast suppression to improve energy-efficient routing formation with the RPL routing protocol, *Sustainable Internet and ICT for Sustainability*, pp.1-9, 2013.

[8] T. Coladon, M. Vučinić and B. Tourancheau, Multiple redundancy constants with trickle, *Proc. of the 26th IEEE Annual International Symposium on Personal, Indoor, and Mobile Radio Communications*, pp.1951-1956, 2015.

[9] Y. W. Lin and P.-H. Wang, Performance study of an adaptive trickle scheme for wireless sensor networks, *Lecture Notes in Electrical Engineering*, vol.331, pp.163-173, 2015.

[10] T. Narten, E. Nordmark, W. Simpson and H. Soliman, Neighbor discovery for IP version 6 (IPv6), *IETF RFC4861*, 2007.