Research Article

Entropy based routing for mobile, low power and lossy wireless sensors networks

Celso Carvalho[1] [iD], **Edjair Mota[1], Eric Ferraz[2], Paulo Seixas[2], Paulo Souza[2], Vitor Tavares[2], Walfredo Lucena Filho[3], David Ferreira[1], Pietro Manzoni[4] and Carlos Calafate[4]**

Abstract

Routing protocol for low-power and lossy networks is a routing solution specifically developed for wireless sensor networks, which does not quickly rebuild topology of mobile networks. In this article, we propose a mechanism based on mobility entropy and integrate it into the corona RPL (CoRPL) mechanism, which is an extension of the IPv6 routing protocol for low-power and lossy networks (RPL). We extensively evaluated our proposal with a simulator for Internet of Things and wireless sensor networks. The mobility entropy-based mechanism, called CoRPL+E, considers the displacement of nodes as a deciding factor to define the links through which nodes communicate. Simulation results show that the proposed mechanism, when compared to CoRPL mechanism, is effective in reducing packet loss and latency in simulated mobile routing protocol for low-power and lossy networks. From the simulation results, one can see that the CoRPL+E proposal mechanism provides a packet loss reduction rate of up to 50% and delays reduction by up to 25% when compared to CoRPL mechanism.

Keywords

Wireless sensor networks, low-power and lossy networks, routing protocol for low-power and lossy networks, routing, mobility, Internet of Things

Introduction

Wireless sensor networks (WSNs)[1–4] occupy several environments such as residences, offices, and factories, and perform an important role in integrating smart objects to the Internet of Things (IoT).[5]

Motivating research in the area of IoT and WSNs forecasts point to the existence of about 50 billion devices connected to IoT in 2020,[6] which includes everyday objects such as mobile devices, cars, and also parts, products, and industrial machines. Opportunities also exist for the development of new applications, especially those involving mobility, in areas such as smart cities, robotics, e-health, and sports activities monitoring.

An example scenario that encourages research in WSNs is related to 5G networks that will be used as backhaul interface for WSNs. In 5G networks, small communication cells (SCCs) will coexist with the

[1]Universidade Federal do Amazonas (UFAM), Manaus, Brasil
[2]Sidia: Institute of Science and Technology, Manaus, Brasil
[3]Universidade do Estado do Amazonas, Manaus, Brasil
[4]Universitat Politècnica de València, Valencia, Spain

Corresponding author:
Celso Carvalho, Universidade Federal do Amazonas (UFAM), Av. Gen. Rodrigo Octávio 6200, Coroado I, Campus Universitário, Manaus, Amazonas 69080-900, Brasil.
Email: ccarvalho_@ufam.edu.br

macro-cells in order to increase the quality of communication in the coverage area, increase the transmission rate, and reduce the communication delay.[7] In this scenario, base stations (BSs) of SCs may be used to offer Internet access to WSNs' gateways, also called IoT border routers in IoT terminology. In this scenario, the IoT border routers can employ dedicated hardware or even be smart phones equipped with different technologies to communicate with sensors in the environment and to offer new smart applications.[8–10] From this perspective, it is seen in Ahamed and Faruque[11] that 5G networks will play an important role in services of remote monitoring and real-time control of devices that support machine-to-machine (M2M) and IoT communication such as connected cars, mobile robots, sensors installed in the industry, public infrastructure, and smart homes.[8]

The limitations of devices that compose the WSNs and IoT devices impact the routing protocol used by the wireless nodes. For this reason, specific protocols are required for this type of network.

Due to this, the IETF (Internet Engineering Task Force) developed the specifications of the routing protocol for low-power and lossy networks (RPL), designed to route data in low-power and lossy networks (LLNs).

RPL was initially developed to work on static network scenarios, since it dramatically reduces link maintenance when it assumes that the network has reached a permanent state.

In order to promote mobility support, some research in the literature[12–25] suggests adaptations to the RPL protocol. Among these researches, we noticed the development of solutions that have the following characteristics: (1) are designed for scenarios with few wireless mobile nodes (MNs),[12,13,16,17,19,21,24] (2) focus on the development of localization mechanisms that depend on a large number of fixed reference wireless nodes,[12,14,17–19,21,23,24] (3) use scenarios where MNs have fixed or low dynamic movement trajectories (e.g. line, grid, square),[12,14,16–18,23,24] and (4) are designed to improve downlink traffic[24] or to improve communications in scenarios with mobile root nodes and static wireless nodes.[13,18]

Among the works of the literature, we identified the CoRPL proposal[25] as being the one with the greatest relation to our research and that shows better performance. CoRPL proposes the division of the network into circular sub-areas called Coronas, with radius equal to the maximum transmission range of the wireless nodes—these areas being identified by a Corona ID (C_ID). In Gaddour et al.,[25] the CoRPL mechanism was evaluated through simulations and presented performance improvements when compared to the original

RPL in terms of packet loss rate of the links and average delay of the packets transmitted in the routes. To choose the preferred parent (PP) node of an MN, CoRPL considers only the Corona ID and packet loss rate of the links as criterion. However, to determine the PP of a MN, the CoRPL does not consider the degree of disorder of the MNs' in displacement in each Corona, which has an impact on the packet loss rate of the links.

In this article, we propose a mechanism residing in the MNs of an RPL network that extends the CoRPL mechanism.[25] In our proposal, we used the Corona ID, the mobility entropy, and the expected transmission count (ETX) metric in order to select among the next-hop MNs, which will be the PP node for an MN. The use of the mobility entropy in the proposed mechanism had the objective of selecting the PP node with the least movement instability among the possible next-hop PP nodes. The proposed mechanism was tested using the Cooja simulator,[26] in scenario of dozens of MNs moving with dynamic trajectory. RPL is currently considered the most important routing protocol in the IoT scenario, having the need to operate in applications with dynamic movement of several types of smart objects such as wearable gadgets, mobile robots, and unmanned air vehicles (UAVs).[27] In the simulated dynamic motion scenarios, the results showed positive aspects of the proposed idea: lower packet loss rate and lower average delay of packets transmitted in uplink toward the root nodes of an RPL network.

In order to present the subjects discussed, the article is divided into the following sections. The "Related work" section explains the RPL protocol and the concept of mobility entropy. The "Theoretical background" section discusses related works. The "CORPL + E architecture" section explains the architecture of the proposal CoRPL+E. The "Performance evaluation" section shows the results obtained. The "Conclusion" section presents conclusions and future works.

Related work

Lee et al.'s[12] study adjusts RPL parameters to improve performance of RPL in Vehicular Ad Hoc Network (VANET) scenarios. The adjustments made consist of (1) increasing the transmission frequency of probes used to determine the ETX metric, (2) ignoring the trickle timer used to periodically send DIO (DODAG Information Object) messages, and (3) ignoring the Delay-DAO (Destination Advertisement Object) timer used for DAO message transmission. As a result of the adjustments, the packet loss rate was reduced. However, the timer settings were executed statically for

scenarios with few MNs moving in fixed trajectories. Therefore, the work does not evaluate or propose solutions for scenarios with dynamic mobility or with greater node density.

Safdar et al.[13] propose adapting the RPL to create a hybrid protocol, both reactive and proactive, but for the scenario with multiple mobile root nodes (mobile roots (MRs)), not considering the mobility of MNs. In addition, the proposal was not evaluated experimentally or in simulations.

In Korbi et al.,[14] the authors present the Mobility-Enhanced RPL (ME-RPL), in which wireless nodes are classified as SNs (static nodes) and MNs. In ME-RPL, an SN is prioritized as the parent node. Next, the lowest ranking node is prioritized, and finally, if there is a tie, the objective function (OF) rules apply. Another change is the dynamic and more frequent delivery of DIS (DODAG (Destination-Oriented Directed Acyclic Graph) Informational Solicitation). Simulation results comparing RPL and ME-RPL demonstrate a reduction in packet loss and better route stability for the latter. However, the simulations were performed with a small amount of MNs in relation to the total number of nodes. In mobile networks, most nodes are in motion to impair the ME-RPL model.

The proposal of Hong and Choi[15] extends the RPL protocol to create the DMR (DAG (Destination Acyclic Graph)-based Multipath Routing for mobile sensor networks) protocol. The model uses rank as a metric in DIO messages to build the DODAG and to create alternate routes whenever possible using sibling nodes. The authors present performance simulation results of the proposed model, but the proposal is compared with networks that employ the AODV (On-demand Distance Vector),[28] and AOMDV (On-demand Multipath Distance Vector routing in ad hoc networks),[29] routing metrics. These two metrics were not designed for LLNs.

In Fotouhi et al.,[16] the authors integrate a handoff mechanism for RPL aiming to handle long latencies, high packet losses, and high overhead. The mechanism uses RPL control messages, such as beacons, and adds four timers that help notice degradation of links. Simulation results show that the mechanism reduces packet loss and handoff time but results in a large overload in networks with high transmission rates, and the handoff process is less responsive in low traffic networks. Moreover, the authors evaluated their proposal in scenarios with only one MN that uses several SNs as reference to execute handoff between them.

The Kalman Positioning RPL (KP-RPL)[17] divides the routing into hierarchical levels and employs the conventional RPL protocol between SNs and geographic routing between MNs and SNs. Thus, the authors use

an approach that is not normally employed in RPL networks.

Tahir et al.[18] adapt the RPL to scenarios with multiple mobile root nodes, not considering the mobility of MNs. The proposal is evaluated using mobility with semi-static behavior, which compromises performance evaluation.

EKF-MRPL (Enhanced Kalman Filter–Mobile RPL)[19] provides connectivity without interruptions to MNs while it is executing handover between parent SNs. However, the EKF-MRPL is able to determine the correct positioning of the MN, not aiming to propose an OF, and therefore was evaluated in a scenario composed of 25 reference SNs and 01 single MN limited to leaf node of the network.

Kharrufa et al.[20] propose D-RPL (Dynamic RPL) that implements a trickle timer called reverse which determines the transmission period of DIS messages. The authors use the MRHOF (minimum rank with hysteresis objective function) implemented in the Contiki operating system,[30] along with the ETX, energy, and Link Quality Indicators (LQI) metrics to propose their OF. However, Kharrufa et al.[20] do not clarify how the metrics were combined in its proposed D-RPL.

Sanshi and Jaidhar[21] propose the ERPL (Enhanced RPL) protocol that updates the PP of an MN when such MN moves away from its current PP. The proposal does not allow MNs to transmit DIO messages and thus become PP nodes. Note that making it impossible for MNs to become PPs can cause a network with many MNs to become disconnected.

In Wang et al.,[22] the authors propose monitoring the received signal strength indicator (RSSI) of DIO messages received by an MN to update its rank. Some flaws in the proposed algorithm are evident, since it is possible for an MN to increase its rank value when it moves toward its PP.

Fotouhi et al.[23] propose an interoperability framework between SNs and MNs, so an MN can execute handoff of PP node. However, the proposal, despite assuming the positional science of all nodes, is highly dependent on SNs for its correct operation and therefore is tested in a grid scenario with 13 reference SNs and only one MN moving with deterministic trajectory defined between the lines of the grid.

Carels et al.[24] propose a mechanism to reduce connectivity loss and packet loss in downlink transmission; however, this does not improve performance for uplink traffic.

The CoRPL[25] divides the network into circular areas, called coronas with radius equal to the maximum node transmission range, being identified by a Corona ID (C_ID). CoRPL incorporates C_ID into DIO messages, using it as the highest priority metric, when

Table 1. Articles that address mobility.

Paper	MN/MR/SN/(No. of nodes)	Solution	Scenario
Lee et al.[12]	Root node (1), MNs (10)	Static adjustment of parameters	10 MNs moving in a straight line
Safdar et al.[13]	MR (not available), MN (not available)	Hybrid protocol (reactive and proactive) for the scenario with multiple MRs	Not available
Korbi et al.[14]	MNs (6 and 9), SNs (5 and 12)	Differentiation between MNs and SNs	Line and grid
Hong and Choi[15]	Root nodes (4), MNs (100), only 30 MNs are source nodes	Multipath protocol for MN transmission	Random waypoint
Fotouhi et al.[16]	Root node (1), MNs (2, 4, 8, 12)	Integration of the smart-hop handoff mechanism using RSSI and proposed in Fotouhi et al.[31]	Straight line and between rows of a grid
Barcelo et al.[17]	MN (1), SNs (100)	Geographical and hierarchical routing	Square or zigzag trajectories
Tahir et al.[18]	MRs (5), SNs (130)	QuickBeta Algorithm that switches between RPL and routing backpressure	9 preset paths, 1 random
Bouaziz et al.[19]	MN (1), SNs (25)	Algorithm that estimates the future position of an MN; MNs are only network sheets	Random walk
Kharrufa et al.[20]	MNs (25), SN (1)	Reverse Trickle; OF that combines MRHOF function with ETX, energy, and LQI metrics	Random waypoint
Sanshi and Jaidhar[21]	MNs (9), SNs (30)	Algorithm that determines whether an MN is moving away from its PP; different OFs for SNs and MNs	Random waypoint
Wang et al.[22]	MNs (20, 40, or 60), root node (01)	Algorithm that monitors the RSSI to update the rank	Random waypoint
Fotouhi et al.[23]	MN (01), SNs (17)	Algorithm that performs soft or hard handoff	Grid
Carels et al.[24]	MN (01), SNs (36)	Mechanism that postpones the transmission of DAO messages for cancellation of routes; downlink traffic	Diagonal of a grid
Gaddour et al.[25]	MNs (20, 40, 60, 80, 100), root node (01)	Corona mechanism that the scenario in circular areas with radius is equal to the maximum transmission range of the nodes	Random waypoint

MN: mobile node; MR: mobile root; SN: static node; RPL: routing protocol for low-power and lossy networks; OF: objective function; MRHOF: minimum rank with hysteresis objective function; PP: preferred parent; DAO: Destination Advertisement Object; RSSI: received signal indicator; ETX: expected transmission count; LQI: link quality indicators.

compared to the ETX metric. However, considering the trajectory, its OF is not able to define which is the best MN of next hop within the same corona.

Table 1 shows that most of the works intend to improve the RPL performance but focus on the localization systems and, therefore, are highly dependent on static wireless reference nodes or low dynamic trajectories.[12,14,16,17,18,19,21,23] Some aim to improve the performance of RPL for downlink traffic,[24] others[13,18] aim to improve the RPL in the scenario of mobile root nodes. Works like ours appear in previous works.[20,22,25] The drawbacks of such approaches are that Kharrufa et al.[20] do not explain how the metrics are combined in their D-RPL proposal and[22] focus on updating the rank of MNs and not on the proposal of a new OF.

The results of the Wang et al.[22] show that, even in simulations with low packet transmission rates (e.g. 1 pkts/min), their proposal obtains packet delivery rate of up to 63% when the amount of MNs is equal to 60. In a similar situation, Gaddour et al.[25] show superiority obtaining a packet delivery rate of 67%, even when the packet transmission rate is much higher, with a value of 30 pkts/min. Compared to Gaddour et al.,[25] our proposal improves the OF of the cited article, since we use the mobility entropy as one of the criteria to select the best PP for an MN. Due to this, our proposal obtains better results in terms of packet loss rate and average packet delay, when varying packet's transmission rate, number of MNs, node's velocity, and number of root nodes, as shown in the next section.

Theoretical background

RPL protocol

RPL is an IPv6 protocol (Internet Protocol version 6) that builds a DODAG[32] to target the root node of the network. Each node maintains a table of neighbors containing possible parent nodes to reach the root node. One of the parent nodes is defined as preferred parent. The RPL defines the following ICMP-v6[19] route discovery messages: DIO, DIS, and DAO. DIO messages start from the root node and are propagated by the other DODAG nodes. Each network node uses a timer called trickle[33] that establishes the frequency of transmission of DIO messages to verify connectivity with nearby nodes. When a node verifies that there are differences between its neighborhood table and the network around it, the trickle timer increases the frequency of DIO messages and, when there are no inconsistencies, the trickle drastically reduces the transmission of DIO messages. DIO messages report DODAG characteristics such as the OF in use. The OF[34] informs the routing metrics used to give weight to the links. The rank value is a metric that indicates the position of a node, in number of hops, relative to the root node.[35] DIS messages are transmitted by nodes wishing to connect to DODAG. DAO messages propagate DODAG information from the network nodes toward the root node.[36]

The proposed mechanism in this article is an extension of the original RPL protocol and the CoRPL extension. Thus, the proposal herein preserves RPL protocol messages such as DIO, DIS, and DAO. As in Gaddour et al.,[25] the root node transmits DIO messages periodically, so that network topology changes are perceived more quickly.

Mobility entropy

Entropy is defined as the degree of disorder in a system.[37] This article uses the concept of entropy to create a mechanism that selects more stable links.

The entropy $H_m(t, \Delta t)$ (equation (1)) is defined by the F_m set of neighboring nodes of a node m in a time interval Δt. The entropy value is in the interval $(0 \leqslant H_m(t, \Delta t) \leqslant 1)$, with more stable links with entropy values close to 1 and unstable links close to 0

$$H_m(t, \Delta t) = \frac{-\sum_{k \in F_m} P_k(t, \Delta t) \log P_k(t, \Delta t)}{\log C(F_m)} \quad (1)$$

In equation (1), $C(F_m)$ is the cardinality degree of the subset F_m, formed by neighbors of a node m

$$P_k(t, \Delta t) = \left(\frac{a_{m,k}}{\sum_{i \in F_m} a_{m,i}} \right) \quad (2)$$
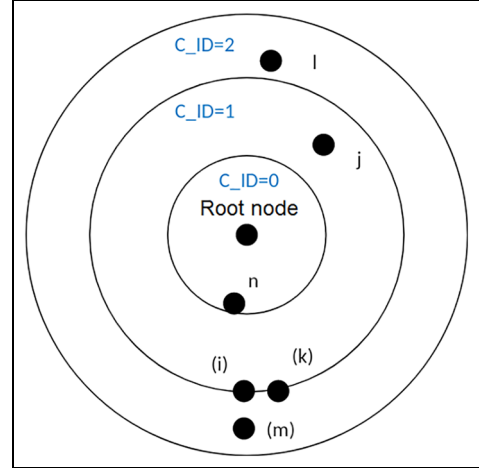

Figure 1. Criteria and example of PP selection in CoRPL + E proposal. Each MN uses as criteria for PP selection, the metrics Corona ID, mobility entropy, and the link quality metric.

$P_k(t, \Delta t)$ from equation (1) is detailed in equation (2), where $a_{m,n}$ is the mean relative mobility between two nodes (m, n). Equation (3) formally presents $a_{m,n}$, where N is the amount of discrete time instants in a time interval Δt

$$a_{m,n} = \frac{1}{N} \sum_{i=1}^{N} |v(m, n, t_i)| \quad (3)$$

In equation (4), $v(m, n, t)$ is the relative velocity between two nodes, where $v(m, t)$ is the velocity vector of a node m and $v(n, t)$ is the velocity vector of a node n

$$v(m, n, t) = v(m, t) - v(n, t) \quad (4)$$

In this article, we use the mobility entropy as a criterion used by a network node to determine its PP node. PP candidates with the mobility entropy closer to 1 are prioritized, as is presented in the following section.

CORPL+E architecture

In this article, we propose the CoRPL+E mechanism which is an extension of the CoRPL mechanism proposed in Gaddour et al.[25] For the implementation of CoRPL+E, all the available functionalities of CoRPL[25] were attached to Contiki. According to Figure 1, the proposal of Gaddour et al.[25] divides the network around the root node into concentric circular areas, with range given by the largest transmission radius of a wireless node, called Corona. As observed in example given in Figure 1, in the original CoRPL, if node m has C_ID equal to 2 and wishes to select its PP, node m will be able to choose nodes i and k, since C_ID is equal to 1. As a tiebreaker criterion, node m uses the link quality metric (e.g. ETX metric) to decide between nodes i and

Table 2. Neighbors' table example.

ID node	ID Corona	Mobility entropy	Link's quality metric
i	1	0.3	4
k	1	0.2	2

k. In our CoRPL+E proposal, we added a new PP node selection criterion. This criterion, the mobility entropy, has higher priority with respect to link quality metric.

The value of the mobility entropy is achieved through equation (1) and the result has a value from 0 to 1. The closer to 1, the more stable the link is.

We use Figure 1 and Table 2 to explain the idea of applying the Corona ID, the mobility entropy, and the quality metric of the link as criteria for PP node selection. This assumes that node m is stopped and placed at the point 0,0 of Cartesian plane x, y. With this in mind, in the following example, we can determine the speed of other MNs in relation to node m.

The node k moves from point (1,1) to the point (3,3) in the interval of 1 s and the node i moves from point (0,1) to the point (0,2) in the interval of 1 s. In this case, as will be shown in a later calculation, the nodes i and k will have mobility entropy, in relation to node m, equal to 0.3 and 0.2, respectively, whereas values closest to 1 determine more stable links. In our proposal, even if the link quality metric ETX of nodes i and k are 4 and 2, respectively, which favors the node k, we determine the node i as the PP for node m. For this, we justify that if node i retains the same velocity of displacement, at a future time, it will maintain more lasting links and tend to have better value of the ETX metric since it is not moving so abruptly. Using this criterion, the node m also selects a PP that will keep a longer connection, decreasing the quantity of needed PP changes, due to quality degradation of communication links. As shown in the performance evaluation section, this approach yields better results for the CoRPL+E mechanism when compared to CoRPL.

Mobility entropy applied to CoRPL

In Algorithm 1, we can see the CoRPL+E mechanism that resides in each MN (variable *thisMN* refers to the current MN). When an MN is started and does not receive DIO messages from neighbor nodes in 5 s of wait, the MN starts to continuously transmit DIS messages, asking for the transmission of DIO messages, until the MN receives DIO messages from neighbor nodes (lines 1–3). In line 4, upon receiving DIO messages, each MN initially reads its information (e.g. *C_ID*). The idea of the algorithm in lines ranging from 4 to 8 is to select nodes with smaller values of *C_ID* and

that, therefore, are closer to the root node. In line 5, an MN verifies if the parameter *C_ID* of a neighbor node (variable *DIO.C_ID*) is equal or smaller than the *C_ID* of neighbors that are stored in its neighbor table (variable *thisMN − >neighborsTable.C_ID*). In case the test result of line 5 is true, the MN stores the neighbor node ID in its list *thisMN − >list_ID* (line 6). Otherwise, the DIO message is deleted (line 8). The mobility entropy calculation is performed for MNs that have smaller *C_ID* values (line 10), and among these nodes, the one with the highest mobility entropy value is chosen as the PP (line 11). In cases where two or more nodes have the same mobility entropy value, the ETX metric is used as a decision factor. In line 12, it is verified whether there was a change of PP node and, if so, DIO messages are transmitted immediately in order to warn the other nodes of the network (line 13). Otherwise, the MN node waits for the time of the trickle timer to transmit a new DIO (line 14).

Algorithm 1. Choice of PP using CoRPL $+$ E mechanism

```
01: repeat
02:     thisMN → broadcast(DIS);
03: until thisMN → receive(DIO)
04: for each received DIO do
05:     if DIO.C_ID ≤ thisMN − > neighborsTable.C_ID then
06:         thisMN → store (DIO.MN_ID in thisMN → list_ID)
07:     else
08:         thisMN → deletes(DIO)
09: for each MN_ID in thisMN → list_ID do
10:     calculate_mobility_entropy(MN_ID)
11:     thisMN → PP.ID ← higher_entropy(thisMN → lista_ID)
12: if thisMN − >PP.ID has change then
13:     thisMN → broadcast(DIOs)
14: else
15:     thisMN → wait_until_trickle_timer_expire( ).
```

Choice of routes using CoRPL+E mechanism

Next, we show an example of the mobility entropy calculation for three nodes i, k, m that have positioning coordinates x, y in the plane. The nodes i and k are mobile, whereas the node m is static and is placed at coordinate $(0, 0)$. The node k moves from point $(-1, 0.5)$ to point $(1, 1)$ in time interval of 1 s and the node i moves from point $(-1, 1.5)$ to point $(-1, 1)$ in time interval of 1 s. Initially, we calculate the speed vector of the three nodes

$$\vec{vm} = (0, 0) - (0, 0) = (0, 0) \tag{5}$$

$$\vec{vk} = (3, 3) - (1, 1) = (2, 2) \tag{6}$$

$$\vec{vi} = (0, 2) - (0, 1) = (0, 1) \tag{7}$$

Applying the Pythagoras theorem

$$vm = \sqrt{0^2 + 0^2} = 0\,\text{m/s} \qquad (8)$$

$$vk = \sqrt{2^2 + 2^2} = 2.82\,\text{m/s} \qquad (9)$$

$$vi = \sqrt{0^2 + 1^2} = 1\,\text{m/s} \qquad (10)$$

Applying equation (4) as relative speed

$$v_{m,k} = 0 - \sqrt{8} = -2.82\,\text{m/s} \qquad (11)$$

$$v_{m,i} = 0 - \sqrt{1} = -1\,\text{m/s} \qquad (12)$$

Applying equation (2)

$$P_k(t, \Delta t) \rightarrow \left(\frac{-2.82}{-3.82}\right) = 0.73 \qquad (13)$$

$$P_i(t, \Delta t) = \left(\frac{-1}{-3.82}\right) = 0.26 \qquad (14)$$

Finally, the entropy between the nodes is calculated applying equation (1)

$$H_{m.k}(t, \Delta t) \rightarrow \left(\frac{-0.73 \times (-0.13)}{0.301}\right) 0.3 \qquad (15)$$

$$H_{m,i}(t, \Delta t) \rightarrow \left(\frac{-0.26 \times (-0.58)}{0.301}\right) 0.5 \qquad (16)$$

From these calculations, we conclude that the most stable link is between nodes m and i of Figure 1, since the mobility entropy value between these two nodes is closer to 1 (e.g. value 0.5).

Determination of nodes positioning in the CoRPL+E mechanism

To determine the velocity vector of the MNs, it was necessary to implement a mechanism, shown in Figure 2, to determine the position of the MNs during the simulation. We used the BonnMotion tool,[38] which can generate random motion mobility scenarios by writing in a log, the x, y position of each node of the network, according to Figure 2 (block 1). We installed its mobility plugin (block 2) into the Cooja simulator,[26] which uses the Contiki operating system.[30] The mobility plugin is capable of reading a motion log file generated by the BonnMotion tool to simulate the movement of the MNs in simulation (interaction between blocks 2 and 1). The CoRPL+E mechanism uses the speed vector of the MNs to determine the mobility entropy value of each MN in relation to the next-hop MN toward root nodes. To determine the velocity vector, it was necessary to determine the positioning of each node at each time instance of the simulation. For this propose, we modified the Move class coded in Java from the mobility plugin with the actual position of each MN to direct to a position vector stored in file, the ID and the
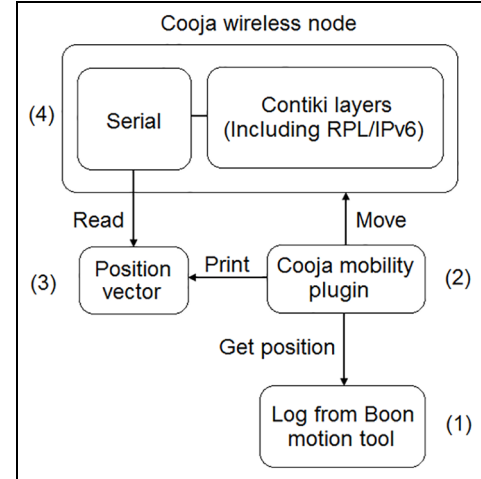

Figure 2. Architecture of the mechanism implemented in Cooja that determines the position x, y of each MN during simulations. The position is used to calculate the mobility entropy of MNs.

positions x and y, updated every second, of each MN present in the simulation (block 3). We used the Cooja wireless nodes serial library (block 4) so that each MN could access the nodes' position vector. In this case, the serial library simulates a communication interface with a location system which is coupled to the wireless node. Finally, we modified the code of rpl.c file of Contiki (block 4, Figure 2) in which the OF is determined so that each MN obtains its position x, y from the position vector reading (block 3). The architecture created aims to simulate MNs that interface with an external location system, for example, global positioning system (GPS)[39] and/or other method such as iterative trilateration.[40] Therefore, herein we focus on creating an OF to improve the routing process.

The CoRPL+E protocol

In the following two sections we present, respectively, the protocol already implemented in the CoRPL+E protocol, referring to modifications in the DIS and DAO messages. In the following section, we present the proposed implementation of part of the CoRPL+E protocol, related to the location of the MNs in the network, to be used for the determination of the entropy of mobility.

Messages implemented in the current CoRPL+E protocol. To implement the CoRPL+E mechanism, we extend the Cooja simulator with an approach similar to Gaddour et al.[25] For this, we modify the DIS and DIO messages as shown in Figures 3 and 4.

Figure 3 shows the structure of a DIS message that has the function of making probe of nearby DODAGs
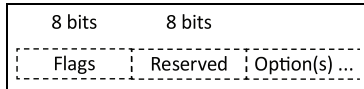
Figure 3. DIS messages of CoRPL + E and CoRPL. The binary values 1 and 2 in the two most significant bits of the flags field were used to identify DIS messages from CoRPL and CoRPL + E, respectively.
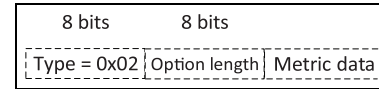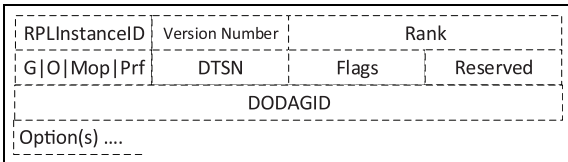

Figure 4. DIO message of the CoRPL + E and CoRPL protocols. We used the two most significant bits of the field flags with binary values 1 and 2 to represent the transmission of messages of CoRPL and CoRPL + E, respectively. The reserved field was used to transmit the value of *C_ID*.

and requesting DIO messages. The first field is reserved for the use of flags, is not used, and has 8 bits of length.[41] We used this field so that an MN can identify the DIS messages used in CoRPL (02 most significant bits with 01 binary values) or CoRPL+E (02 most significant bits with 10 binary values). Using some bits from the flags field to differentiate types of DIS messages allows MNs that implement only the RPL protocol to coexist with MNs that implement CoRPL or CoRPL+E mechanisms. This is because nodes that do not use CoRPL+E or CoRPL ignore the flags field so they can use the default OF of RPL. However, in this article, we use the flags field, as commented above, to execute individual simulations of the CoRPL and CoRPL+E mechanisms. In the implementation of this article, as in Gaddour et al.,[25] upon receiving a DIS message from a neighbor, the DODAG root broadcasts DIO messages immediately without any waiting time.

Figure 4 shows the structure of a DIO message used in the RPL protocol so that an MN can select its PP, discover, and select parameters of an RPL instance that is defined as one or more DODAGs that use the same OF.[41]

In the RPL protocol, the flags field present in DIO and DAO messages is reserved, unused, and has an 8-bit size. In the implementation of this article, we used the values 01 and 10 binary, in the 2 most significant bits, to represent messages of the CoRPL and CoRPL+E mechanisms, respectively. As in Gaddour et al.,[25] we used the 3 most significant bits of the 8-bit reserved field to represent the Corona ID (*C_ID*) that is used in CoRPL+E and CoRPL as a relative coordinate of MNs in relation to the network root nodes.

Figure 5. Format of the DAG Metric Container (DMC) option field that can load metrics (e.g. metrics that result in a sum or product) or recorded metrics of each link in a route.

Proposed distributed protocol for MN localization. In the CoRPL+E mechanism, we use metrics *C_ID*, mobility entropy, and ETX, respectively, as criteria for an MN to select its PP. In order to determine the value of the mobility entropy, an MN requires information about the positioning of the MNs, which are within its reach and have a lower *C_ID* value than its own.

As presented in the "Determination of nodes positioning in the CoRPL+E mechanism" section, in this article we employ a centralized simulation scheme, with a position vector that contains the position of all MNs in the simulation, in order that each MN determines the position of the candidate nodes to be its PP.

In this section, we discuss a proposed distributed protocol to allow MNs to exchange information of their positioning with each other and thus MNs can determine the speed vector of the candidates to be their PPs.

According to RFC 6550,[41] the DAG Metric Container (DMC) field may be present as an option field in DIO or DAO messages. In Figure 5, we introduce the DMC field that has up to 255 bytes of total length. The DMC field has the subfields type, option length, and metric data, which in this order carry the value 0×02 stating that the DMC field carries a metric, the total size of the DMC field, and, finally, the content and coding of the metric transported.

The DMC option may appear more than once in the same DIO or DAO message, for example, to accommodate a Metric Data field longer than 255 bytes. The DMC option field can contain several aggregated metrics (e.g. ETX metric), or recorded metrics, the latter being used when each node adds a sub-object reflecting the local value of a metric in a path.

In this sense, we suggest that the DMC field of DIO and DAO messages is carried as recorded metric; the *x*, *y* position information of each wireless node is present in the path between a given MN and the root node of the network, where positioning variables *x*, *y* can carry either global absolute positioning information (e.g. latitude and longitude) or relative node positioning information.[42] As an example of transporting absolute positioning information in the metric data field, 32 bits (4 bytes) are required to store latitude information and further 32 bits (4 bytes) are necessary to store longitude information. As an example, the latitude values 42.915512 (e.g. east) and longitude –99.521654 (e.g.

south) may be each stored as 32-bit integers when multiplied by the value 1000000.

In the following section, we discuss about existing positioning methods, listing the most relevant ones for implementing a proposal of distributed localization protocol to be used in conjunction with the CoRPL+E mechanism.

Discussion about positioning methods

The CoRPL+E is a routing mechanism that requires the location of the MNs in order to determine the mobility entropy of them. To locate unknown positioning MNs (called unknown nodes), special anchor nodes are required to be aware of their positions through the use of an absolute location system (e.g. GPS), relative location system, or because they have been placed in known coordinates. The unknown nodes use the anchor nodes, along with algorithms in order to determine their own locations.[43]

According to Halder and Ghosal,[42] localization algorithms can be classified into distributed, centralized, iterative, mobility assisted, and statistical. In the next paragraphs, we describe and compare these algorithms qualitatively.

In distributed algorithms, the unknown nodes are located through the exchange of messages with neighboring anchor nodes that can be more than a hop away. In the distributed system, the smaller the number of anchor nodes, the greater the need for the localization algorithm to use spatial information exchanged between the unknown nodes themselves, which makes the location inaccurate. Thus, this type of algorithm has the disadvantage of having a large density of anchor nodes to promote the location. However, using wireless sensors with new physical layer patterns such as IEEE 802.15.4g (Smart Grid Standards for Operation in Sub-1 GHz Bands) can reduce the number of anchor nodes used as references for locating other MNs. In this sense, it is reported in Oh et al.[44] that the IEEE 802.15.4g physical layer when using bandwidths ranging from megahertz to kilohertz can transmit at distances ranging from tens of meters to up to 20 km. Increased communication distance makes it possible for MNs to locate themselves over long distances with few reference anchor nodes in the network.

In the centralized algorithms, the unknown nodes participate in a network where there is exchange of messages between the wireless nodes and the root node, the latter node determining the location of the other wireless nodes. The centralized algorithms have the advantage of acuricity and disadvantage as they are expensive in terms of message exchange, making these algorithms have scalability problems, especially in mobile networks. By using physical layer patterns with higher transmission range (e.g. IEEE 802.15.4g), it is possible

to employ the centralized approach in mobile WSNs. In this sense, MNs would communicate with anchor nodes and these, having greater battery capacity, would execute algorithms to determine the location of MNs. However, for this to be possible, it is necessary to use special anchor nodes with greater processing power to run the centralized location algorithm and consequently batteries with greater energy levels.

Iterative algorithms apply the centralized approach using clusters formed by wireless nodes, creating sub-areas of location. Iterative algorithms are used by distributed algorithms to refine the initially determined location. For this, the iterative algorithms add more spatial information and distance measures between the nodes. We comment that the CoRPL+E mechanism uses the concept of Coronas, areas determined by the greater transmission range of the wireless nodes. We have seen that although RPL networks work in the DODAG structure, it is possible to create virtual cluster structures within the areas of each Corona. We add that iterative algorithms can be considered an option for distributed implementation of a localization system for the CoRPL+E mechanism in order to reduce the overhead of the centralized algorithms and the number of anchor nodes required for the distributed algorithms.

The mobility-assisted algorithms use mobile anchor nodes, equipped with GPS, that move along with the other MNs and use beacons to periodically emit their location to the unknown nodes. We realize that this type of architecture offers great benefits since mobile anchor wireless nodes can be nodes that have lower energy restriction (e.g. mobile robots), when compared with the other MNs in the network. In addition, we noticed that the localization process of MNs can be improved by moving the mobile anchor nodes in paths that facilitate the process of locating the unknown nodes. As commented, the CoRPL+E mechanism uses Coronas to divide the area of the network into sub-areas. As the MNs of the network have mobility, the sub-areas of the Coronas also are mobile. In this sense, one can use the mobility of the moving anchor nodes to move them strategically next to the area of each Corona, providing coverage of the location beacons for the largest possible number of MNs. We add that mobile anchor nodes can be represented by robots with few energy constraints and that can easily carry GPS devices to determine their own locations. We conclude that the mobility-assisted algorithms are an option for the implementation of a localization system for CoRPL+E mechanism.

Statistical methods use signal signatures, measured at points where previous signal measurements were performed. The signatures infer the location by exploring the relationship between distances and characteristics measured in the signals for a particular measurement point. These algorithms can work with the presence of

noise and uncertainties of signal measurements. We note that statistical methods can be used mainly to detect the position of a node when it passes in points of the space in which previously signal measurements have already been made. Thus, there is a previous stored signature for that particular measurement point. In order to make it possible to identify the signature of the signals, it is necessary to have static wireless nodes in known positions that can be used as reference by the unknown nodes. We realized that this approach may require a large amount of static wireless reference nodes to provide signal signatures for all points in the space through which the MNs of the network can move. However, we also have noticed that SNs with greater communication range can transmit location beacons at a much greater distance when using, for example, the IEEE 802.15.4g physical layer. For this situation, it is noteworthy the feasibility of using statistical methods to provide localization.

Threats and preventions to RPL-based protocols

Security is one of the challenges to be addressed in IoT, with secure packet routing being a critical issue. In the context of RPL, hereafter we make a brief list of threats and preventive actions related to denial-of-service attack, black hole and gray hole attack, sinkhole attack, identity-based attacks, and wormhole attacks.[45]

One type of threat to wireless networks, including those based on the RPL protocol, is the black hole and gray hole attack, the latter also called selective forwarding attack.[46] In this attack, the malicious node drops all or a portion of the packets it would forward. In this selective forwarding attack, for example, an attacker can forward all RPL control messages and discard all data packets. For the black hole and gray hole attack, Raza et al.[47] performed measurements of the amount of transmitted DIO messages, packet loss rate, and delay of packets in order to identify the attackers. The authors of the article proposed that a wireless node informs its neighboring nodes after detecting the malicious behavior, in order that the attacking node is placed in a blacklist.

Another type of threat is the sinkhole attack where the malicious node informs its neighbors that has a low rank value, in other words, that is near to the root node of the DODAG. In the case of the CoRPL and CoRPL+E protocols, the attacker could report a low Corona ID value. With this, the attacker's neighboring nodes would choose the malicious node as PP. Finally, the malicious node upon receiving packets transmitted by neighboring nodes would discard such packets. To address this problem, Perrey et al.[48] proposed using hash chains to sign the DIO messages used to carry the rank information. In this approach, the receiving node of the DIO message is able to authenticate the message,

verifying whether any node in the path has announced a false rank value. A similar approach can be employed in CoRPL+E to protect the protocol from the sinkhole threat.

One more type of threat to RPL-based networks is the identity-based attack. In this type of attack, a malicious node forges the identity of another MN in order to control and attack the network. One variant of the identity attack is the Sybil attack where an attacker forges several virtual copies of a single physical node.[49] CoRPL+E uses node positioning to perform routing. Thus, in CoRPL+E, it is possible to implement a mechanism to mitigate identity attacks by verifying the identity of the node that must be present only in a single position in the network.

In the wormhole attack, malicious nodes modify routing tables, creating a tunnel to deviate traffic from one location to another. With this attack, it is possible, for example, that attackers deviate traffic destined for the root nodes and then carry out the sinkhole attack concomitantly. CoRPL+E uses geographic information to determine the routing entropy. This geographic information when present in the routing tables and used in conjunction with traffic information can be used to identify this type of attack, as done in Wallgren et al.[46]

Among the possible threats are denial of service attacks. For example, RPL-based protocol packets carry information such as the 1-bit O and R flags. The first flag indicates whether a packet is transmitted in downlink (value 1) or uplink (value 0) and the flag R indicates error in the rank of the transmitter or receiver nodes. An attacking wireless node may generate a loop, if upon receiving a packet, for example in uplink, set the O flag to 1 to indicate downlink and then forward the packet in the uplink direction.[50] In RPL, if the receiver node detects the loop, it will set the rank error R flag to 1 and then make a drop of the packet. These flag changes performed maliciously and at high frequency cause a reset of the trickle timer, resulting in a denial of service attack, due to the large number of control messages on the network. To work around this problem, RFC 6553[51] proposes limiting the amount of resets of the trickle timer to 20. When reaching this threshold, the inconsistent packets are discarded, without resetting the trickle timer. Taking this possibility into account, Mayzaud et al.[50] dynamically adjust the threshold of inconsistencies that cause reset in the trickle timer. In the proposal of Mayzaud et al.[50] the threshold is reduced proportionately to the aggressiveness of the attack and increases slowly when the attack is stopped. This approach is effective in reducing this type of denial of service attack and can therefore be used in networks based on the RPL protocol, including CoRPL+E.

Table 3. Parameters used in simulation.

Fixed parameters	Value			
Area	$500 \times 500\ m^2$			
Emulated node	Tmote sky			
Transmission power	0 dBm			
Packet type	UDP (User Datagram Protocol)			
Packet size	127 bytes			
Transmission range radius	50 m			
Simulation time	60 min			
Mobility model	Random waypoint			
Variable parameters	Number of MNs	Number of root nodes	MNs' speed (m/s)	Transmission rate (pkts/min)
Packet transmission rate	10–20–40–60–80–100	3	3	10, 30, 60
MNs' velocity	10–20–40–60–80–100	3	1, 2, 3, 4	30
Number of root nodes	10–20–40–60–80–100	1, 2, 3	3	30

MNs: mobile nodes; UDP: user datagram protocol.

Performance evaluation

The performance evaluation of the CoRPL+E proposal was performed by comparing the results of packet loss rate and average delay through simulation experiments, with variations of transmission rates, MN speed, and number of root nodes. The mobility model used is the Random Waypoint Mobility Model,[52] whose operation consists of choosing a random position for which the MN moves with random speed. The velocity is defined within the interval $[0, V_{MAX}]$, where V_{MAX} is the maximum speed allowed for the nodes. The V_{MAX} variable of the Random Waypoint model is defined in the MNs' speed parameter of Table 3. For performance evaluation, we used a set of 10 quad core computers, with 16 GB RAM, network connected and equipped with Ubuntu 16.04. We use shell script to run Cooja on each of the network connected computers. In Cooja, we create code in Java script programming language that calculates in each simulation round the packet loss rate of the links and the average delay of the established routes. At the end of each simulation round, the shell script directs the packet loss rate and average packet delay results to a file to generate graphs.

Evaluation scenarios

Table 3 presents the simulation parameters.[25] In each round of simulation, each MN originates packets destined to the root nodes, according to the transmission rate specified in Table 3. The simulations were performed in scenarios with 1, 2 or 3 root nodes. In the scenario with 01 root node, this was positioned at the point (250 *m* 250 *m*); In the scenario with 02 roots, they were positioned in the points (125 *m*, 250 *m*) and

(325 *m*, 250 *m*); and in the scenarios with 03 root nodes, they were positioned individually at (125 *m*, 125 *m*), (375 *m*, 125 *m*) and (250 *m*, 375 *m*) points.

In the simulations, we aim to answer the following research questions. Is it possible to use the CoRPL+E mechanism to reduce, in relation to the CoRPL mechanism, the packet loss rate and average delay of the packets transmitted from the MNs to the root nodes of an RPL network? Is it possible to reduce packet loss and the average packet delay in networks with MNs that transmit packets with different rates, move at different speeds, and even in scenarios where there are different amounts of root nodes in the network?

Results

The results were generated using the average of 100 simulation values with a confidence interval of 95%.

Figure 6 shows the results of packet loss with varying transmission rates of the nodes for CoRPL and the CoRPL+E proposal.

Figure 6 shows the results of packet loss with variation of the node's transmission rate for CoRPL and proposed CoRPL+E when we used 03 root nodes. The results demonstrate a decrease in the packet loss rate as the number of nodes increases. This behavior is expected because the larger the number of nodes in the same area, the smaller the distance between them, which results in better link quality. With this, nodes have more route choices. Comparing the graphs in Figure 6(a) and (b), one can see that the CoRPL+E proposal obtains lower packet loss rate values for 13 out of the 18 average values presented. The percentage of packet loss rate reduction in favor of CoRPL+E reached 27.45%, when the number of nodes is 100 and
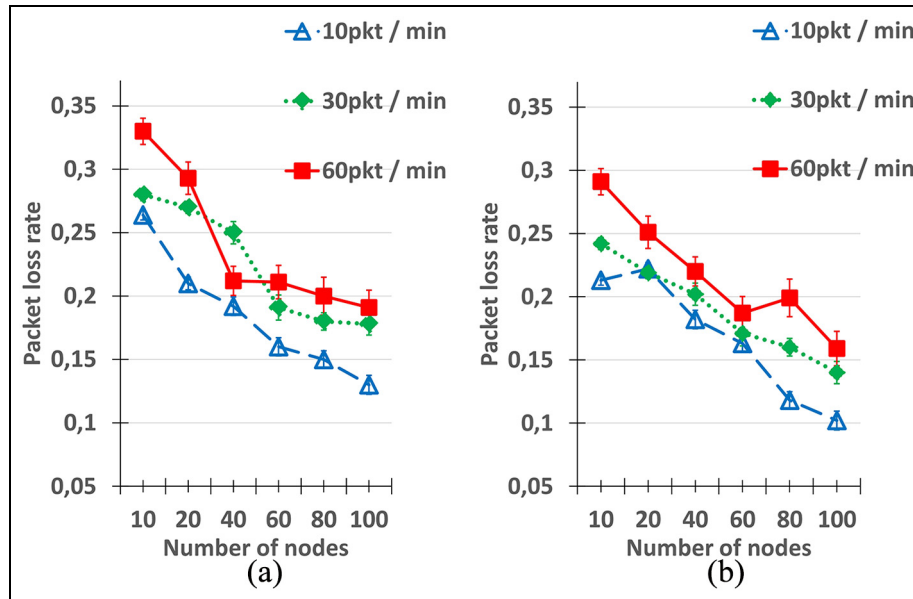
Figure 6. Packet loss rate obtained by fixing nodes' velocity and with variation of nodes' packet transmission rate: (a) CoRPL and (b) CoRPL + E.

Table 4. The p values for two-sample t-test used to verify if the average values of packet loss rate, shown in Figure 6 and obtained using the CoRPL and CoRPL+E mechanisms, differ statistically at the level of significance (α) of 0.05.

pkts/min	Mechanism	Number of nodes											
		10		20		40		60		80		100	
10	CoRPL	0.26	$p < 0.05$	0.21	$p = 0.18$	0.19	$p = 0.20$	0.16	$p = 0.73$	0.15	$p < 0.05$	0.13	$p < 0.05$
	CoRPL+E	0.21		0.22		0.17		0.16		0.12		0.10	
30	CoRPL	0.28	$p < 0.05$	0.27	$p < 0.05$	0.25	$p < 0.05$	0.19	$p < 0.05$	0.18	$p < 0.05$	0.17	$p < 0.05$
	CoRPL+E	0.24		0.22		0.20		0.17		0.16		0.14	
60	CoRPL	0.33	$p < 0.05$	0.29	$p < 0.05$	0.21	$p = 0.42$	0.21	$p < 0.05$	0.20	$p = 0.92$	0.19	$p < 0.05$
	CoRPL+E	0.29		0.25		0.22		0.19		0.20		0.16	

The p values greater than 0.05 mean that there is insufficient evidence to conclude that the means of the samples differ at the level of significance of 0.05.
The level of significance used to compare the samples of CoRPL and CoRPL+E was 0.05 for all the tests, including the cells in gray shading.

when the transmission rate of packets has a value of 10 pkts/min.

Table 4 shows numerically the packet loss rate values of the CoRPL and CoRPL+E mechanisms exposed in Figure 6. In addition, Table 4 presents p-value results for two-sample t-test. The two-sample t-test was used to verify if the samples applied to determine the average of the results of the CoRPL and CoRPL+E mechanisms, shown in Figure 6, had statistical differences in the level of significance (α) of 0.05. In Table 4, p values lower than 0.05 mean that, with 95% confidence level, there is a difference between the average of the results obtained by using the CoRPL and CoRPL+E mechanisms. In Table 4, we visualize cells in gray color where the p values were greater than 0.05. In these cases, because the results obtained when using both mechanisms are very close, there is insufficient evidence to conclude that the average of the samples differs in the level of significance of 0.05. The results presented in Table 4 show that there are 13 packet loss rate results obtained by using the CoRPL+E mechanism, which have average values different from the results obtained when using the CoRPL mechanism. In these 13 results, as commented in Figure 6, the CoRPL+E mechanism obtained lower packet loss rate values than those of the CoRPL mechanism.

Figure 7 shows that, for both CoRPL and CoRPL+E, the average delay decreases as the number of nodes increases. Figure 7(a) and (b) shows an average delay reduction of up to 1.6 s in favor of the CoRPL+E proposal when the transmission rate is 30 pkts/min and when the number of nodes is equal to 80. The best results in favor of CoRPL+E are shown in Figures 6 and 7, since MNs tend to establish and maintain established routes for longer times before a route repair is needed in our proposal.
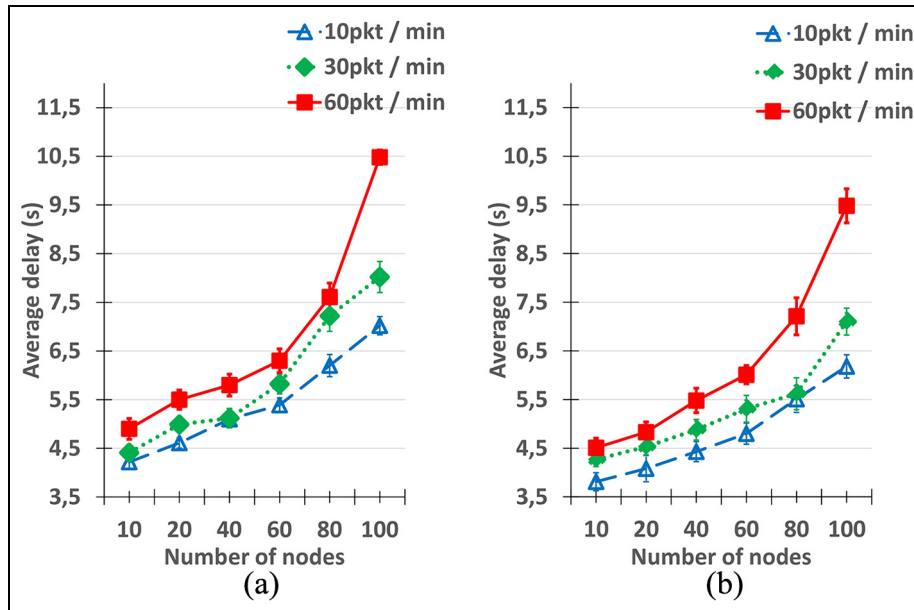
Figure 7. Packet's average delay with fixed nodes' velocity and variation of the packet transmission rate: (a) CoRPL and (b) CoRPL + E.

Table 5. The p values for two-sample t-test used to verify if the average packet delay values presented in Figure 7 and obtained using the CoRPL and CoRPL+E mechanisms differ statistically at the significance level (α) of 0.05.

pkts/min	Mechanism	Number of nodes											
		10		20		40		60		80		100	
10	CoRPL	4.22	$p < 0.05$	4.61	$p < 0.05$	4.1	$p < 0.05$	5.39	$p < 0.05$	6.2	$p < 0.05$	7.02	$p < 0.05$
	CoRPL+E	3.81		4.08		4.43		4.8		5.51		6.18	
30	CoRPL	4.41	$p = 0.11$	4.99	$p < 0.05$	5.12	$p < 0.05$	5.82	$p < 0.05$	7.22	$p < 0.05$	8.02	$p < 0.05$
	CoRPL+E	4.27		4.53		4.88		5.31		5.62		7.1	
60	CoRPL	4.9	$p < 0.05$	5.5	$p < 0.05$	5.8	$p < 0.05$	6.3	$p < 0.05$	7.61	$p < 0.05$	10.4	$p < 0.05$
	CoRPL+E	4.51		4.83		5.48		6.01		7.21		9.48	

The p values greater than 0.05 mean that there is insufficient evidence to conclude that the means of the samples differ at the level of significance of 0.05.

In Table 5, we verify that the results of average packet delay obtained with the CoRPL+E mechanism and presented in Figure 7 are, with the exception of one result that is in a cell painted in gray, lower than those obtained with the use of the CoRPL mechanism. For the result with exception, it cannot be affirmed that the average values obtained through the use of the mechanisms CoRPL and CoRPL+E differ at the level of significance of 0.05.

In Figure 8, we observe that as the velocity of nodes increases, the packet loss rate also increases for both CoRPL and CoRPL+E, since there is a greater number of link breaks between MNs and their PPs. Comparing Figure 8(a) and (b), we observe a reduction in the packet loss rate when using the CoRPL+E proposal.

The reduction is around 50% when the nodes' velocity has a value of 1 m/s and the number of nodes assumes the value of 80 and 100 nodes.

In Table 6, we show numerically the packet loss rate results of Figure 8. In the table, we find that the CoRPL+E mechanism, when compared to the CoRPL mechanism, obtained in most cases, lowers packet loss rate values. In 19 of the 24 average packet loss rate results presented in the table, the CoRPL+E mechanism obtained better results, with 95% confidence level.

Figure 9 shows that the higher the velocity and the greater the number of nodes, the greater packets' average delay is transmitted to the root node. This result occurs because as the node density increases, the distance between the nodes decreases and, consequently,
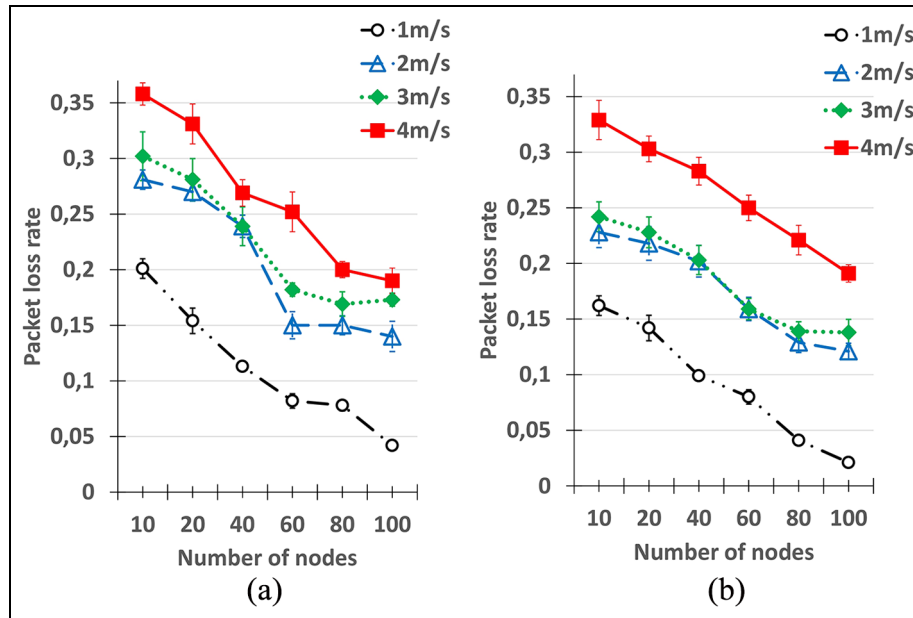
Figure 8. Packet loss rate obtained with variation of node's velocity and fixing the packet transmission rate: (a) CoRPL and (b) CoRPL+E.

Table 6. The p values for two-sample t-test used to verify that the average packet loss rate values presented in Figure 8 and obtained using the CoRPL and CoRPL+E mechanisms differ statistically at the level of significance (α) of 0.05.

Nodes' velocity (m/s)	Mechanism	Number of nodes											
		10		20		40		60		80		100	
1	CoRPL	0.20	$p < 0.05$	0.15	$p < 0.05$	0.11	$p < 0.05$	0.08	$p = 0.09$	0.08	$p < 0.05$	0.04	$p < 0.05$
	CoRPL+E	0.16		0.13		0.09		0.07		0.04		0.02	
2	CoRPL	0.28	$p < 0.05$	0.27	$p < 0.05$	0.24	$p < 0.05$	0.15	$p = 0.27$	0.15	$p < 0.05$	0.14	$p < 0.05$
	CoRPL+E	0.23		0.22		0.20		0.16		0.13		0.12	
3	CoRPL	0.30	$p < 0.05$	0.28	$p < 0.05$	0.24	$p < 0.05$	0.18	$p < 0.05$	0.17	$p < 0.05$	0.17	$p < 0.05$
	CoRPL+E	0.24		0.23		0.20		0.16		0.14		0.14	
4	CoRPL	0.36	$p < 0.05$	0.33	$p < 0.05$	0.27	$p = 0.11$	0.25	$p = 0.85$	0.20	$p < 0.05$	0.19	$p = 0.88$
	CoRPL+E	0.33		0.30		0.28		0.25		0.22		0.19	

The p values greater than 0.05 mean that there is insufficient evidence to conclude that the means of the samples differ at the level of significance of 0.05.

the length of the links between the nodes decreases. With this, a packet goes through more hops when traveling from an MN to the root node, and both the CoRPL and CoRPL+E mechanisms do not consider the delay in determining routes. Figure 9 shows the average delay gains for CoRPL+E compared to CoRPL, for all number of nodes and all simulated nodes' velocity values. For simulations with 100 nodes and nodes' velocities of 4 m/s, for example, the average delay reaches 12 s for CoRPL, and for CoRPL+E, the average delay does not exceed 9 s.

In Table 7, we present average packet delay values, shown in Figure 9. In this table, we see numerically that the CoRPL+E mechanism obtains lower average delay results when compared to the CoRPL mechanism

in 21 of 24 of the values presented in Table 7. For three test cases, the results obtained when using the CoRPL and CoRPL+E mechanisms are similar and, therefore, it cannot be stated with 95% confidence level that the CoRPL+E mechanism got lower results than those obtained with the CoRPL mechanism, and vice versa.

In Figures 10 and 11, we vary the number of root nodes in the network by measuring the packet loss rate and average delay transmitted by the MNs when they are moving at a speed of 3 m/s and transmitting packets with rate of 30 pkts/min. As shown in Figure 10, the results obtained by using CoRPL+E when compared to CoRPL show reduced packet loss rate that varies from 22% to 17%, when there were 10 MNs and the number of root nodes was equal to 1 and 3, respectively. Similar
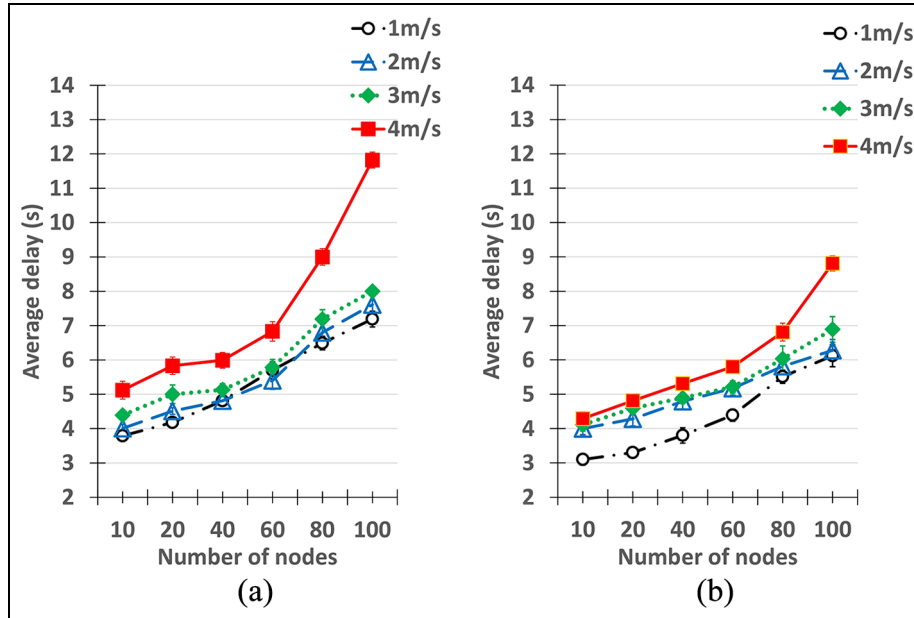
Figure 9. Average delay with variation of nodes' velocity and with fixed packet transmission rate: (a) CoRPL and (b) CoRPL + E.

Table 7. The p values for two-sample t-test used to verify if the average packet delay values presented in Figure 9 and obtained using the CoRPL and CoRPL+E mechanisms differ statistically at the significance level (α) of 0.05.

Nodes' velocity (m/s)	Mechanism	Number of nodes											
		10		20		40		60		80		100	
1	CoRPL	3.79	$p < 0.05$	4.18	$p < 0.05$	4.81	$p < 0.05$	5.7	$p < 0.05$	6.49	$p < 0.05$	7.19	$p < 0.05$
	CoRPL+E	3.10		3.31		3.81		4.39		5.51		6.12	
2	CoRPL	4.01	$p = 0.92$	4.51	$p < 0.05$	4.81	$p = 0.87$	5.4	$p = 0.16$	6.8	$p < 0.05$	7.61	$p < 0.05$
	CoRPL+E	4.01		4.28		4.79		5.17		5.81		6.28	
3	CoRPL	4.39	$p < 0.05$	5.01	$p < 0.05$	5.13	$p < 0.05$	5.78	$p < 0.05$	7.19	$p < 0.05$	8.11	$p < 0.05$
	CoRPL+E	4.11		4.59		4.89		5.21		6.03		6.89	
4	CoRPL	5.12	$p < 0.05$	5.83	$p < 0.05$	5.99	$p < 0.05$	6.83	$p < 0.05$	9.01	$p < 0.05$	11.8	$p < 0.05$
	CoRPL+E	4.29		4.81		5.31		5.81		6.81		8.81	

The p values greater than 0.05 mean that there is insufficient evidence to conclude that the means of the samples differ at the level of significance of 0.05.

packet loss rate reductions were obtained when there were 100 MNs and when the number of root nodes was equal to 1 and 3. In Figure 11, we observe average packet delay reductions for the CoRPL+E proposal in the described scenario. This result appears more expressive when the amount of MNs is equal to 40 or more. For example, when the number of MNs is equal to 100 and the number of root nodes is equal to 1 and 2, respectively, the gains of CoRPL+E reach up to 1 s compared to CoRPL. When the number of root nodes increases to 3 and the number of MNs remains at 100, CoRPL+E gains reduce to 0.86 s. When there are less root nodes, the CoRPL+E proposal can establish less delayed routes. As the number of root nodes increases, traffic flow possibilities for these root nodes also

increase and so CoRPL gets closer to our CoRPL+E proposal.

Tables 8 and 9 present, respectively, the results of average packet loss rate and average packet delay presented in Figures 10 and 11. In addition, we present the results of p value for two-sample t-test, verifying whether the samples used to determine the graphs of Figures 10 and 11 have average values that differ statistically at the level of significance (α) of 0.05.

In Table 8, we see that the CoRPL+E mechanism obtained smaller values of average packet loss rate, with 95% confidence level, when compared with the CoRPL mechanism in 15 of 18 simulated test cases. In Table 9, we see that the CoRPL+E mechanism obtained smaller average packet delay values, with 95% confidence level,
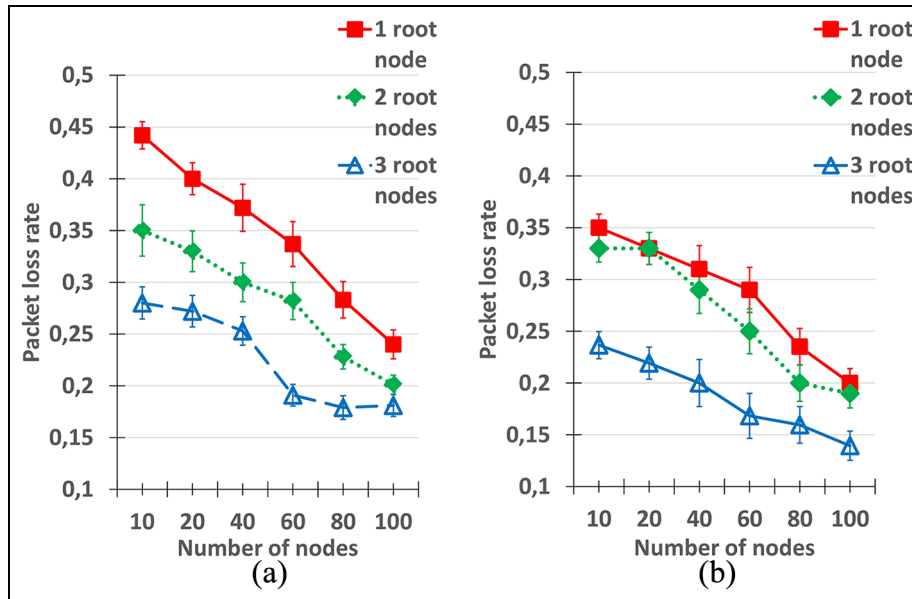
Figure 10. Packet loss rate with variation in the number of root nodes and fixed packet transmission rate and MNs' speed: (a) CoRPL and (b) CoRPL+E.
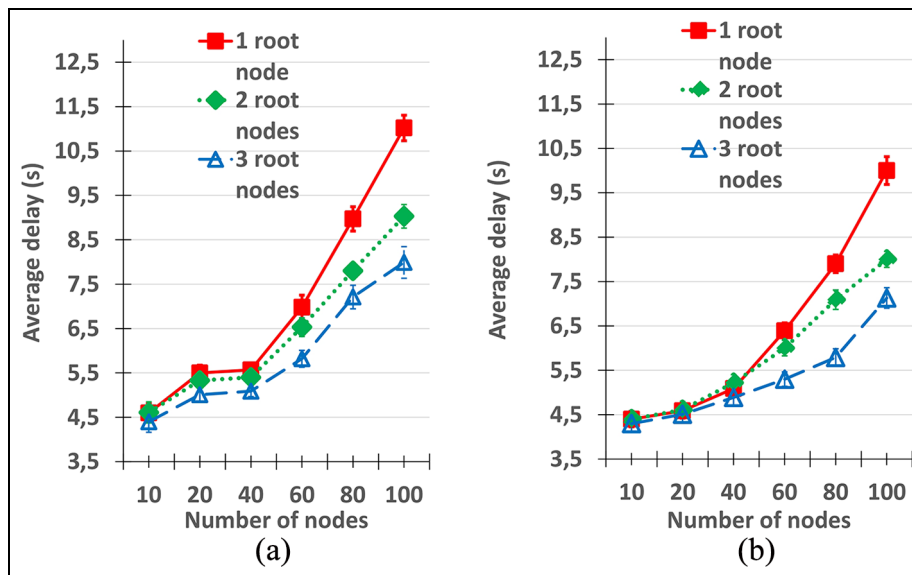

Figure 11. Average packets' delay with variation in the number of root nodes and fixed packet transmission rate and MNs' speed: (a) CoRPL and (b) CoRPL+E.

when compared to the CoRPL mechanism in 14 of 18 simulated test cases.

Conclusion

In this article, we propose the CoRPL+E mechanism that integrates mobility entropy into the CoRPL

extension to reduce packet loss rate and average packets' delay in simulated RPL networks that have MNs.

More specifically, the proposed mechanism includes an OF that uses Corona ID, mobility entropy, and link quality metric as criteria for an MN to select the PP of MNs. The use of mobility entropy as a criterion allows for the selection of more stable links with more lasting

Table 8. The p values for two-sample t-test used to verify that the average packet loss rate values presented in Figure 10 and obtained using the CoRPL and CoRPL+E mechanisms differ statistically at the level of significance (α) of 0.05.

No. of root nodes	Mechanism	Number of nodes											
		10		20		40		60		80		100	
1	CoRPL	0.44	$p < 0.05$	0.40	$p < 0.05$	0.37	$p < 0.05$	0.34	$p < 0.05$	0.28	$p < 0.05$	0.24	$p < 0.05$
	CoRPL+E	0.35		0.32		0.31		0.29		0.24		0.21	
2	CoRPL	0.35	$p = 0.34$	0.33	$p = 0.80$	0.30	$p = 0.62$	0.28	$p < 0.05$	0.23	$p < 0.05$	0.20	$p < 0.05$
	CoRPL+E	0.33		0.33		0.29		0.25		0.2		0.19	
3	CoRPL	0.28	$p < 0.05$	0.27	$p < 0.05$	0.25	$p < 0.05$	0.19	$p < 0.05$	0.18	$p < 0.05$	0.18	$p < 0.05$
	CoRPL+E	0.24		0.22		0.20		0.17		0.16		0.14	

The p values greater than 0.05 mean that there is insufficient evidence to conclude that the means of the samples differ at the level of significance of 0.05.

Table 9. The p values for two-sample t-test used to verify that the mean packet delay values shown in Figure 11 and obtained using the CoRPL and CoRPL+E mechanisms differ statistically at the significance level (α) of 0.05.

No. of root nodes	Mechanism	Number of nodes											
		10		20		40		60		80		100	
1	CoRPL	4.60	$p = 0.14$	5.50	$p < 0.05$	5.57	$p < 0.05$	6.98	$p < 0.05$	8.97	$p < 0.05$	11.0	$p < 0.05$
	CoRPL+E	4.40		4.59		5.09		6.39		7.90		10.0	
2	CoRPL	4.61	$p = 0.15$	5.33	$p < 0.05$	5.40	$p = 0.15$	6.53	$p < 0.05$	7.8	$p < 0.05$	9.03	$p < 0.05$
	CoRPL+E	4.40		4.61		5.22		6.00		7.09		8.00	
3	CoRPL	4.61	$p < 0.05$	5.01	$p < 0.05$	5.09	$p = 0.07$	5.82	$p < 0.05$	7.21	$p < 0.05$	7.99	$p < 0.05$
	CoRPL+E	4.3		4.53		4.88		5.31		5.79		7.13	

The p values greater than 0.05 mean that there is insufficient evidence to conclude that the means of the samples differ at the level of significance of 0.05.

connections between an MN and its PP. This reduces the use of RPL repair mechanisms, thus reducing the packet loss rate and packet delay in the network.

The proposal was extensively evaluated using the Cooja simulator for WSNs. The results of the simulations of our CoRPL+E proposal were compared to the results obtained using the CoRPL mechanism, and the mechanism proposed in this article presented better performance.

For future studies, we suggest developing a protocol that uses iterative multilateration[40] for MNs to determine their positions, using a small number of MNs that are equipped with GPS system as a reference.[39] This mechanism can act alongside the OF that uses the mobility entropy to choose the best routes in the network.

In addition, we intend to validate the CoRPL+E proposal by testing its performance in real networks using the Contiki operating system and compatible sensors.

Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

ORCID iD

Celso Carvalho https://orcid.org/0000-0002-7378-8893

References

1. Blanco-Novoa O, Fernández-Caramés TM, Fraga-Lamas P, et al. A cost-effective IoT system for monitoring indoor radon gas concentration. *Sensors (Basel)* 2018; 18: 2198.

2. Ding X, Tian Y and Yu Y. A real-time big data gathering algorithm based on indoor wireless sensor networks for risk analysis of industrial operations. *IEEE T Ind Inform* 2016; 12: 1232–1242.

3. Rashid B and Rehmani MH. Applications of wireless sensor networks for urban areas: a survey. *J Netw Comput Appl* 2016; 60: 192–219.

4. Fan X, Wei W, Wozniak M, et al. Low energy consumption and data redundancy approach of wireless sensor networks with big data. *Inf Technol Control* 2018; 47(3): 406–418.

5. Laurindo S, Moraes R, Nassiffe R, et al. An optimized relay selection technique to improve the communication reliability in wireless sensor networks. *Sensors (Basel)* 2018; 18: 3263.

6. Airehrour D, Gutierrez J and Ray SK. Secure routing for internet of things: a survey. *J Netw Comput Appl* 2016; 66: 198–213.

7. Mesodiakaki A, Zola E, Santos R, et al. Optimal user association, backhaul routing and switching off in 5G heterogeneous networks with mesh millimeter wave backhaul links. *Ad Hoc Netw* 2018; 78: 99–114.

8. Winnicka A, Kęsik K, Połap D, et al. A multi-agent gamification system for managing smart homes. Sensors (Basel) 2019; 19(5): 1249.

9. Lauraitis A, Maskeliunas R, Damasevicius R, et al. A smartphone application for automated decision support in cognitive task based evaluation of central nervous system motor disorders. IEEE J Biomed Health. Epub ahead of print 9 January 2019. DOI: 10.1109/JBHI.2019.2891729.

10. Marszałek Z, Woźniak M and Połap D. Fully flexible parallel merge sort for multicore architectures. *Complexity* 2018; 2018: 8679579.

11. Ahamed MM and Faruque S. 5G backhaul: requirements, challenges, and emerging technologies. In: *Proceedings of the broadband communications networks—recent advances and lessons from practice*, 5 November 2018. IntechOpen, https://www.intechopen.com/books/broadband-communications-networks-recent-advances-and-lessons-from-practice/5g-backhaul-requirements-challenges-and-emerging-technologies

12. Lee KC, Sudhaakar R, Ning J, et al. A comprehensive evaluation of RPL under mobility. *Int J Veh Technol* 2012; 2012: 904308.

13. Safdar V, Bashir F, Hamid Z, et al. A hybrid routing protocol for wireless sensor networks with mobile sinks. In: *Proceedings of IEEE international symposium on wireless and pervasive computing (ISWPC)*, Dalian, China, 3–5 July 2012, pp.1–5. New York: IEEE.

14. Korbi IEl, Brahim MB, Adjih C, et al. Mobility enhanced RPL for wireless sensor networks. In: *Proceedings of the IEEE network of the future (NOF)*, Gammarth, Tunisia, 21–23 November 2012, pp.1–8. New York: IEEE.

15. Hong KS and Choi L. DAG-based multipath routing for mobile sensor networks. In: *Proceedings of the IEEE international conference on information and communication technology convergence (ICTC)*, Seoul, South Korea, 28–30 September 2011, pp.261–266. New York: IEEE.

16. Fotouhi H, Moreira D and Alves M. mRPL: boosting mobility in the internet of things. *Ad Hoc Netw* 2015; 26: 17–35.

17. Barcelo M, Correa A, Vicario JL, et al. Addressing mobility in RPL with position assisted metrics. *IEEE Sens J* 2016; 16: 2151–2161.

18. Tahir Y, Yang S and McCann J. BRPL: backpressure RPL for high-throughput and mobile IoT. *IEEE T Mobile Comput* 2018; 17: 29–43.

19. Bouaziz M, Rachedi A and Belghith A. EKF-MRPL: advanced mobility support routing protocol for internet of mobile things: movement prediction approach. *Future Gener Comp Sy* 2019; 93: 822–832.

20. Kharrufa H, Al-Kashoash H, Al-Nidawi Y, et al. Dynamic RPL for multi-hop routing in IoT applications. In: *Proceedings of the annual conference on wireless on-demand network systems and services (WONS)*, Jackson, MI, 21–24 February 2017, pp.100–103. New York: IEEE.

21. Sanshi S and Jaidhar CD. Enhanced mobility aware routing protocol for low power and lossy networks. *Wirel Netw* 2019; 25: 1641–1655.

22. Wang J, Chalhoub G and Misson M. Mobility support enhancement for RPL. In: *Proceedings of international conference on performance evaluation and modeling in wired and wireless networks (PEMWN)*, Paris, 28–30 November 2017, pp.1–6. New York: IEEE.

23. Fotouhi H, Moreira D, Alves M, et al. mRPL + : a mobility management framework in RPL/6LoWPAN. *Comput Commun* 2017; 104: 34–54.

24. Carels D, Poorter ED, Moerman I, et al. RPL mobility support for point-to-point traffic flows towards mobile nodes. *Int J Distrib Sens N* 2015; 11: 1–13.

25. Gaddour O, Koubaa A, Rangarajan R, et al. Co-RPL: RPL routing for mobile low power wireless sensor networks using Corona mechanism. In: *Proceedings of the international symposium on industrial embedded systems (SIES)*, Pisa, 18–20 June 2014, pp 200–209. New York: IEEE.

26. Osterlind F, Dunkels A, Eriksson J, et al. Cross-level sensor network simulation with COOJA. In: *Proceedings of the 31st IEEE conference on local computer networks (LCN)*, Tampa, FL, 14–16 November 2006, pp.641–648. New York: IEEE.

27. Iova O, Picco P, Istomin T, et al. RPL: the routing standard for the internet of things … or is it? *IEEE Commun Mag* 2016; 54: 16–22.

28. Perkins C, Belding-Royer E and Das S. *Ad hoc on-demand distance vector (AODV) routing*. Report No. RFC3561, July 2003, https://tools.ietf.org/html/rfc3561

29. Marina MK and Das SR. On-demand multipath distance vector routing in ad hoc networks. In: *Proceedings of the 9th IEEE international conference on network protocols (ICNP)*, Riverside, CA, 11–14 November 2011, pp.14–23. New York: IEEE.

30. Dunkels A, Gronvall B and Voigt T. Contiki—a lightweight and flexible operating system for tiny networked sensors. In: *Proceedings of the 29th annual international conference on local computer networks (LCN)*, Tampa,

FL, 16–18 November 2004, pp.455–462. New York: IEEE.

31. Fotouhi H, Alves M, Zamalloa MZ, et al. Reliable and fast hand-offs in low-power wireless networks. *IEEE T Mobile Comput* 2014; 13: 2620–2633.

32. Kamgueu PO, Nataf E and Ndie TD. Survey on RPL enhancements: a focus on topology, security and mobility. *Comput Commun* 2018; 120: 10–21.

33. Park J, Kim K-H and Kim K. An algorithm for timely transmission of solicitation messages in RPL for energy-efficient node mobility. *Sensors (Basel)* 2017; 17: 899–910.

34. Lassouaoui L, Rodevakis S, Sailhan F, et al. Comparison of RPL routing metrics on grids. *Ad Hoc Netw* 2017; 184: 64–75.

35. Sanmartin P, Rojas A, Fernandez L, et al. Sigma routing metric for RPL protocol. *Sensors (Basel)* 2018; 18: 1277.

36. Cobarzan C, Montavont J and Noel T. Analysis and performance evaluation of RPL under mobility. In: *Proceedings of the IEEE symposium on computers and communication (ISCC)*, Funchal, Portugal, 23–26 June 2014, pp.1–6. New York: IEEE.

37. An B and Papavassiliou S. An entropy-based model for supporting and evaluating route stability in mobile ad hoc wireless networks. *IEEE Commun Lett* 2002; 6: 328–330.

38. Aschenbruck N, Ernst R, Gerhards-Padilla E, et al. BonnMotion: a mobility scenario generation and analysis tool. In: *Proceedings of the international conference on simulation tools and techniques (ICST)*, Malaga, 15–19 March 2010, pp.1–10. New York: IEEE.

39. Han G, Jiang J, Zhang C, et al. A survey on mobile anchor node assisted localization in wireless sensor networks. *IEEE Commun Surv Tut* 2016; 18: 2220–2243.

40. Rahman MN, Hanuranto MTIAT and Mayasari STMTR. Trilateration and iterative multilateration algorithm for localization schemes on wireless sensor network. In: *Proceedings of the international conference on control, electronics, renewable energy and communications (ICCREC)*, Yogyakarta, Indonesia, 26–28 September 2017, pp.88–92. New York: IEEE.

41. Winter T, Thubert P, Clausen T, et al. *RPL: IPv6 routing protocol for low-power and lossy networks*. Report No. RFC 6550, March 2012, https://tools.ietf.org/html/rfc6550

42. Halder S and Ghosal A. A survey on mobility-assisted localization techniques in wireless sensor networks. *J Netw Comput Appl* 2016; 60: 82–94.

43. Stanoev A, Filiposka S, In V, et al. Cooperative method for wireless sensor network localization. *Ad Hoc Netw* 2016; 40: 61–72.

44. Oh ES, Lee SH, Lee SH, et al. Comparison of SUN and Wi-Fi P2P WSN in M2M environments. *Int J Distrib Sens N* 2015; 11: 1–7.

45. Mayzaud A, Badonnel R and Chrisment I. A taxonomy of attacks in RPL-based internet of things. *Int J Netw Secur* 2016; 18(3): 459–473.

46. Wallgren L, Raza S and Voigt T. Routing attacks and countermeasures in the RPL-based internet of things. *Int J Distrib Sens N* 2013; 9: 794326.

47. Raza S, Wallgren L and Voigt T. SVELTE: real-time intrusion detection in the internet of things. *Ad Hoc Netw* 2013; 11(8): 2661–2674.

48. Perrey H, Landsmann M, Ugus O, et al. TRAIL: topology authentication in RPL. In: *Proceedings of the international conference on embedded wireless systems and networks (EWSN)*, Graz, 2016, pp.59–64. New York: ACM.

49. Zhang K, Liang X, Lu R, et al. Sybil attacks and their defenses in the internet of things. *IEEE Internet Things* 2014; 1: 372–383.

50. Mayzaud A, Sehgal A, Badonnel R, et al. Mitigation of topological inconsistency attacks in RPL-based low-power lossy networks. *Int J Netw Manag* 2015; 25(5): 320–339.

51. Hui J and Vasseur JP. *The routing protocol for low-power and lossy networks (RPL) option for carrying RPL information in data-plane datagrams*. Report No. RFC6553, March 2012, https://tools.ietf.org/html/rfc6553

52. Navidi W and Camp T. Stationary distributions for the random waypoint mobility model. *IEEE T Mobile Comput* 2004; 3: 99–108.