Hindawi Publishing Corporation
Journal of Sensors
Volume 2016, Article ID 8128651, 11 pages
http://dx.doi.org/10.1155/2016/8128651

Research Article

Toward Improved RPL: A Congestion Avoidance Multipath Routing Protocol with Time Factor for Wireless Sensor Networks

Weisheng Tang,[1,2] Xiaoyuan Ma,[1] Jun Huang,[1] and Jianming Wei[1]

[1] *Shanghai Advanced Research Institute, Chinese Academy of Sciences, Shanghai 201210, China*
[2] *University of Chinese Academy of Sciences, Beijing 100049, China*

Correspondence should be addressed to Jianming Wei; wjm@sari.ac.cn

Received 15 January 2015; Revised 16 June 2015; Accepted 28 June 2015

Academic Editor: Giuseppe Ferri

Designing routing protocols in Low power and Lossy Networks (LLNs) imposes great challenges. In emergency scenarios, the large and rapid data traffic caused by emergencies will lead to network congestion and bring about significant packet loss and delay. Routing protocol for LLNs (RPL) is the IETF standard for IPv6 routing in LLNs. The basic version of RPL uses Expected Transmission Count (ETX) as the default routing metric; it cannot solve the problem of sudden large data traffic. In this paper, we propose a congestion avoidance multipath routing protocol which uses composite routing metrics based on RPL, named CA-RPL. A routing metric for RPL that minimized the average delay towards the DAG root is proposed, and the weight of each path is computed by four metrics. The mechanism is explained and its performance is evaluated through simulation experiments based on Contiki. Simulation results show that the proposed CA-RPL reduces the average time delay by about 30% compared to original RPL when the interpacket interval is short and has almost 20% reduction in packet loss ratio. The CA-RPL can effectively alleviate the network congestion in the network with poor link quality and large data traffic and significantly improve the performance of LLNs.

1. Introduction

In recent years, various kinds of natural disasters and hazardous substances accidents have occurred repeatedly, causing great losses. The deployment of wireless sensor networks (WSNs) in the hazardous area is helpful in real-time monitoring of environmental indicators and improving the efficiency and quality of emergency rescue [1, 2].

In most emergencies, wireless sensor networks are deployed in harsh areas, and the link quality between sensor nodes is poor. Different from ordinary environment monitoring system, the monitoring system under emergency scenarios needs to not only periodically collect the information gathered by sensor nodes, but also more importantly make rapid responses to some emergencies. In emergency scenarios, emergencies can cause large amounts of data traffic; this can lead to network congestion and make the monitoring system unable to detect information timely or lose some important information. Therefore, in the face of increased data traffic and accelerated interpacket intervals, the WSN needs to be adaptive dynamic traffic, has a good ability to resist congestion in the case of unstable links, and has high reliability and low delay.

The network mentioned above can be summarized as Low power and Lossy Network (LLN). Designing routing protocols in LLNs imposes great challenges [3, 4]. The main reason is that a LLN consists of a number of constrained nodes, with limited processing capacity, battery power, and memory. Moreover, the lossy links between devices are characterized by low data rate, high loss rate, and instability. Existing routing protocols such as Ad Hoc On Demand Vector (AODV), Open Shortest Path First (OSPF), Intermediate System to Intermediate System (IS-IS), and Optimized Link State Routing (OLSR) have been extensively evaluated by the Internet Engineering Task Force (IETF) Routing Over Low power and Lossy networks Working Group (ROLL WG) [5] and have been found to not satisfy the requirements of LLNs. Therefore, in the light of the characteristics of LLNs, the

ROLL WG has designed and specified the routing protocol for LLNs (RPL) in [6].

To cover the diverse requirements imposed by different applications, ROLL has specified in RFC 6551 [7] a set of routing metrics and constrains suitable for LLNs to be used in RPL. Whereas this document only provides examples, it does not provide details on the quantification of each routing metric. The RPL routes are built according to an Objective Function (OF) and a set of metrics and constraints. So far, the ROLL working group has specified two kinds of Objective Function, one is the Objective Function Zero (OF0) [8], where the hop count is the only routing metric adopted, and the other is Minimum Rank with Hysteresis Objective Function (MRHOF) [9], where Expected Transmission Count (ETX) is the routing metric adopted. However, these two OFs both take only a single metric as routing decision metric and perform poorly in LLNs, especially in emergency scenarios. If the RPL only considers the hop count, nodes will suffer from uneven energy and the rise of the packet loss ratio. If it only considers the ETX, the nodes will suffer from the rise of time delay.

In this paper, we propose a congestion avoidance multipath routing protocol based on RPL, named CA-RPL. The purpose is to make the monitoring information of sudden events in emergency scenarios be quickly and reliably transmitted to the sink node. We design a routing metric for RPL that minimized the average delay towards the DAG root, named DELAY_ROOT. The CA-RPL combines DELAY_ROOT and other three routing metrics as a composite routing metric, to calculate the weight of paths in the multipath routing. The CA-RPL is suitable for LLNs in emergency scenarios, and it can effectively alleviate network congestion by dispersing a large amount of data traffic to different paths and reduce the packet delay. Using CA-RPL, the LLNs can balance on reliability and latency, alleviate the network congestion, and have an improved performance.

The rest of the paper is organized as follows. Section 2 briefly summarizes the framework of RPL and surveys work related to our proposal. Section 3 designs the DELAY_ROOT routing metric and describes the composite routing metric and the multipath mechanism. Section 4 evaluates the impact of the proposed CA-RPL in terms of network performances with respect to the basic version of RPL protocol. Finally, Section 5 concludes the paper.

2. Related Work

RPL is a distance vector IPv6 routing protocol designed for LLNs. RPL organizes the network as a directed acyclic graph (DAG) that is partitioned into one or more destination oriented DAGs (DODAG) according to the number of DODAG roots. Routes are built according to an Objective Function (OF) and a set of metrics and constrains. The OF identifies routing performance objectives as well as the specific routing metrics to use in determining link costs. Each node maintains its rank towards the DAG root, which describes the relative location of the node in the DADAG. The exact way the rank is computed depends on the OF.

RPL specifies a set of new ICMPv6 (Internet Control Management Protocol Version 6) control messages [10] to construct and maintain a DODAG, and the three main messages are the DODAG Information Object (DIO), the DODAG Information Solicitation (DIS), and the Destination Advertisement Object (DAO). The DODAG root broadcasts DIO messages to construct a DODAG. The nodes that are closest to the root will first hear this message and decide whether to join this DODAG. Once they decide to join, each node computes its own rank according to Objective Code Point (OCP) field in the DIO which identifies OF. After the process, the nodes transmit the DIO messages with their own rank values and the DODAG identifier to their neighbors, until a leaf message receives it. If a node has already joined the DODAG, it can discard the DIO message or process the DIO message to modify its rank. Therefore, each node has an upward route through its parents to the DODAG root. In order to build download routes, destination information of a child node is embedded in a DAO message. The message is then sent to the parent of the node as a unicast packet, and the same procedure continues until the DODAG root receives the DAO message. A node can also ask DIO messages from its neighbors in case of having no parent or detecting loop packets.

There have been a lot of researches on the performance of RPL. The work in [11] introduces a study of RPL under the data collection model, compared with the Collection Tree Protocol (CTP), and the results show the advantages of RPL in terms of improving Packet Reception Ratio (PRR) and keeping low energy consumption. In [12], the authors compare the performance of RPL and LOADng protocol (Lightweight On-demand Ad hoc Distance-vector Routing Protocol-Next Generation) and indicate that RPL provides less control overhead and shorter delays and requires less memory than LOADng. Reference [13] indicates that DAO messages are responsible for the dominant component of the RPL signaling overhead. Other researches on performance of RPL are, for example, [14, 15]. These studies summarized the performance and advantages of RPL and pointed out some of its shortcomings.

Aimed at the deficiency of RPL, there have been extensive studies on optimizing RPL in LLNs. In [16], the authors provided ways to quantify the routing metrics so that they can be combined in an additive or lexical manner. Reference [17] proposes a routing metric which minimizes the delay towards the DAG root, but it has not been tested on other performance indicators. In [18], the authors proposed an energy-oriented routing mechanism to improve RPL routing protocol by combining the ETX and remaining energy metrics. Other studies on optimizing RPL are, for example, [19–21]. However, these optimizations are not very suitable for LLNs with sudden large data traffic.

3. CA-RPL Routing Protocol

A LLN consists of a number of constrained nodes, and the lossy links between devices are characterized by low data rate, high loss rate, and instability. RPL is a single path routing protocol where a node transmits packets to the preferred parent. Once the existing path fails, the network needs to select links to build a new topology, which will cause unnecessary cost. At the same time, RPL takes only a single metric as routing decision metric, and the large and rapid data traffic caused by emergencies will lead to network congestion and bring about significant packet loss and latency. Therefore, we propose a congestion avoidance multipath routing protocol which uses composite routing metrics based on RPL, named CA-RPL. We design a routing metric for RPL that minimized the average delay towards the DAG root, and the weight of every path is computed by the new metric and three other metrics, including ETX, rank, and the number of received packets. The CA-RPL considers the link reliability, load balance, and time factor, and it makes the LLN be balanced in terms of ease of congestion and reduction of the time delay.

3.1. Design of Minimized Delay Metric. The basic version of the RPL uses Expected Transmission Count (ETX) as the default routing metric, and it does not consider time factor. In order to achieve less time delay, we propose a minimized delay metric based on the ContikiMAC radio duty cycling protocol. Under this mechanism, a node learns the wake-up phase of its parents and sends packets to the first awaken parent. Therefore, it can reduce the time to wait until the receiver wakes up and finally minimize the time delay between DODAG nodes and DODAG root by reducing the time delay of each jump.

The wireless transceiver is usually the component with the highest power consumption in a low-power wireless device; it consumes as much power when passively listening for transmissions from other devices as it does when actively transmitting. Therefore, a duty cycling mechanism to turn the transceiver off periodically is needed to save power [22–24].

ContikiMAC is an asynchronous, sender-initiated radio duty cycling protocol that has a significantly more power-efficient wake-up mechanism than other previous duty cycling mechanisms [25]. ContikiMAC uses a set of timing constraints to achieve precise timing, a fast sleep optimization to allow receivers to quickly detect false-positive wake-up, and a transmission phase-lock optimization to allow run-time optimization of the energy-efficiency of transmissions.

ContikiMAC uses periodical wake-up to listen for packet transmissions from neighbors. When transmitting a data packet, the sender node repeatedly sends its packet until it receives a link layer acknowledgment packet from the receiver. Once a packet transmission is detected during a wake-up time, the receiver remains open to receive the packet and sends a link layer acknowledgment packet after the packet is successfully received. On the other hand, when sending a broadcast, the sender repeatedly sends the packet during the full wake-up interval to ensure that all neighbors have received it; in this case, the receiver does not need to reply to an ACK (acknowledgment) packet. The mechanism of ContikiMAC is depicted in Figures 1 and 2 (redrawn according to [25]).

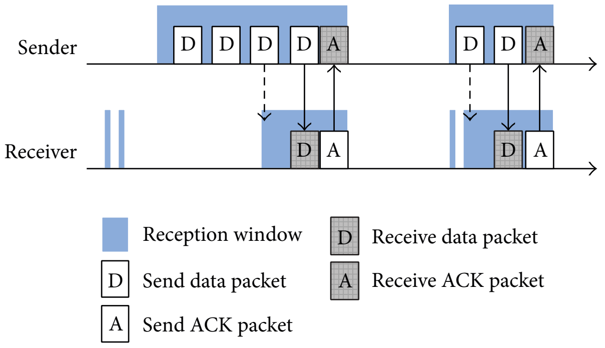


FIGURE 1: ContikiMAC mechanism of sending a data packet.

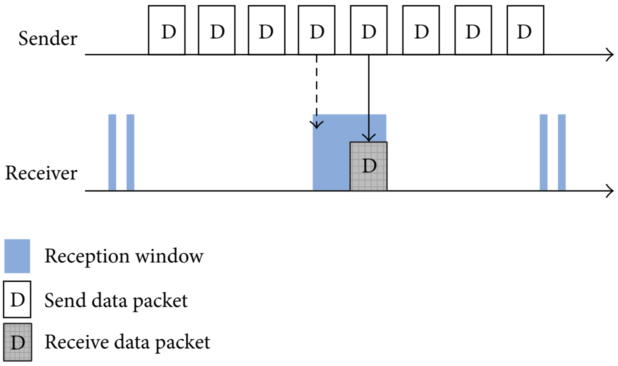


FIGURE 2: ContikiMAC mechanism of broadcast.

We use ContikiMAC as the radio duty cycling protocol in the Contiki operating system [26]. In the current implementation, ContikiMAC assumes that all nodes have the same wake-up interval, expressed as cycle time (C_T). We assume that node B forwards a packet from node A to node C. The forwarding delay (FD) equals the time interval between the moment node B receives the packet from node A and the moment node C finally receives the packet.

In [17], the authors proposed that the forwarding delay consists of seven parts; the only part we can minimize is the waiting time till the receiver's wake-up. Therefore, a lower bound for FD is given by the sum of the other time intervals, and we define it as Minimum Forwarding Time (MFT).

When computing the forwarding delay from B to C, we can assume that B and C have the same cycle time (i.e., $C_{T,B} = C_{T,C}$), and the difference between their wake-up phases is constant. Therefore, FD is given by

$$\mathrm{FD} = \begin{cases} \varphi_C - \varphi_B, & \varphi_C - \varphi_B \geq \mathrm{MFT}, \\ \varphi_C - \varphi_B + C_{T,C}, & \varphi_C - \varphi_B < \mathrm{MFT}, \end{cases} \quad (1)$$

where φ_B is B's wake-up phase and φ_C is C's wake-up phase.

The second equation expresses that if the wake-up times of nodes B and C are too close to each other, node B has to wait for the next C's wake-up time to send the packet.

In future implementation, ContikiMAC may support nodes to have different cycle time (namely, $C_{T,B} \neq C_{T,C}$), and the difference $\varphi_C - \varphi_B$ is not constant. In this case, only the average forwarding delay can be calculated, which can be expressed as

$$\mathrm{FD} = \frac{[(C_{T,C} + \mathrm{MFT}) + \mathrm{MFT}]}{2} = \left(\frac{C_{T,C}}{2}\right) + \mathrm{MFT}. \quad (2)$$

The minimized delay metric is defined as the cumulative sum of the hop-by-hop forwarding delays along the path from a node towards the DAG root. It can be expressed as

$$\mathrm{DELAY\_ROOT} = \begin{cases} 0, & \text{root}, \\ \mathrm{DELAY\_ROOT}_p + \mathrm{FD}_p, & \text{others}, \end{cases} \quad (3)$$

where $\mathrm{DELAY\_ROOT}_p$ is the average delay to the DAG root announced by a candidate parent p; FD_p is the forwarding delay between the node and its candidate parent p. In order to compute FD_p, the node uses (1), if it has the same cycle time of its parent, or (2), if it has a different cycle time from its parent.

3.2. Design of Multipath CA-RPL. CA-RPL is a dynamic weight multipath routing protocol based on RPL, to deal with dynamic changes of topology and traffic. Its main idea is based on several routing metrics, dynamically adjusting the path selection scheme and balancing the network load according to the different link weights.

The link weight W of CA-RPL is given by

$$\begin{aligned} W\left(e_{uv}\left(n_u, n_v\right)\right) &= \frac{\gamma}{a \cdot \mathrm{ETX}\left(e_{uv}\left(n_u, n_v\right)\right) + b \cdot \mathrm{REC}_v + c \cdot \mathrm{RANK}_v + d \cdot \mathrm{DELAY\_ROOT}\left(u, v\right)}, \\ \sum_{t \subset L} W\left(e_{uv}\left(n_u, n_t\right)\right) &= M, \end{aligned} \quad (4)$$

where $e_{uv}(n_u, n_v)$ is the link between node u and its parent v and $\mathrm{ETX}(e_{uv}(n_u, n_v))$ is the number of retransmissions required to successfully send a data packet in the link $e_{uv}(n_u, n_v)$; it is a measure of the quality of the channel. REC_v is the number of packets node v received in a period of time; if the value is small, it means that the link between u and v is relatively free, which can be used to transmit data packets in order to balance the network load. The rank value of node v RANK_v is used to reduce the occurrence of loops and make it a priority to send packets to the nodes closer to the DAG root. $\mathrm{DELAY\_ROOT}(u, v)$ is the minimized delay metric of node u if it transmits data packets to its parent v. The parameters a, b, c, and d are the corresponding weight coefficient of the routing metrics. γ is an adjustment factor. We need to decide the five parameters according to different application requirements, adjust the weights allocation of load balancing, and then select the best corresponding value through tests. In the simulations in Section 4 below, we set the parameters $a = 10$, $b = 10$, $c = 2$, and $d = 10$, where $\gamma = 100000$. This is designed to meet the needs of alleviating the network congestion. M is the total number of the data packets that node u needs to transmit. L is the set of node u's parents.

This algorithm is to find a set of path P, which makes

$$\begin{aligned} W_1\left(P\right) &= \max W\left(e_{uv}\left(n_u, n_v\right)\right), \\ W_2\left(P\right) &= \max\left\{W\left(e_{uv}\left(n_u, n_v\right)\right) - W_1\left(P\right)\right\}. \end{aligned} \quad (5)$$

Find two paths in turn in the path set P to maximize $W(P)$, as the optimal path of CA-RPL.

Node u calculates the weight of every parent node according to (4), and the number of packets assigned to the path through node v is given by

$$N\left(e_{uv}\left(n_u, n_v\right)\right) = M \cdot \frac{W\left(e_{uv}\left(n_u, n_v\right)\right)}{\sum_{t \subset L} W\left(e_{uv}\left(n_u, n_t\right)\right)}. \quad (6)$$

The CA-RPL can adaptively adjust the routes according to the dynamic traffic of each node. As $W(e_{uv}(n_u, n_v))$ is inversely proportional to ETX, REC_v, RANK_v, and $\mathrm{DELAY\_ROOT}(u, v)$, the LLN can balance on reliability and latency, alleviate the network congestion, and have an improved performance.

3.3. The Procedure for Building and Maintaining CA-RPL Routes. In the process of network building, each node records the number of packets it receives in a period of time. A node periodical broadcasts its DIO message, where ETX, REC, RANK value, cycle time C_T, wake-up phase, its DELAY_ROOT value, and other fields are included. Once a child node u receives the DIO message from node v, it first extracts the ETX, REC_v, and RANK_v and then calculates the minimized delay metric $\mathrm{DELAY\_ROOT}(u, v)$. The forwarding delay FD_v is calculated using (1), if it has the same cycle time of its parent v, or (2), if its cycle time is different from that of node v. The minimized delay metric $\mathrm{DELAY\_ROOT}(u, v)$ is equal to the forwarding delay FD_v plus the candidate parent node v reserved $\mathrm{DELAY\_ROOT}_v$ in DIO. Then node u computes the weight of path through parent node v to the root. After that, the child node u contrasts DIO messages received from different parents, computes the weight of each path, ranks the parent nodes according to their weights and forms a link table, and selects the best two candidate parent nodes. Finally, the child node u sends its DAO to the first parent node in the link table and builds the route to DAG root. At the same time, the child node puts the latest DELAY_ROOT parameter in its own DIO and broadcasts to its neighbors so that they can update the weight of links. The flowchart is shown in Figure 3.

Later in the process of transmitting, a node will send data packets to different parent nodes according to their weight. At the same time, because the parent node sends DIO messages

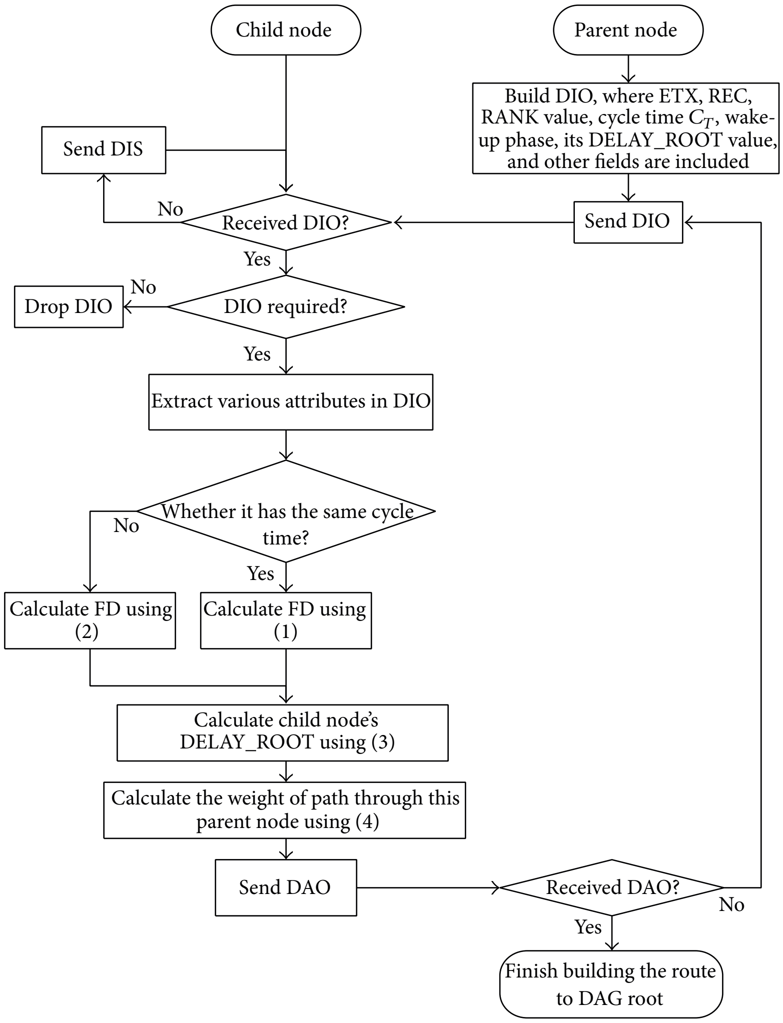

Child node
Parent node
Build DIO, where ETX, REC, RANK value, cycle time C_T, wake-up phase, its DELAY_ROOT value, and other fields are included
Send DIS
No
Received DIO?
Send DIO
Yes
No
Drop DIO
DIO required?
Yes
Extract various attributes in DIO
No
Whether it has the same cycle time?
Yes
Calculate FD using (2)
Calculate FD using (1)
Calculate child node's DELAY_ROOT using (3)
Calculate the weight of path through this parent node using (4)
Send DAO
Received DAO?
No
Yes
Finish building the route to DAG root

Figure 3: The procedure for building CA-RPL routes.

regularly, the child node calculates the link weight when it receives a new DIO message and updates its link table to ensure that the link table is up to date.

4. Performance Evaluation

4.1. Application Scenario and Simulation Environment. The CA-RPL routing protocol is designed for wireless sensor networks in emergency scenarios. Emergency scenarios include the accident scene and the potential accident scene. For instance, a typical application is the monitoring and warning system in a chemical industrial park, for monitoring the leakage of methane at the scene of the dangerous chemicals leakage accident. We may deploy a WSN consisting of 20 or more Tmote Sky [27] devices in storage areas of dangerous chemicals. A Tmote Sky device is composed of a MSP430F1611 microcontroller and a CC2420 radio with a data rate of 250 kbps using IEEE 802.15.4 MAC and physical layer specifications. The Tmote Sky device connects with a methane sensor through its expansion connector. For most of the time, the sensor nodes are in sleep mode. However, when methane leakage is detected, the nodes will produce large amounts of data and transmit it to the sink node. Figures 4 and 5 show the application scenarios.

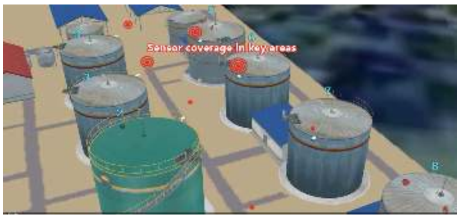


Figure 4: Application scenario.

According to the project experience of related cases based on the products of Helicomm Inc., the typical actual application system in emergency scenario needs to meet the following criteria: a node sends data packets at least once in every 10 s; the maximum hop count to the sink node is not more than 12; the data bandwidth of single channel reaches 250 kbps; and the response time of the system is not more

Figure 5: The sketch of methane leakage accident.

than 5 s. The effectiveness of the system will be low if it exceeds this standard.

In order to verify the proposal CA-RPL is a practicable routing protocol and suitable for LLNs with sudden large data traffic in emergency scenarios, the CA-RPL and the original RPL are simulated and analyzed. With the use of ContikiMAC duty cycling protocol which has a significantly more power-efficient wake-up mechanism, both original RPL and CA-RPL have good performance in energy consumption. Furthermore, although energy conservation is one of the most important issues in WSN, a WSN must make rapid response to emergencies whatever the energy cost. For instance, when the methane leakage is detected, it is no longer necessary for a monitoring and warning system to conserve energy. Therefore, in this paper we mainly focus on the packet reception number (PRN) of the root node per unit time, throughput, packet loss ratio, and latency. We implement our proposal CA-RPL in Contiki 2.6 operating system. Contiki is an open source, highly portable multitasking operating system, suitable for embedded systems and wireless sensor networks [26]. In order to simulate the Low power and Lossy Network, we have used Cooja, a flexible Java-based simulator designed for WSNs running Contiki OS. Cooja simulates networks of sensor nodes where each node can be of different type in terms of software and hardware [28].

The simulated platform is Tmote Sky, and we use the Cooja simulator along with the MSPsim [29] emulator, which provide accurate emulation in both cycle-level of the MSP430 microcontroller and bit-level of the CC2420 radio transceiver. The simulated WSN includes 21 nodes with an equal horizontal and vertical separation of 100 m; node 1 is a sink node located at the edge of the network. The nodes are arranged as depicted in Figure 6, the radio communication range is 40 m, and the interference range is 80 m. Nodes are assumed to be running Contiki OS, ContikiMAC as a MAC layer duty cycling protocol, RPL or CA-RPL as a routing protocol, and Unit Disk Graph Medium (UDGM) as the radio communication model. Table 1 describes the summary of the simulation environment.

We evaluate the proposal routing protocol from three aspects. Firstly, we count the packet reception number of root nodes per unit time and evaluate the throughput of CA-RPL. Secondly, we compute the packet loss rate of the network, which represents the link quality and network congestion situation. Thirdly, we compute the average delay of nodes to transmit data packets to the sink node. Each simulation run lasts for 6 minutes. All results presented in the figures below are averaged over 10 simulation runs, and error bars show the 95% confidence intervals.

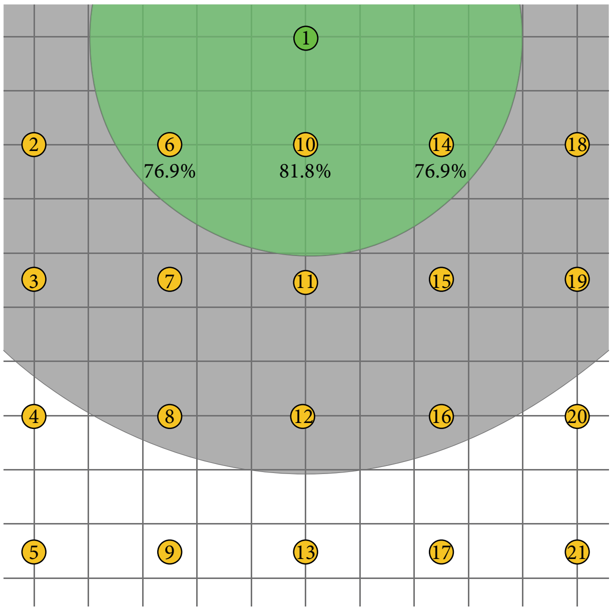


Figure 6: Simulation environment.

Table 1: The simulation environment summary.

Simulation environment	
Operating system	Contiki 2.6
Simulator	Cooja
Radio model	Unit Disk Graph Medium (UDGM)
Motes	Tmote Sky
Deployment area	100 m ∗ 100 m
Transmission range	40 m
Communication protocols	
Network	Routing (RPL, CA-RPL)
Adaption	6LowPAN
Data-link	CSMA Radio duty cycling (ContikiMAC)
Physical	IEEE 802.15.4

4.2. Packet Reception Number (PRN) and Throughput. Set the interpacket interval to 4 s; test the packet reception number (PRN) of the root node per unit time at different transmit success ratio (TX) and receive success ratio (RX). Here we only refer to data packets. The results are represented in Figure 7. The transmit success ratio and receive success ratio reflect the link quality of the network. The PRN of root node increases with transmit success ratio and receive success ratio. When TX/RX ratio is low, the link quality is poor and the retransmission due to failure to send or receive packets causes a large amount of data traffic in the network. This leads to network congestion and packet loss and decreases the PRN of root. As is shown in Figure 7, compared with original RPL which uses ETX as its routing metric, the proposal CA-RPL

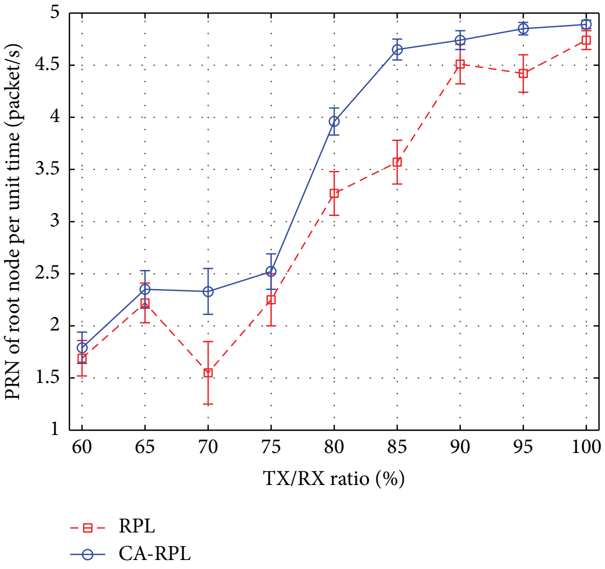


FIGURE 7: PRN in the case of different TX/RX ratio.

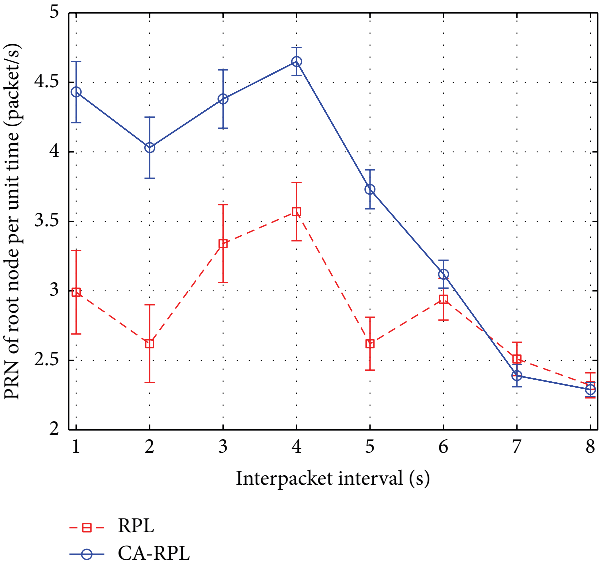


FIGURE 8: PRN in the case of different interpacket interval.

increases PRN parameters under the condition of different TX and RX ratio. It relieves the network congestion and makes information from each node transmit to the sink node timely.

Set both the transmit success ratio and receive success ratio to 85% and test the packet reception number of root nodes per unit time at different interpacket interval. The results are represented in Figure 8. When the interpacket interval is short (i.e., 1 s and 2 s), there is a large amount of data packets transmitting in the network, and the PRN of the sink node would have been increased. However, the excessive amount of data traffic leads to network congestion and a lot of conflicts and packet loss. The data packets cannot be transmitted to the sink node in time, and the PRN parameters are much smaller than expected. CA-RPL increases the PRN by nearly 50% compared to original RPL when the interpacket interval is less than 5 s.

In order to verify the applicability of CA-RPL, we tested the PRN in the case of different network size. As is represented in Figure 9, when the number of nodes is 10, the network is very small, and there is few conflicts in the network. With the growth of network size, more and more conflicts appear in the network and lead to network congestion. It is difficult to transmit data packets to the sink node. CA-RPL increases the PRN by over 20% compared to original RPL in most cases of network sizes.

To verify whether node mobility affects the performance of proposed routing protocol, we test the PRN in case nodes can move. In the simulation environment, nodes move by random waypoint model, the minimum speed of node is 1 m/s, the maximum speed is 5 m/s, and the moving range is 100 m ∗ 100 m, as is shown in Figure 6. The results are represented in Figure 10, and the mobile nodes make the PRN of the network significantly decrease. However, the performance of CA-RPL is still better than original RPL.

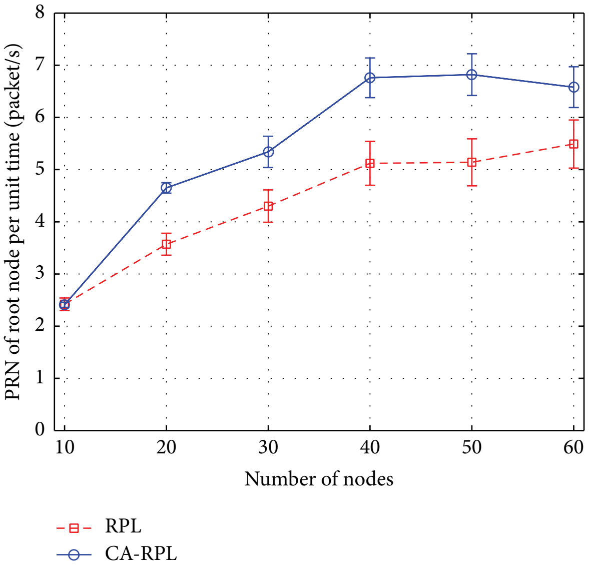


FIGURE 9: PRN in the case of different network size.

Keep the simulation conditions the same as above, and test the network throughput in 6 minutes, which is computed as the amount of bytes received by the sink node. These bytes come from both data packets and control packets such as DAO and DIO. We calculated the average throughput in ten simulation runs. The results are shown in Table 2. As we can observe in Table 2, when the interpacket interval is 1 s, the throughput of the network using CA-RPL is 34% greater than the network with original RPL.

Calculate the average network throughput during 6 minutes, and the network throughput per second is shown in Figures 11 and 12. We can see from Figure 11 that, with the reducing of interpacket interval, there are more and more data packets transmitting in the network, leading to network congestion and making network throughput decrease when

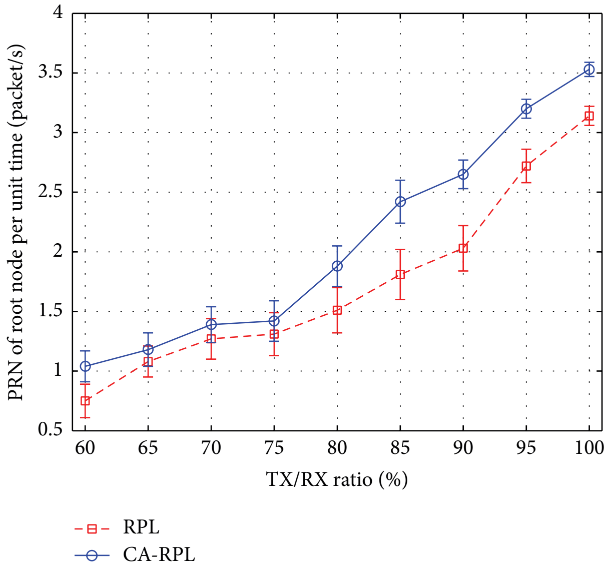


Figure 10: PRN in the case of mobile nodes.

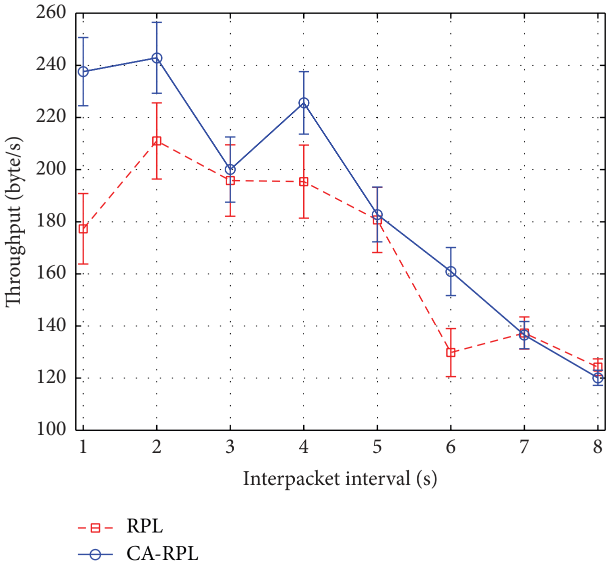


Figure 11: Throughput in the case of different interpacket interval.

Table 2: Network throughput in 6 minutes.

Interpacket interval	Throughput (bytes)	
	Original RPL	CA-RPL
1 s	63836	85545
2 s	75975	87444
3 s	70479	71999
4 s	70338	81230
5 s	65039	65820
6 s	46734	57920
7 s	49412	49143
8 s	44695	43219

interpacket interval is 1 s. Figure 12 shows the throughput in the case of different TX/RX ratio. The TX/RX ratio indicates the link quality, and the result shows that the network with higher link quality has a greater throughput. CA-RPL relieves the network congestion and increases the average network throughput.

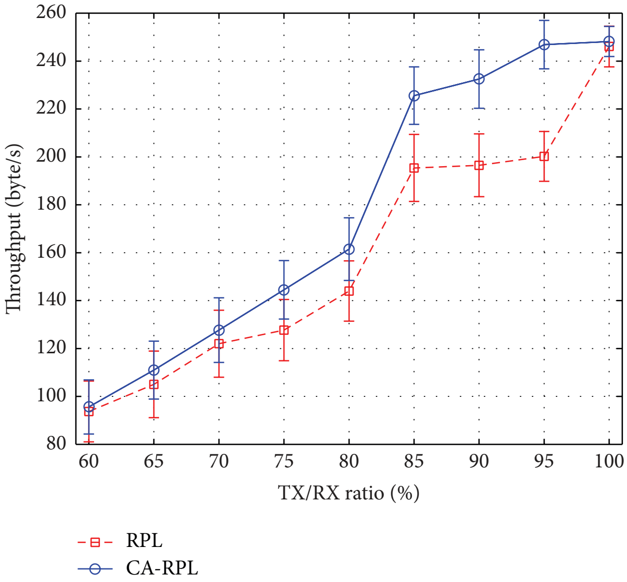


Figure 12: Throughput in the case of different TX/RX ratio.

4.3. Packet Loss Rate. The packet loss rate has been tested under the condition of different TX/RX ratio and interpacket interval.

As is represented in Figure 13, with the increase of TX/RX ratio, the packet loss rate of the network overall trend decreases. The CA-RPL has at least 10% less packet loss rate than original RPL when the link quality is not so good.

Figure 14 represents the packet loss rate of the network with different interpacket interval. When the interpacket interval is short, CA-RPL relieves the network congestion and reduces the packet loss rate. In particular, when the interpacket interval is 4 s or 5 s, CA-RPL reduces the packet loss ratio by 25% compared to original RPL. However, with the reducing of interpacket interval, the packet loss ratio of CA-RPL is also increased. This is because every path in the network is suffering from congestion at the moment. When the interpacket interval is long (i.e., 6 s, 7 s, and 8 s), the number of data packets over the network is less, not enough to cause the network congestion. At this point, both original RPL and CA-RPL have fairly low packet loss rate. The reason that CA-RPL have a higher packet loss rate than original RPL when there is no congestion in the network is because the routing algorithm of the original RPL is simpler than CA-RPL and operates quickly when the data traffic is small.

As is represented in Figure 15, with the growth of network size, more and more conflicts appear in the network and lead to network congestion and packet loss. The CA-RPL has almost 20% reduction in packet loss ratio in most cases of network sizes.

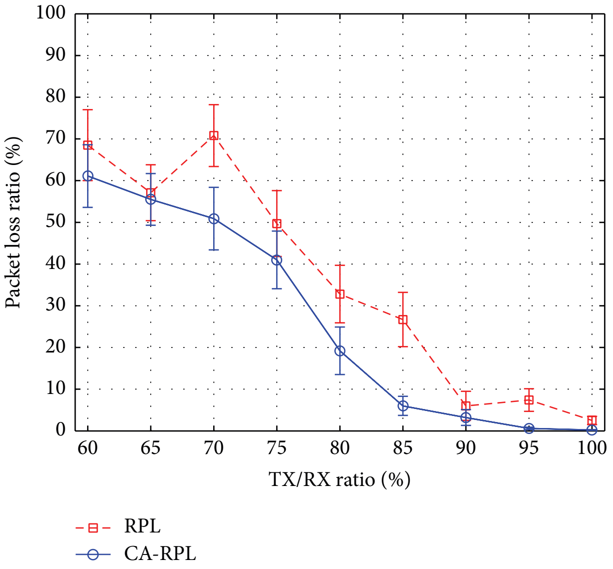


FIGURE 13: Packet loss ratio in the case of different TX/RX ratio.

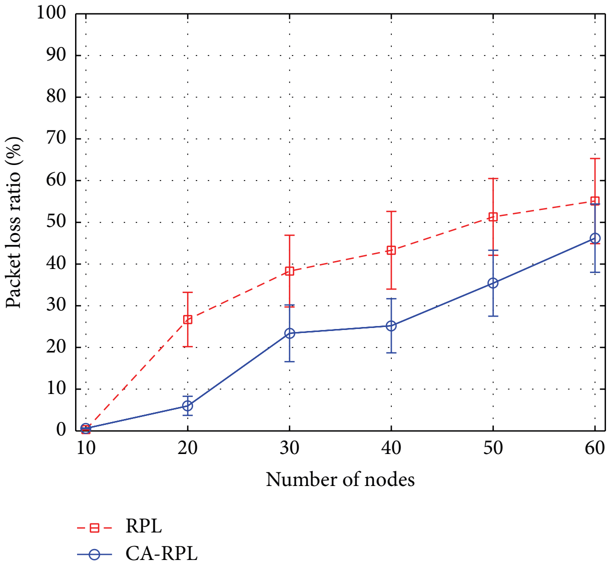


FIGURE 15: Packet loss ratio in the case of different network size.

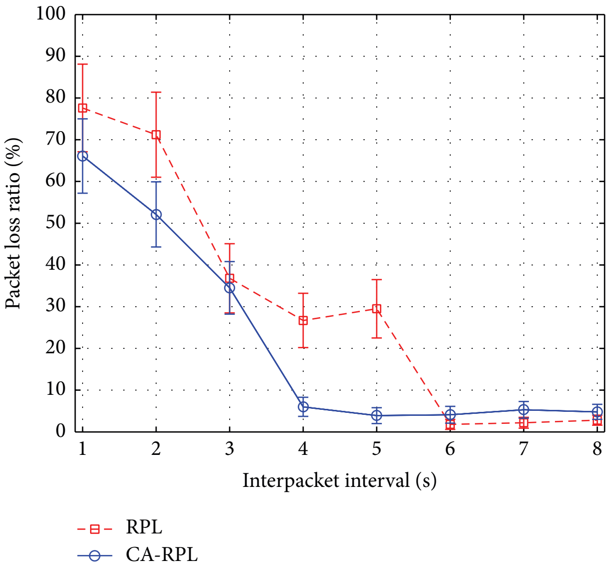


FIGURE 14: Packet loss ratio in the case of different interpacket interval.

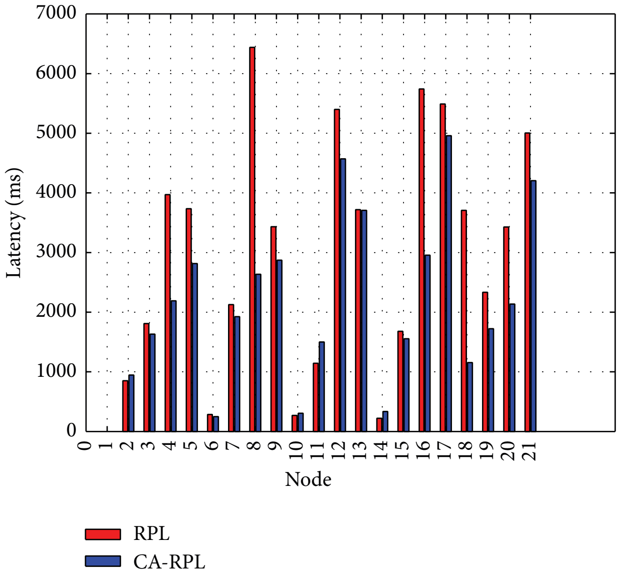


FIGURE 16: Average delay of each node to transmit data packets to the sink node.

4.4. Latency. Set both the transmit success ratio and receive success ratio to 85% and the interpacket interval to 4 s, and test the average delay of each node to transmit data packets to the sink node. The results are represented in Figure 16. As is shown in the figure, when using CA-RPL, the majority of nodes have less time delay than the nodes using original RPL. This is because the multipath mechanism and the composite routing metric of CA-RPL make the nodes able to avoid the congestion link when sending packets, and the consideration of the time factor makes the nodes choose the parent which can first awake and receive packets. This makes the network achieve balance between congestion avoidance and delay control. There are several exceptional nodes in the figure; they are nodes 2, 10, 11, and 14. The reason that these nodes have a higher time delay is because they are close to the sink node, there is only one path to the sink node, and the routes are the same for these nodes when using CA-RPL and original RPL. Because the optimization algorithm requires a series of calculating weights and optimal parent work, it is a little more complicated than the original RPL. Therefore, the time delay of these nodes slightly increased when using CA-RPL. However, the time delay is still maintained at low levels, the time delay of nodes far away from the sink node has been significantly reduced, and the CA-RPL reduces the overall latency of the network.

Next, the average time delay of nodes in the network sending packets to sink node has been tested under the condition of different TX/RX ratio and interpacket interval.

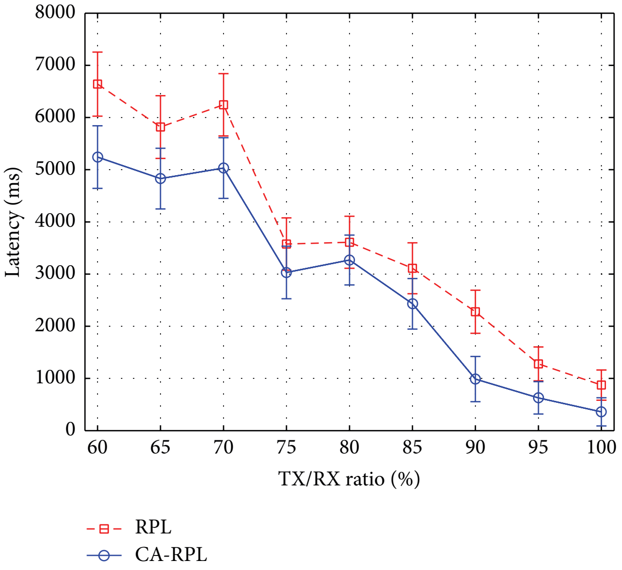


Figure 17: Latency in the case of different TX/RX ratio.

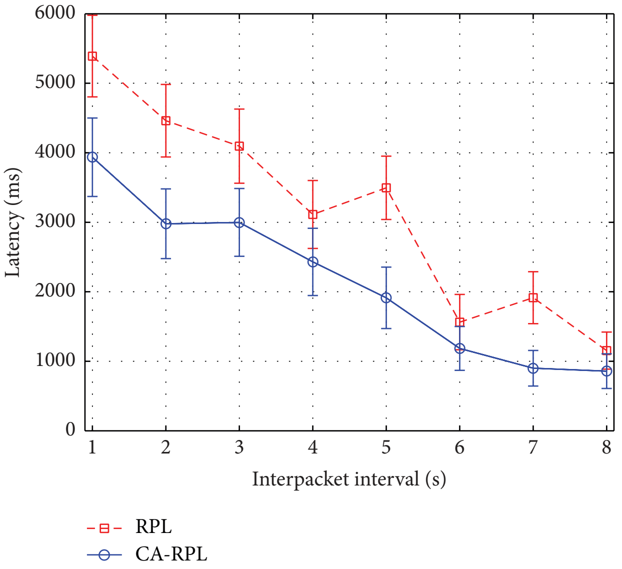


Figure 18: Latency in the case of different interpacket interval.

As is represented in Figure 17, with the decrease of TX/RX ratio, the link quality is worse and leads to network congestion and increased time delay. The CA-RPL decreases the average time delay of nodes in the network for at least 0.5 s.

As is represented in Figure 18, when the interpacket interval is short, there is a large amount of data packets transmitting in the network, which will lead to network congestion and packet loss. The time delay of original RPL increases rapidly. The CA-RPL reduces the average time delay by about 30% when the interpacket interval is short compared to original RPL and is suitable for the condition of sudden large data traffic.

We can see from the above three figures that the latency of WSN has been controlled in the range of 5 s by using CA-RPL as the routing protocol. However, when using the original RPL as the routing protocol, the network performance cannot meet the requirements of the standard we mentioned above. Once the interpacket interval is short or the link quality is poor, the response time of the network using original RPL will be too long for emergency applications, and the system effectiveness is not enough. The CA-RPL can alleviate network congestion by dispersing a large amount of data traffic to different paths and reduce the time delay by using a composite routing metric with time factor. The results show that CA-RPL is suitable for LLNs in emergency scenarios.

5. Conclusion and Future Work

In face of the problems of RPL in LLNs, we propose a congestion avoidance multipath routing protocol which uses composite routing metrics based on RPL and ContikiMAC, named CA-RPL. A routing metric for RPL that minimized the average delay towards the DAG root is proposed; the weight of every path is computed by the new metric and three other metrics. Simulation results show that the proposed CA-RPL provides lower packet loss ratio and time delay than original RPL. The CA-RPL can effectively alleviate the network congestion in the network with poor link quality and large data traffic and significantly improve the performance of LLNs.

In our future work, we will implement the CA-RPL in real Tmote Sky devices and test the network performance based on experiments in realistic usage scenarios. Experimental results will help us make further improvement. We will also try to investigate the proposed metrics under the routing algebra principles. Furthermore, we intend to measure network lifetimes since the devices are battery powered, and we will explore the optimization of LLNs with multisink and mobile nodes.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

The authors acknowledge the support provided by the Safety and Emergency Lab of Shanghai Advanced Research Institute (SARI) for this study. This research is supported by “the next generation of information technology (IT) for sensing China” of the Chinese Academy of Sciences (XDA06010800) and the National Natural Science Foundation of China (nos. 61302151 and 61201446).

References

[1] A. B. A. Dow, A. Sklorz, and W. Lang, “A microfluidic preconcentrator for enhanced monitoring of ethylene gas,” *Sensors and Actuators A: Physical*, vol. 167, no. 2, pp. 226–230, 2011.

[2] A. Somov, A. Baranov, D. Spirjakin, A. Spirjakin, V. Sleptsov, and R. Passerone, "Deployment and evaluation of a wireless sensor network for methane leak detection," *Sensors and Actuators A: Physical*, vol. 202, pp. 217–225, 2013.

[3] U. Herberg and T. Clausen, "Study of multipoint-to-point and broadcast traffic performance in the "IPv6 Routing Protocol for Low Power and Lossy Networks"," *Journal of Ambient Intelligence and Humanized Computing*, vol. 2, no. 4, pp. 293–305, 2011.

[4] J. G. Ko, A. Terzis, S. Dawson-Haggerty, D. Culler, J. Hui, and P. Levis, "Connecting low-power and lossy networks to the internet," *IEEE Communications Magazine*, vol. 49, no. 4, pp. 96–101, 2011.

[5] IETF-ROLL, Routing Over Low power and Lossy Networks (ROLL)—Working Group, 2014, http://datatracker.ietf.org/wg/roll/.

[6] T. Winter, P. Thubert, A. Brandt et al., "RPL: IPv6 Routing Protocol for Low power and Lossy Networks," RFC 6550, IETF ROLL WG, 2012, http://www.ietf.org/rfc/rfc6550.txt.

[7] J. Vasseur, M. Kim, K. Pister, N. Dejean, and D. Barthel, "Routing metrics used for path calculation in low-power and lossy networks," RFC 6551, Internet Engineering Task Force, 2012, http://www.ietf.org/rfc/rfc6551.txt.

[8] P. Thubert, "Objective Function Zero for the Routing Protocol for Low-Power and Lossy Networks (RPL)," RFC 6552, Internet Engineering Task Force, 2012, http://www.ietf.org/rfc/rfc6552.txt.

[9] O. Gnawali and P. Levis, "The Minimum Rank with Hysteresis Objective Function," RFC 6719, Internet Engineering Task Force, 2012, http://www.ietf.org/rfc/rfc6719.txt.

[10] A. Conta, S. Deering, and M. Gupta, "Internet control message protocol (ICMPv6) for the internet protocol version 6 (IPv6) specification," IETF RFC 4443, Internet Engineering Task Force, 2006, http://www.ietf.org/rfc/rfc4443.txt.

[11] N. T. Long, N. De Caro, W. Colitti, A. Touhafi, and K. Steenhaut, "Comparative performance study of RPL in wireless sensor networks," in *Proceedings of the 19th IEEE Symposium on Communications and Vehicular Technology in the Benelux (SCVT '12)*, pp. 1–6, Eindhoven, The Netherlands, November 2012.

[12] M. Vucinic, B. Tourancheau, and A. Duda, "Performance comparison of the RPL and LOADng routing protocols in a home automation scenario," in *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC '13)*, pp. 1974–1979, Shanghai, China, April 2013.

[13] N. Accettura, L. A. Grieco, G. Boggia, and P. Camarda, "Performance analysis of the RPL Routing Protocol," in *Proceedings of the IEEE International Conference on Mechatronics (ICM '11)*, pp. 767–772, Istanbul, Turkey, April 2011.

[14] U. Herberg and T. Clausen, "Study of multipoint-to-point and broadcast traffic performance in the 'IPv6 Routing Protocol for Low Power and Lossy Networks'," *Journal of Ambient Intelligence and Humanized Computing*, vol. 2, no. 4, pp. 293–305, 2011.

[15] T. Clausen, U. Herberg, and M. Philipp, "A critical evaluation of the IPv6 Routing Protocol for Low Power and Lossy Networks (RPL)," in *Proceedings of the IEEE 7th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob '11)*, pp. 365–372, October 2011.

[16] P. Karkazis, H. C. Leligou, L. Sarakis et al., "Design of primary and composite routing metrics for RPL-compliant Wireless Sensor Networks," in *Proceedings of the International Conference on Telecommunications and Multimedia (TEMU '12)*, pp. 13–18, Heraklion, Greece, August 2012.

[17] P. Gonizzi, R. Monica, and G. Ferrari, "Design and evaluation of a delay-efficient RPL routing metric," in *Proceedings of the 9th International Wireless Communications and Mobile Computing Conference (IWCMC '13)*, pp. 1573–1577, Sardinia, Italy, July 2013.

[18] L.-H. Chang, T.-H. Lee, S.-J. Chen, and C.-Y. Liao, "Energy-efficient oriented routing algorithm in wireless sensor networks," in *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics (SMC '13)*, pp. 3813–3818, IEEE, Manchester, UK, October 2013.

[19] B. Pavkovic, A. Duda, W.-J. Hwang, and F. Theoleyre, "Efficient topology construction for RPL over IEEE 802.15.4 in wireless sensor networks," *Ad Hoc Networks*, vol. 15, pp. 25–38, 2014.

[20] G. Oikonomou, I. Phillips, and T. Tryfonas, "IPv6 multicast forwarding in RPL-based wireless sensor networks," *Wireless Personal Communications*, vol. 73, no. 3, pp. 1089–1116, 2013.

[21] O. Iova, F. Theoleyre, and T. Noel, "Using multiparent routing in RPL to increase the stability and the lifetime of the network," *Ad Hoc Networks*, vol. 29, pp. 45–62, 2015.

[22] G. Anastasi, M. Conti, M. Di Francesco, and A. Passarella, "Energy conservation in wireless sensor networks: a survey," *Ad Hoc Networks*, vol. 7, no. 3, pp. 537–568, 2009.

[23] K. Han, J. Luo, Y. Liu, and A. Vasilakos, "Algorithm design for data communications in duty-cycled wireless sensor networks: a survey," *IEEE Communications Magazine*, vol. 51, no. 7, pp. 107–113, 2013.

[24] M. Doudou, D. Djenouri, and N. Badache, "Survey on latency issues of asynchronous MAC protocols in delay-sensitive wireless sensor networks," *IEEE Communications Surveys & Tutorials*, vol. 15, no. 2, pp. 528–550, 2013.

[25] A. Dunkels, "The ContikiMAC radio duty cycling protocol," Tech. Rep. T2011:13, Swedish Institute of Computer Science, 2011, http://www.sics.se/~adam/dunkels11contikimac.pdf.

[26] A. Dunkels, B. Grönvall, and T. Voigt, "Contiki—a lightweight and flexible operating system for tiny networked sensors," in *Proceedings of the 29th Annual IEEE International Conference on Local Computer Networks (LCN '04)*, pp. 455–462, November 2004.

[27] Moteiv, "Tmote Sky—ultra low power IEEE 802.15.4 compliant wireless sensor module," Datasheet, 2014, http://www.eecs.harvard.edu/~konrad/projects/shimmer/references/tmote-sky-datasheet.pdf.

[28] F. Österlind, A. Dunkels, J. Eriksson, N. Finne, and T. Voigt, "Cross-level sensor network simulation with Cooja," in *Proceedings of the 31st Annual IEEE Conference on Local Computer Networks (LCN '06)*, pp. 641–648, November 2006.

[29] J. Eriksson, A. Dunkels, N. Finne, F. Österlind, and T. Voigt, "Mspsim—an extensible simulator for msp430-equipped sensor boards," in *Proceedings of the European Conference on Wireless Sensor Networks (EWSN '07)*, Delft, The Netherlands, 2007.

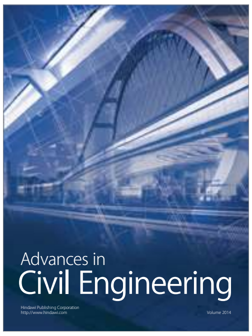
Journal of
Engineering

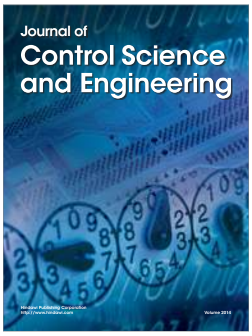
The Scientific
World Journal

International Journal of
Rotating
Machinery

Journal of
Sensors

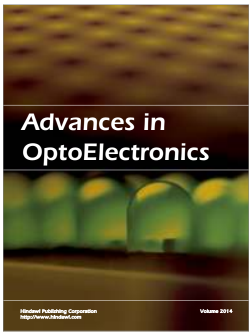
International Journal of
Distributed
Sensor Networks


Advances in
Civil Engineering


Journal of
Control Science
and Engineering

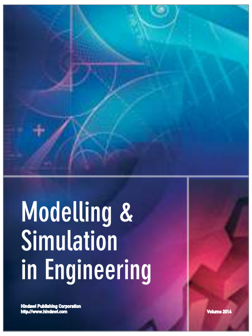
Journal of
Robotics

Journal of
Electrical and Computer
Engineering

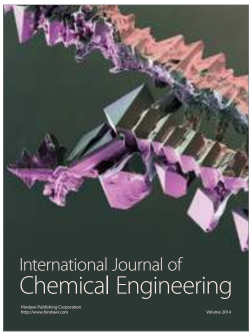

Advances in
OptoElectronics

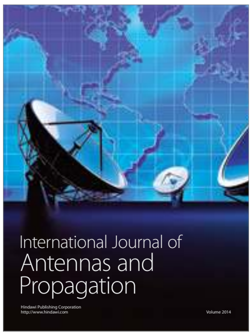
VLSI Design

International Journal of
Navigation and
Observation

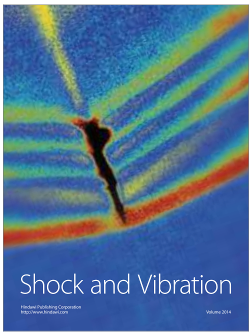

Modelling &
Simulation
in Engineering

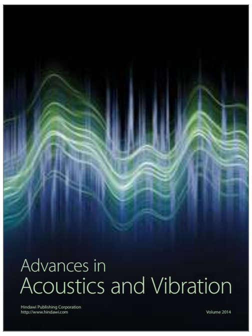
International Journal of
Aerospace
Engineering


International Journal of
Chemical Engineering


International Journal of
Antennas and
Propagation

Active and Passive
Electronic Components


Shock and Vibration


Advances in
Acoustics and Vibration