

Article

# Collision Avoidance Geographic P2P-RPL in Multi-Hop Indoor Wireless Networks

Yunyoung Choi , Jaehyung Park, Jiwon Jung  and Younggoo Kwon 

Department of Electrical and Electronics Engineering, Konkuk University, Seoul 05029, Korea; dsung426@naver.com (Y.C.); jason0124@naver.com (J.P.); metri@naver.com (J.J.)

\* Correspondence: ygkwon@konkuk.ac.kr

**Abstract:** In home and building automation applications, wireless sensor devices need to be connected via unreliable wireless links within a few hundred milliseconds. Routing protocols in Low-power and Lossy Networks (LLNs) need to support reliable data transmission with an energy-efficient manner and short routing convergence time. IETF standardized the Point-to-Point RPL (P2P-RPL) routing protocol, in which P2P-RPL propagates the route discovery messages over the whole network. This leads to significant routing control packet overhead and a large amount of energy consumption. P2P-RPL uses the trickle algorithm to control the transmission rate of routing control packets. The non-deterministic message suppression nature of the trickle algorithm may generate a sub-optimal routing path. The listen-only period of the trickle algorithm may lead to a long network convergence time. In this paper, we propose Collision Avoidance Geographic P2P-RPL, which achieves energy-efficient P2P data delivery with a fast routing request procedure. The proposed algorithm uses the location information to limit the network search space for the desired route discovery to a smaller location-constrained forwarding zone. The Collision Avoidance Geographic P2P-RPL also dynamically selects the listen-only period of the trickle timer algorithm based on the transmission priority related to geographic position information. The location information of each node is obtained from the Impulse-Response Ultra-WideBand (IR-UWB)-based cooperative multi-hop self localization algorithm. We implement Collision Avoidance Geographic P2P-RPL on Contiki OS, an open-source operating system for LLNs and the Internet of Things. The performance results show that the Collision Avoidance Geographic P2P-RPL reduced the routing control packet overheads, energy consumption, and network convergence time significantly. The cooperative multi-hop self localization algorithm improved the practical implementation characteristics of the P2P-RPL protocol in real world environments. The collision avoidance algorithm using the dynamic trickle timer increased the operation efficiency of the P2P-RPL under various wireless channel conditions with a location-constrained routing space.

**Keywords:** point-to-point routing; geographic routing; trickle algorithm; routing convergence time



check for updates

**Citation:** Choi, Y.; Park, J.; Jung, J.; Kwon, Y. Collision Avoidance Geographic P2P-RPL in Multi-Hop Indoor Wireless Networks. *Electronics* **2021**, *10*, 1484. <https://doi.org/10.3390/electronics10121484>

Academic Editor: Christos J. Bouras

Received: 21 May 2021

Accepted: 16 June 2021

Published: 20 June 2021

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

In Internet of Things (IoT) applications, IoT devices have unique addresses and establish ubiquitous connectivity with each other. IoT devices generally have limited CPU, memory, and power resources with low data rates, short communication ranges, and dynamically changing network topologies in LLNs. The IPv6 routing protocols in IoT applications should provide enhanced services and qualities by reducing the protocol complexity and speeding up routing request procedures by supporting seamless connectivity. The Routing Protocol for Low Power and Lossy Networks (RPL) is one of the most important routing protocols utilized in the IoT application network layer [1].

In home and building automation applications, point-to-point (P2P) communication is the prevalent traffic pattern [2,3]. Although the default RPL implementation supports P2P traffic, the P2P approach introduced by the standard RPL has high cost routing

overheads [4,5]. If two nodes want to communicate with each other, packets must be sent upward to the first common ancestor or the root node in the worst case and downward to the destination.

This strategy creates congestion around the root node and increases the route discovery latency significantly. IETF standardized P2P-RPL with a reactive multicast mechanism for quick and root-independent route discovery [6]. The use of P2P-RPL avoids the congested routing traffic around the root node and reduces the routing path length. P2P-RPL floods the entire network with control packets to discover routes, which results in a large amount of control overheads and huge energy consumptions.

A large number of collisions may occur particularly when different sources initiate P2P route discovery simultaneously, which can increase the latency of finding a new route from the source node to destination node. The trickle timer algorithm is one of the major components in an RPL system. The trickle algorithm aims to transmit the code and update it to every connected nodes in the network in a scalable, simple, and energy-efficient manner [7].

The non-deterministic message suppression performed by a trickle algorithm may generate suboptimal route formation when high message suppression is performed. Moreover, the trickle algorithm suffers from a slow convergence time due to the long listen-only period [8–11]. The inherited delay is accumulated at every hop, resulting in an increased overall delay proportional to the number of hops. These large routing overheads and long latency for P2P route discovery are not suitable for the routing requirements of IoT applications in LLNs.

In this paper, we propose the Collision Avoidance Geographic P2P-RPL to reduce the overheads in route discovery packet flooding procedure and the convergence route discovery latency. The location information is obtained from the cooperative multi-hop self localization algorithm in Impulse-Response Ultra-WideBand (IR-UWB) networks. The Collision Avoidance Geographic P2P-RPL exploits the location information to limit the search space for a desired route and to determine the transmission priority value related to the geographic position information. Reducing the network search space results in fewer control messages for P2P route discovery.

Depending on the priority value related to the geographic position, the listen-only period of the trickle algorithm is dynamically selected. The node with the higher priority value has the shorter listen-only period. The higher prioritized node with the shorter listen-only period transmits the message earlier than other nodes, which can reduce the route discovery latency significantly. In indoor location-based services, positional accuracy, privacy, low cost, low power consumption, reliability, continuity of service, and a short response time are required. The proposed Collision Avoidance Geographic P2P-RPL can provide fast, energy-efficient P2P data delivery without reducing the networking reliability.

The remainder of this paper is organized as follows. Section 2 discusses the related work for problems related to P2P-RPL and the trickle algorithm. Section 3 describes the main components of the proposed algorithm and software architecture on Contiki OS. Section 4 presents and discusses the performance evaluations. Section 5 summarizes the work and presents our concluding remarks.

## 2. Research Background

A large number of LLN applications, such as home automation and building control applications, require point-to-point (P2P) routing support. RPL has been defined by IETF as a standard routing protocol for LLNs [1]. RPL organizes a tree-like network topology called Destination-Oriented Directed Acyclic Graph (DODAG). When two non-root nodes need to communicate, the data packets must be routed only along the pre-established DODAGs. A DODAG is built to optimize the routing cost to reach the root node. Hence, such P2P routing may lead to a longer end-to-end delay, low packet delivery ratio, and severe traffic congestion near the root node [4,5]. Complementary to these shortcomings, a reactive discovery of Point-to-Point Routes in Low-Power and Lossy Networks (P2P-RPL) is standardized [6].

P2P-RPL reactively discovers P2P paths by constructing the temporary DODAG. The temporary DODAG is solely used for route discovery. P2P-RPL uses the same mechanism of the basic RPL to form the DODAG. The source node acts as the root node and it originates P2P mode DIO (P2P-DIO) for temporary DODAG construction. The P2P-DIO piggybacks a P2P Route Discovery Option to discover the route for a source-destination pair. Each node receiving the P2P-DIO joins the temporary DODAG and disseminates the P2P-DIO periodically with the trickle algorithm. Once the destination node receives the P2P-DIO, it replies P2P Discovery Reply Object (P2P-DRO) to the source node along the discovered route.

The discovered route is used for data packet transmission from the source node to destination node. RPL and P2P-RPL adopt trickle algorithm to control the transmission rate of the DIO control packet with the goal of a fast network convergence time and low energy consumption of the nodes in the network. The two conflicting goals are achieved by adaptive transmission periods and message suppression mechanism [7]. A node exponentially increases the transmission period when a network is in a stable state. To detect the state of a network, a node has a listen-only period before each transmission period.

When receiving an inconsistent message (e.g., a route advertised by the message is better than the existing route in terms of the routing metrics in use), a node decreases the transmission period to the minimum value to quickly resolve the inconsistency. When a node receives a consistent message and the number of consistent messages is more than predefined value  $k$ , the node suppresses a DIO transmission. However, the non-deterministic message suppression forces some nodes to remain silent for a long time, resulting in some routes remaining undiscovered [12].

This non-deterministic nature of the trickle algorithm can lead to sub-optimal route formation when high message suppression is performed (e.g.,  $k = 1$ ). Moreover, the listen-only period imposed by the trickle algorithm delays a transmission at every hop, resulting in an increase of the network convergence time [8]. P2P-RPL constantly broadcasts DIO control packets throughout the whole network using trickle algorithm. The network-wide broadcast approach could lead to serious medium contention known as the broadcast storm problem, especially in dense networks [13,14].

The Lightweight On-demand Ad hoc Distance-vector routing protocol-next generation (LOADng) uses a flooding mechanism to discover and maintain routes only on-demand. LOADng is a reactive routing protocol based on AODV and was extended for use in LLNs. When a new route is required, LOADng disseminates route discovery packets throughout the entire network. A node forwards route discovery packets until they reach the destination node according to the flooding operation. This classic flooding mechanism uses reduced relay sets [15] or the Trickle algorithm [16].

LOADng suffers from broadcast storms caused by flooding route discovery packets and does not include mechanisms to stop the spread of control packets after the target is found [17]. The LOADng protocol is suitable under few concurrent traffic flows in the network. Geographic routing makes use of the locations of the nodes and relies mainly on a simple greedy forwarding strategy. In a greedy forwarding scheme, a router simply forwards a data packet to the neighbor that is closest to the destination. The geographic routing approach is largely known by its low control packet overhead, low route stretch, and scalability.

However, this approach assumes sufficient network density, accurate localization, and significant link reliability. The existence of unreliable links leads to a drastic reduction of the packet delivery ratio or increased energy consumption if a retransmission mechanism is used [18]. Barriquello et al. presented a geographic approach for large-scale 6LoWPAN networks [19].

To better support P2P communication, they integrate a GOAFR (Greedy Other Adaptive Face Routing) algorithm [20] in the RPL protocol. The GOAFR algorithm is able to get closer to the optimum path with a high link density. On the other hand, in networks with low link density, RPL performs better and may find shorter paths than GOAFR. The loca-

tion inaccuracy causes routing failure. Chakraborty et al. combined RPL with geographic routing for P2P communication, where only nodes on the way to the destination node joined the route discovery procedure [21]. These geographic routing approaches, however, have the limitation that all nodes in the network must be static and need to have prior knowledge of the location.

The use of mobile nodes is possible only when they are one hop away from a static node. Zhao et al. designed the Energy-Efficient Region-based Routing Protocol for Low-Power and Lossy Networks (ER-RPL), which exploits the region features of static networks to suppress flooding overhead [13]. ER-RPL obtains the location of each node through a self-configuration localization scheme and is designed to limit the search space for P2P route discovery by using the position information of each node in the network. The approach aims to reduce the amount of control packet overhead and energy consumption without harming the network's reliability.

In ER-RPL, a network area is divided into a configurable number of non-overlapping regions, and each node belongs to one region. A distributed self-regioning algorithm based on a simple DV-hop algorithm is proposed for nodes to estimate the regions of themselves. Based on the estimated region information, only a portion of nodes join the route discovery procedure and forward route discovery packets. Xu et al. proposed energy-efficient region source routing (ER-SR), which exploits a distributed energy region algorithm [22].

Based on the residual energy of the nodes, the proposed routing scheme selects a part of nodes with high energy as the source routing nodes and only source routing nodes participating in source routing procedure. The other nodes join the region of source routing node and transmit the data packet through the source routing node. The accuracy of the estimated region information severely depends on the network configurations. If the nodes are distributed non-uniformly, DV-hop-based localization schemes provide inaccurate location information for each node, resulting in routing performance degradation.

To reduce the cost of P2P-RPL route discovery in terms of the control packet overhead, energy consumption, and fast network convergence time, we exploited the IR-UWB-based cooperative multi-hop positioning with smartphone INS tracking by particle filtering. The Collision Avoidance Geographic P2P-RPL algorithm is proposed with limiting the network search space for the desired route discovery based on the location information of the source and destination nodes. To decrease the delay in propagating P2P-DIO transmissions for route discovery, the listen-only period of the trickle algorithm is selected dynamically based on the physical location information of each node.

### 3. Collision Avoidance Geographic Point-to-Point RPL

In this section, the Collision Avoidance Geographic P2P-RPL is proposed to discover the on-demand P2P route based on the location information from IR-UWB-based cooperative self localization. Collision Avoidance Geographic P2P-RPL was operated in RPL-based networks to find a P2P path with location information of each node. All nodes were UWB-enabled RPL routers and each node was able to estimate its location information by the fully-distributed Bayesian cooperative multi-hop positioning method. A node exchanged the statistical characteristics of the location error with neighbors by piggybacking a location error field on the standard RPL control messages (DIO and DAO).

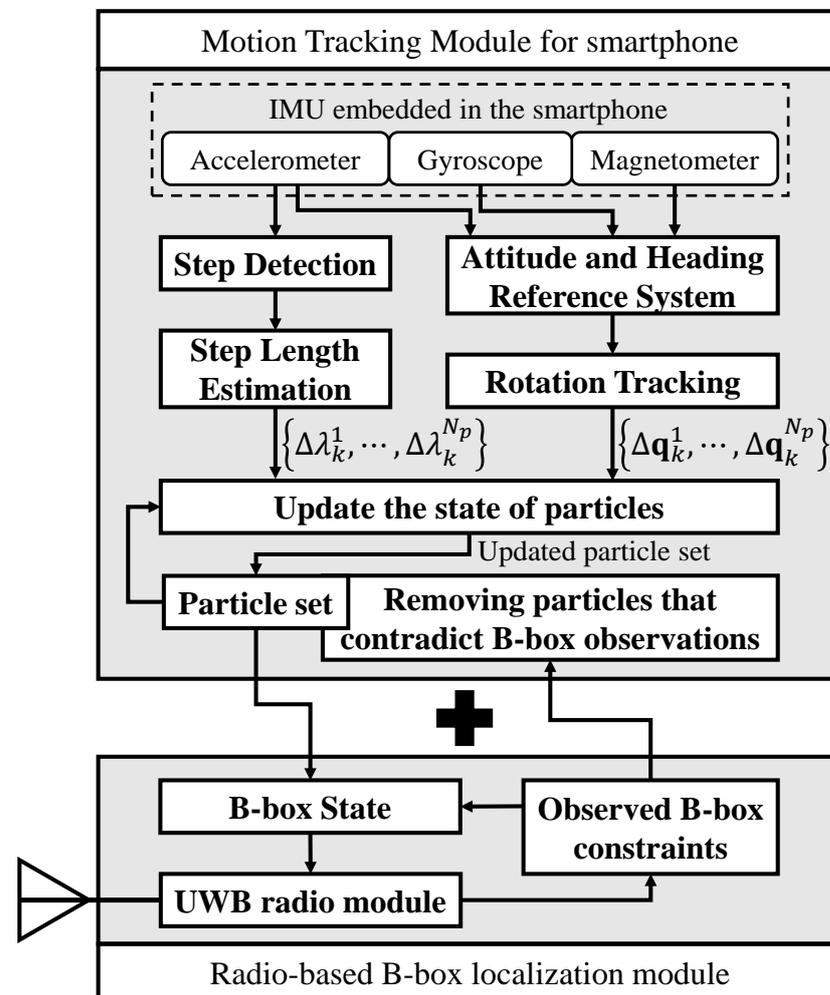
Location error characteristics are represented by the location state, which includes the two-dimensional position information of the lower and upper bounds of the position estimates respectively. For P2P route discovery, the location-constrained forwarding zone is defined as the smallest rectangular region including the estimated region of the source and destination nodes. A node joins the temporary DAG rooted at the source and forwards a P2P-DIO only if it belongs to the location-constrained forwarding zone. Only the nodes within the location-constrained forwarding zone can participate in the route discovery procedure, resulting in reduced routing overhead of the P2P-RPL algorithm.

To mitigate the long listen-only period and non-deterministic suppression problem, the collision avoidance elastic trickle timer algorithm is proposed. Each node is given

a different length of listen-only period depending on the transmission priority. The order of the transmission priority is determined by the distance from the source node, and the angle between the line connecting the sender and the destination nodes and the line connecting the sender and the target nodes. If a higher order of transmission priority is given, then a shorter listen-only period is selected in the trickle timer. The shorter listen-only period allows a node to transmit a P2P-DIO earlier than the other nodes and the probability of collisions becomes significantly lower.

### 3.1. IR-UWB-Based Cooperative Multi-Hop Self Localization in LLNs

Figure 1 describes the proposed IR-UWB-based cooperative multi-hop self-localization system. Each node is location-aware by exchanging the statistical properties of the location error with its neighbors. The UWB radio module provides distance information between the nodes. In a two-dimensional network scenario, each node perceives a rectangular area in which it exists. Despite the short communication range of UWB technology, the proposed algorithm only needs a small number of anchor nodes with known locations.



**Figure 1.** Impulse-Response Ultra-WideBand (IR-UWB)-based cooperative multi-hop self-localization system.

The proposed localization module using UWB radio is based on the Bounding-box (B-box) algorithm [23]. The B-box state represents the rectangular region where each node is located, and the B-box localization module informs the neighboring nodes of its B-box state. For smartphones with UWB transceivers and inertial sensors, a motion tracking

module was added. Pedestrian movement was tracked through particle filtering, and the updated state of the particle set was converted to a B-box state.

The B-box state  $\mathbf{B}_T$ , representing the estimated location region of a node  $T$ , is defined as Equation (1).

$$\mathbf{B}_T = [x_T^{lb} \quad x_T^{ub}] \times [y_T^{lb} \quad y_T^{ub}] \quad (1)$$

The superscripts  $lb$  and  $ub$  denote the lower and upper bounds of the rectangular area. The actual coordinates of node  $T$ ,  $(x_T^a, y_T^a)$ , satisfy  $x_T^{lb} \leq x_T^a \leq x_T^{ub}$  and  $y_T^{lb} \leq y_T^a \leq y_T^{ub}$ . For an anchor node  $A$  with known position coordinates  $(x_A^a, y_A^a)$ , the B-box state  $\mathbf{B}_A$  is determined as

$$\mathbf{B}_A = [x_A^a \quad x_A^a] \times [y_A^a \quad y_A^a] \quad (2)$$

A non-anchor node updates its own B-box state using the B-box information of one-hop neighbor nodes. With the measured distance  $d_N$  to node  $N$  and the B-box state  $\mathbf{B}_N$ , a non-anchor node  $C$  updates its B-box state  $\mathbf{B}_C$  as follows [24]:

$$\begin{aligned} \mathbf{B}_C \leftarrow & \left[ \max(x_C^{lb}, x_N^{lb} - (d_N + \delta)) \quad \min(x_C^{ub}, x_N^{ub} + (d_N + \delta)) \right] \\ & \times \left[ \max(y_C^{lb}, y_N^{lb} - (d_N + \delta)) \quad \min(y_C^{ub}, y_N^{ub} + (d_N + \delta)) \right] \end{aligned} \quad (3)$$

The margin value  $\delta$  prevents the B-box state of a node from becoming invalid. According to the worst negative ranging error of the Decawave DW1000 module [25–27], the experiments in this paper set  $\delta$  to 60 cm. As the B-box states of non-anchor nodes become more specific, the network-wide localization accuracy improves over time.

Smartphones with inertial sensors and UWB transceivers accurately track their location through the particle filter [28]. For mobile nodes, the motion tracking module with the Inertial Navigation System (INS) is added to the radio-based B-box localization module. Translational and rotational movements of a pedestrian are tracked through rotation tracking, step detection, and step length estimation procedures. For particle filtering, each mobile node keeps one set of weighted particles. All  $N_p$  particles each have a state  $\mathbf{s}_k$  and a weight  $\omega_k$  at the  $k$ th iteration. The weighted particle set  $\zeta_k$  is defined as:

$$\zeta_k \equiv \left\{ \langle \mathbf{s}_k^l, \omega_k^l \rangle \mid l \in \{1, \dots, N_p\} \right\} \quad (4)$$

where

$$\mathbf{s}_k^l \equiv [\mathbf{r}_k^l \quad \mathbf{q}_k^l]^T = [x_k^l \quad y_k^l \quad q_{0,k}^l \quad q_{1,k}^l \quad q_{2,k}^l \quad q_{3,k}^l]^T \quad (5)$$

The state vector of each particle contains the two-dimensional location coordinates  $\mathbf{r}_k$  and the orientation quaternion  $\mathbf{q}_k$ . When the B-box state of a smartphone is initialized by the B-box localization module, the particles are sampled uniformly within the B-box region and the entire quaternion state space. The posterior probability of a node position at  $k$ th iteration is described as:

$$p(\mathbf{s}_k) \approx \sum_{l=1}^{N_p} \omega_k^l \delta(\mathbf{s}_k - \mathbf{s}_k^l) \quad (6)$$

where  $\delta(\cdot)$  is the Dirac delta function.

The Attitude and Heading Reference System (AHRS) continuously tracks the three-dimensional rotation of the smartphone [29]. The update of the particle state is triggered by the detection of step events [30]. When a step event is detected, the moving distance of the step and the rotation between the recent two steps are derived from the IMU output sequence.

The particle filter updates the state of the particle set by taking  $N_p$  samples from the approximate posterior distribution of the measured step length  $\Delta\lambda_k$  and rotation  $\Delta\mathbf{q}_k$  [31,32]. The

samples are labeled with superscripts as  $\{\langle \Delta\lambda_k^1, \Delta\mathbf{q}_k^1 \rangle, \dots, \langle \Delta\lambda_k^{N_p}, \Delta\mathbf{q}_k^{N_p} \rangle\}$ . By the sampled rotation estimates, the quaternion state of each particle is updated as [33]:

$$\mathbf{q}_{k+1}^l = \Delta\mathbf{q}_k^l \otimes \mathbf{q}_k^l \quad (7)$$

where

$$\mathbf{q} \otimes \mathbf{q}' = \begin{bmatrix} q_0 & -q_1 & -q_2 & -q_3 \\ q_1 & q_0 & -q_3 & q_2 \\ q_2 & q_3 & q_0 & -q_1 \\ q_3 & -q_2 & q_1 & q_0 \end{bmatrix} \begin{bmatrix} q'_0 \\ q'_1 \\ q'_2 \\ q'_3 \end{bmatrix}. \quad (8)$$

To calculate the 2D translational movement of the pedestrian, the 3D attitude state is projected in the 2D heading direction  $\psi$  as follows:

$$\psi_{k+1}^l = \arctan \frac{2(q_{0,k+1}^l q_{3,k+1}^l + q_{1,k+1}^l q_{2,k+1}^l)}{1 - 2\left\{ (q_{2,k+1}^l)^2 + (q_{3,k+1}^l)^2 \right\}}. \quad (9)$$

Finally, the location state of each particle is updated according to the two-dimensional pedestrian motion model.

$$\mathbf{r}_{k+1}^l = \mathbf{r}_k^l + \Delta\lambda_k^l \begin{bmatrix} \cos(\psi_{k+1}^l) \\ \sin(\psi_{k+1}^l) \end{bmatrix}. \quad (10)$$

When the B-box localization module observes the B-box constraints of neighboring nodes, particles with a state contrary to the radio observation are removed, and the total weights of the particles are normalized. To prevent the situation where the weight of a small number of particles dominate the rest of the particles [34], it uses a deterministic resampling method that avoids uncensored discarding of low-weighted particles [35]. The B-box localization module of a smartphone uses the smallest rectangle containing all surviving particles as the B-box state. Since smartphones move the network deployment space, spatial diversity leads to improvement in the overall localization performance of the network.

### 3.2. Location-Aided P2P-RPL with Location-Constrained Forwarding Zone

From the result of cooperative self localization, all nodes have up-to-date position states. This position information is piggybacked on the standard RPL control messages (e.g., DIO and DAO). A location-constrained forwarding zone is defined to limit the propagation area for route discovery messages. A node within the location-constrained forwarding zone joins the temporary DAG formation procedure and forwards a route discovery packet (i.e., P2P-DIO) to its neighbors. This results in a great reduction of the routing overhead and energy consumption, especially when the distance between the source and destination nodes is short [27].

We consider when a source node (say, node S) tries to find an optimal P2P path to a destination node (say, node D) as shown in Figure 2. Node S measures the path cost reaching node D by sending a measurement request along the pre-established global DAG. The measurement request carries the accumulated path cost (ETX). On receiving the measurement request, an intermediate node updates the path cost and forwards it to the next hop.

When node D receives the measurement request, node D sends a Measurement Reply back through the reverse path to node S. The measurement reply carries the cost of the route and the B-box state of node D with the location option. Upon receiving the measurement reply, and if the accumulated path cost does not meet the specified routing metric constraints, node S initiates the P2P route discovery.

As shown in Figure 3a, node S defines the location-constrained forwarding zone for P2P route discovery as the rectangular region that includes the current estimated location

states of the node S and node D. The four corners of the forwarding zone are computed as follows:

$$\begin{aligned} \mathbf{B}_{fwd} &= \begin{bmatrix} x_{fwd}^{lb} & x_{fwd}^{ub} \\ y_{fwd}^{lb} & y_{fwd}^{ub} \end{bmatrix} \\ &= \begin{bmatrix} \min(x_S^{lb}, x_D^{lb}) & \max(x_S^{ub}, x_D^{ub}) \\ \min(y_S^{lb}, y_D^{lb}) & \max(y_S^{ub}, y_D^{ub}) \end{bmatrix} \end{aligned} \quad (11)$$

where  $lb, ub, S$ , and  $D$  denote the lower bound, upper bound, source, and destination, respectively.

Node S then piggybacks the determined location-constrained forwarding zone  $\mathbf{B}_{fwd}$  on a P2P-DIO message and disseminates the P2P-DIO message to one-hop neighbor nodes. Upon receiving P2P-DIO, an intermediate node I joins P2P route discovery only if it belongs to the location-constrained forwarding zone (i.e.,  $\mathbf{B}_I \cap \mathbf{B}_{fwd} \neq \emptyset$ ). If there is an intersection area between the estimated region of node itself and the location-constrained forwarding zone, node I decides that it is inside the location-constrained forwarding zone and forwards a P2P-DIO. If not, node I simply discards P2P-DIO.

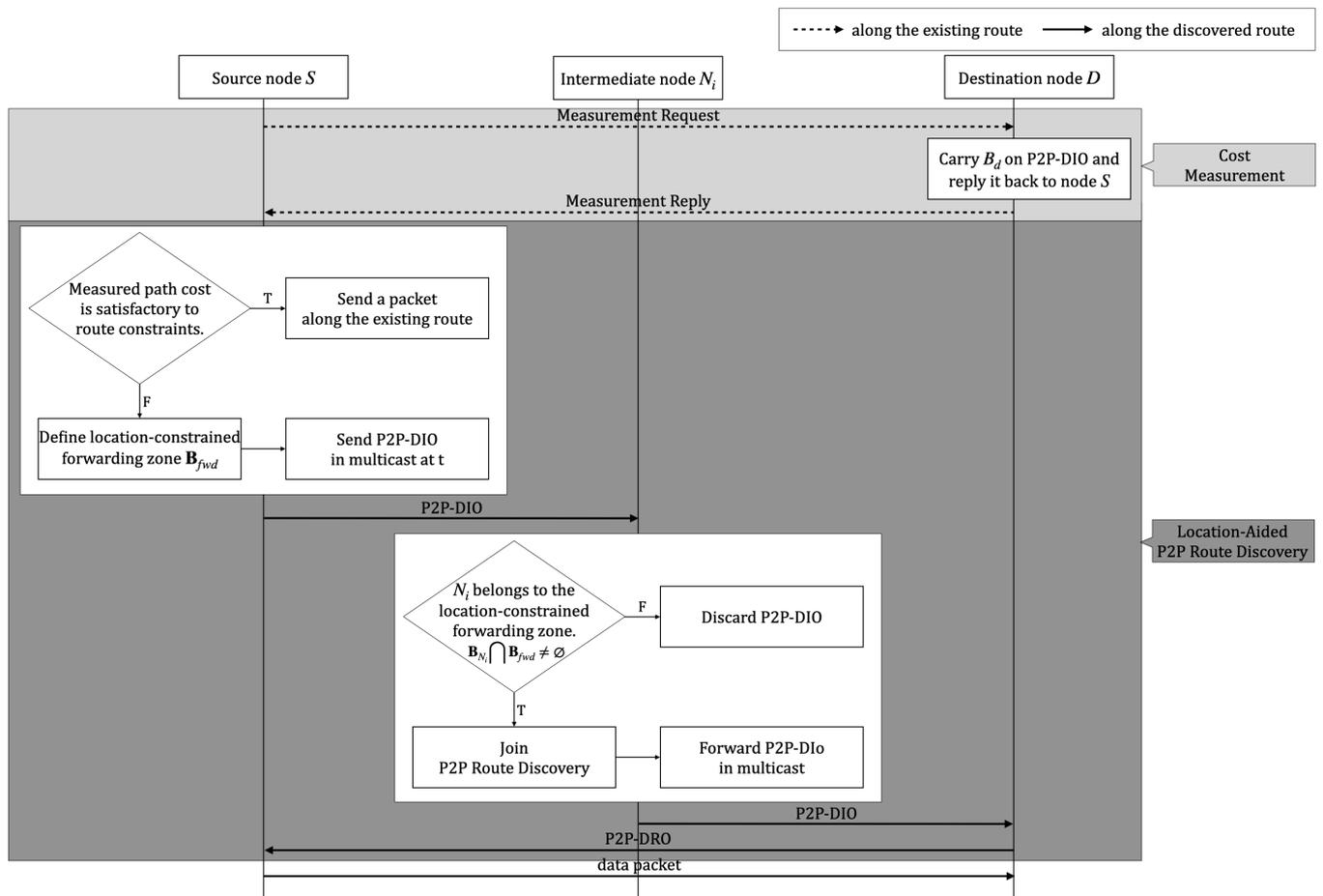
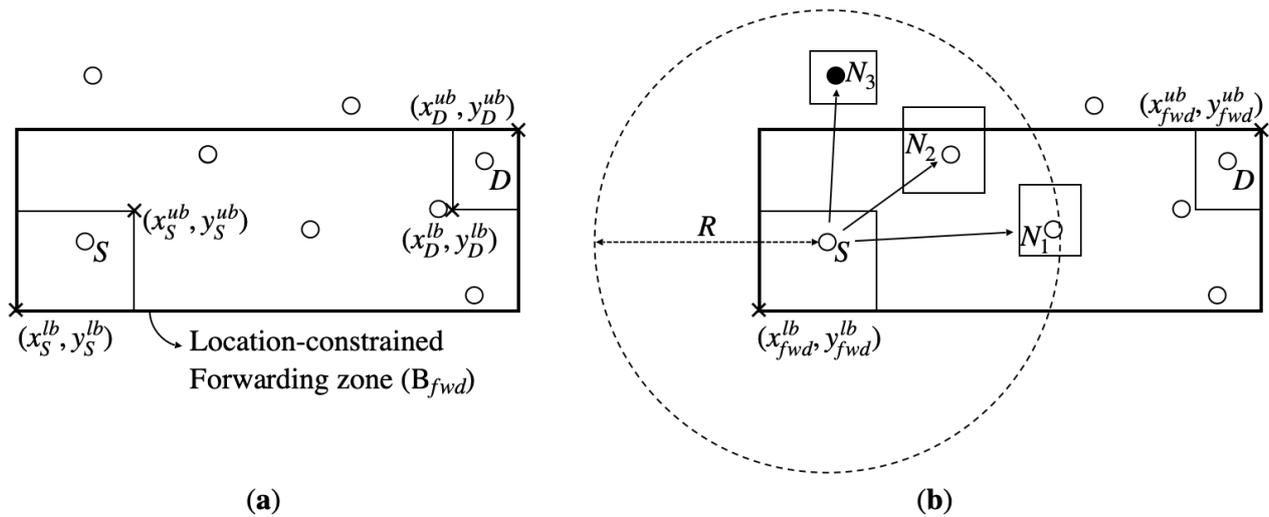


Figure 2. The flowchart of the proposed algorithm.



**Figure 3.** The process of (a) Determining the location-constrained forwarding zone and (b) location-aided P2P route discovery.

Figure 3b shows that nodes  $N_1$  and  $N_2$  satisfy the condition and participate in the route discovery procedure. On the other hand, node  $N_3$  discards the P2P-DIO as  $\mathbf{B}_{N_3} \cap \mathbf{B}_{fwd} = \emptyset$ . Once the node D receives P2P-DIO, the P2P path is established as the upward route to the temporary root (node S). Node D sends P2P-DRO, which contains the discovered route and stop flag set by 1 back to node S so that no more P2P-DIO messages are forwarded for this temporary DAG. Upon receiving P2P-DRO, intermediate nodes stop sending P2P-DIO periodically and send P2P-DRO to their neighbors. When P2P-DRO reaches node S, node S then sends a data packet along the discovered route.

In this way, only a part of the nodes participate in the route discovery procedure, and hence network-wide flooding rarely occurs. The size of the location-constrained forwarding zone is determined depending on the distance between the source and destination nodes. A shorter distance between node S and node D results in a smaller size of forwarding zone, and a fewer number of nodes join the P2P route discovery procedure. This results in a smaller number of route request packets (i.e., P2P-DIOs) being disseminated.

### 3.3. Collision Avoidance Geographic P2P-RPL with Elastic Trickle Algorithm

In standard P2P-RPL, the recommended value of the redundancy constant  $k$  is 1 to reduce the number of messages generated during a route discovery procedure [6]. This setting reduces the energy consumption of each node but increases the time required for the route discovery procedure to complete. Especially when high message suppression is performed (e.g.,  $k = 1$ ), the non-deterministic nature of trickle leads to suboptimal route formation.

Low minimum interval size  $I_{min}$  values result in the creation of the first DODAG within a short period of time; however, a long listen-only period problem inherently increases the DODAG formation time. In order to mitigate these problems, we propose the collision avoidance elastic trickle timer algorithm for P2P route discovery. To make the network convergence time faster, the relay nodes send a P2P-DIO faster than other nodes.

The collision avoidance elastic trickle timer algorithm allows a node to select the listen-only period dynamically depending on the priority value related to the geographic position of each node. Given a higher transmission priority value, the listen-only period will become shorter. The priority value is determined by the distance from the sender node, and the angle difference between the connecting line of sender-target nodes and the connecting line of sender-destination nodes. The farther distance and the smaller angle difference assigns the higher transmission priority value.

Each node has own interval  $I$ , and this interval is halved into two subsequent time intervals: the listening interval (i.e., listen-only period) and the transmitting interval, respectively. The trickle algorithm begins with  $I_{min}$  as the first interval. At the start of new

interval, a DIO message transmission is scheduled at a time  $t$  in the range of  $[I/2, I]$  and the redundancy constant  $c$  is reset to 0. A node only listens and receives messages from its neighbors until time  $t$ . During the listening interval, a node keeps track of the received DIO messages by increasing  $c$  each time a consistent DIO message is received. At time  $t$ , if  $c$  is lower than  $k$ , the node transmits DIO. If not, it suppresses DIO message transmission. At the end of the interval, the next interval size is doubled with respect to current one until the interval reaches  $I_{max}$ . The procedure is summarized as follows.

1. The transmission priority constant  $s$  is added, and it is used when dynamically selecting the listen-only period. The priority constant  $s$  is initialized every time a new interval starts.
2. The specific time  $t$  is chosen randomly in  $[I/2^{s+1}, I]$ .
3. When a node  $N_i$  receives DIO transmission from sender node  $A$ , node  $N_i$  calculates  $d_{AN_i}$  and  $\angle AN_iD$  using the location information piggybacked in the received message, where  $d_{AN_i}$  denotes the distance between node  $A$  and node  $N_i$ ,  $\angle AN_iD$  denotes the angle difference from node  $N_i$  to the line connecting node  $A$  and node  $D$ .
4. The node counts the transmission priority constant  $s$  when the calculated distance from the sender is greater than the predefined threshold value  $D_{TH}$  or when the calculated angle difference to sender–destination line is greater than the predefined threshold value  $\theta_{TH}$ .

A pseudo-code of collision avoidance elastic trickle timer algorithm is presented in Algorithm 1. Algorithm 1 describes how a relay node selects the length of listen-only period according to the locations of the sender and destination nodes. The “+” and “−” signs in Algorithm 1 indicate the modifications with respect to the standard, which has been added and removed respectively [1].

Figure 4 shows an example of a collision avoidance elastic trickle operation with four nodes,  $k = 1$ . A node in the black area has the first transmission priority. A node in the gray area has a second transmission priority. As  $d_{AN_3} \geq D_{TH}$  and  $\angle AN_3D \leq \theta_{TH}$ , node  $N_3$  has the first transmission priority, with the value equal to 2. Node  $N_1$  only satisfies  $d_{AN_1} \geq D_{TH}$  and, thus, has the second transmission priority. In order to achieve this, the listen-only period is dynamically selected. The listening period is set to a variable length that is inversely proportional to the transmission priority. The node with the higher transmission priority has a shorter listen-only period. In the case where a node has the first transmission priority, the length of the listen-only period is equal to  $I/8$ .

Figure 5 shows the operations when node S attempts to find the P2P path to node D. The bold arrow indicates the shortest route, and the dotted arrow indicates an alternate route discovered by P2P-RPL. In (a) and (b), all nodes except  $N_3$  joined the P2P route discovery. The value of redundancy constant  $k$  was equal to 1. In Figure 5a, the length of the listen-only period was selected in an elastic manner. The value of  $s$  of  $N_1$  and  $N_2$  were equal to 2 and 1, respectively, and hence the lengths of the listen-only period of  $N_1$  and  $N_2$  were equal to  $I/8$  and  $I/4$ , respectively. Node  $N_1$  had a higher probability to send a P2P-DIO in the current interval than that of node  $N_2$ . This resulted in a considerably decreased latency for the route discovery procedure. Moreover, decreasing the listen-only period allows the node to contend in a wider interval size, lowering the probability of collisions.

**Algorithm 1:** Collision Avoidance Elastic Trickle Algorithm

---

```

Function Initialization():
  |  $I \leftarrow I_{min}$ 
  |  $+ s \leftarrow 0$ 

Function StartNewInterval():
  |  $- t \leftarrow random\left(\frac{I}{2}, I\right)$ 
  |  $+ t \leftarrow random\left(\frac{I}{2^{s+1}}, I\right)$ 
  |  $c \leftarrow 0$ 
  |  $+ s \leftarrow 0$ 

Function ReceivedTransmission():
  | if  $d_{AN_i} \geq D_{TH}$  then
  | |  $+ s \leftarrow s + 1$ 
  | end
  | if  $\angle AN_i D \leq \theta_{TH}$  then
  | |  $+ s \leftarrow s + 1$ 
  | end
  | if ReceivedConsistentTransmission then
  | |  $c \leftarrow c + 1$ 
  | else if ReceivedInconsistentTransmission then
  | |  $I \leftarrow I_{min}$ 
  | | StartNewInterval()
  | end

Function TimerExpires():
  | if  $c < k$  then
  | | TransmitDIO
  | else
  | | SuppressDIO
  | end

Function IntervalEnds():
  |  $I \leftarrow I \times 2$ 
  | if  $I > I_{max}$  then
  | |  $I \leftarrow I_{max}$ 
  | end

```

---

As shown in Figure 5b,c, when the standard trickle timer was used and the length of the listen-only period of two nodes were the same, the probabilities of the P2P-DIO sent in the current interval of  $N_1$  and  $N_2$  were the same. If  $N_2$  sends a P2P-DIO, a node  $N_1$  keeps silent in the current interval as the redundancy message counter  $c$  is greater than the redundancy constant  $k$ . As node S can reach node D only through  $N_1$  inside the location-constrained forwarding zone, the latency to reach node D increases as much as the interval length  $N_1$  keeps silent.

However, as shown in Figure 5c, if all nodes in the network participate in P2P route discovery in P2P-RPL, node S can reach node D through an alternate relay node  $N_2$ . Even though the path length of this route (i.e.,  $S - N_2 - N_3 - D$ ) is longer than the shortest route (i.e.,  $S - N_1 - D$ ), the minimum interval length ( $I_{min}$ ) is much longer than the transmission delay, and thus node S reaches node D faster.

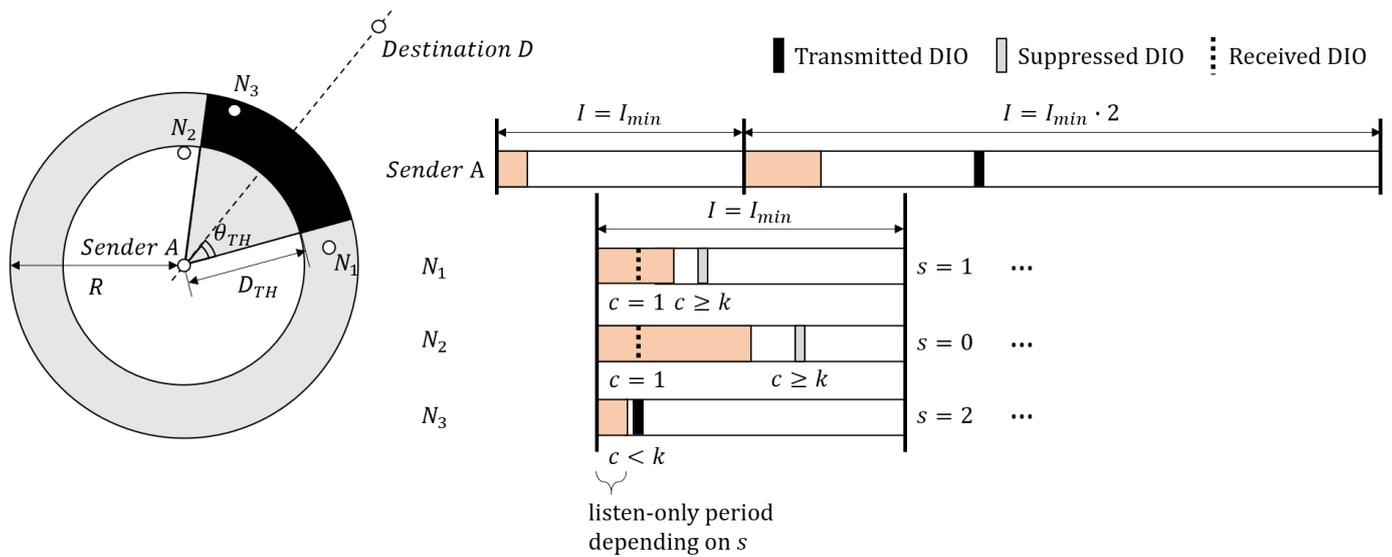


Figure 4. Example of a Collision Avoidance Elastic Trickle operation,  $k = 1$ .

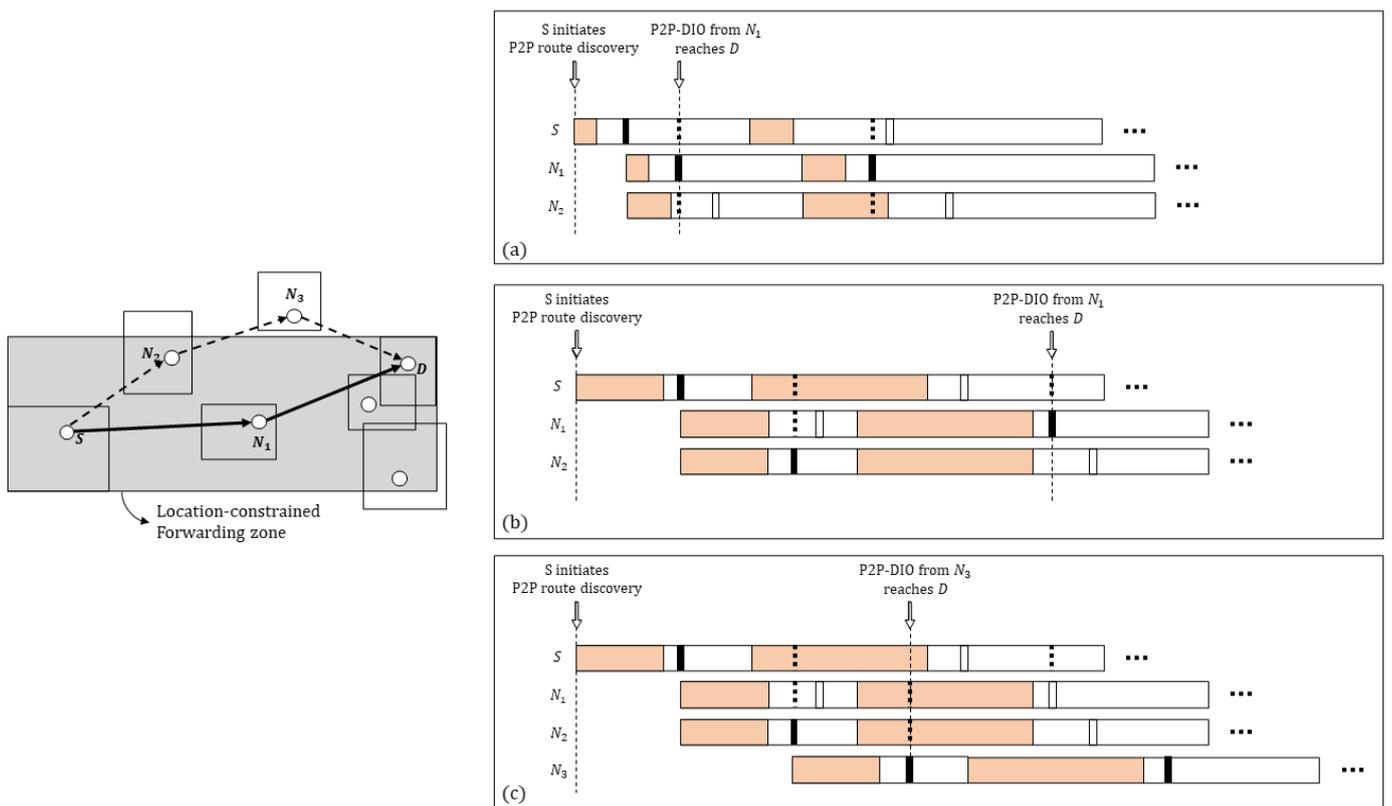


Figure 5. Trickle operations of (a) Collision Avoidance Geographic P2P-RPL, (b) P2P route discovery with only the location-constrained forwarding zone and (c) P2P-RPL,  $k = 1$ .

### 3.4. Software Architecture

We implemented the Collision Avoidance Geographic P2P-RPL in Contiki OS [36]—an open source operating system for LLNs and IoT. We made use of the ContikiRPL implementation [37] and P2P-RPL implementation [38].

The software architecture of Collision Avoidance Geographic P2P-RPL is illustrated in Figure 6. The arrow in Figure 6 depicts the P2P-DIO packet flow.

The key components are as follows:

- RPL ICMP packet management communicates with ICMPv6 to send/receive RPL control messages including MO, P2P-DIO, and P2P-DRO. When the ICMPv6 message is received, the RPL ICMP packet management component first checks the type of message and extracts the payload. Then, it notifies RPL DAG management component. Cost measurement is implemented inside this component.
- RPL DAG management contains all the logic related to DAGs with the IPv6 routing table management and the preferred parent determination based on the rank result of the RPL objective function component.
- The RPL Objective Function is the objective function that calculates the routing metric to select the parents. In ContikiRPL, OF0 [39] and MRHOF [40] are implemented.
- Cooperative Self Localization includes the B-box algorithm with utilizing UWB-based ranging measurements and particle filter-based inertial navigation. It receives the location states of one-hop neighbors, which are extracted by the RPL ICMP packet management component. After estimating the current location state of a node, it forwards the result to the location-constrained forwarding zone component.
- The location constrained forwarding zone is responsible for deciding whether to join the temporary DAG formation advertised by the incoming RPL ICMP packet (i.e., P2P-DIO). Given the location information extracted from the incoming packet, this component computes the four corners of the location-constrained forwarding zone  $B_{fwd}$ . It drops the packet if there is no intersection between its estimated region and the forwarding zone.
- The Collision Avoidance Elastic Trickle Timer holds the calculation of the distance and angle difference and the dynamic selection of the listen-only period based on the transmission priority constant  $s$  in addition to standard trickle-related logic.

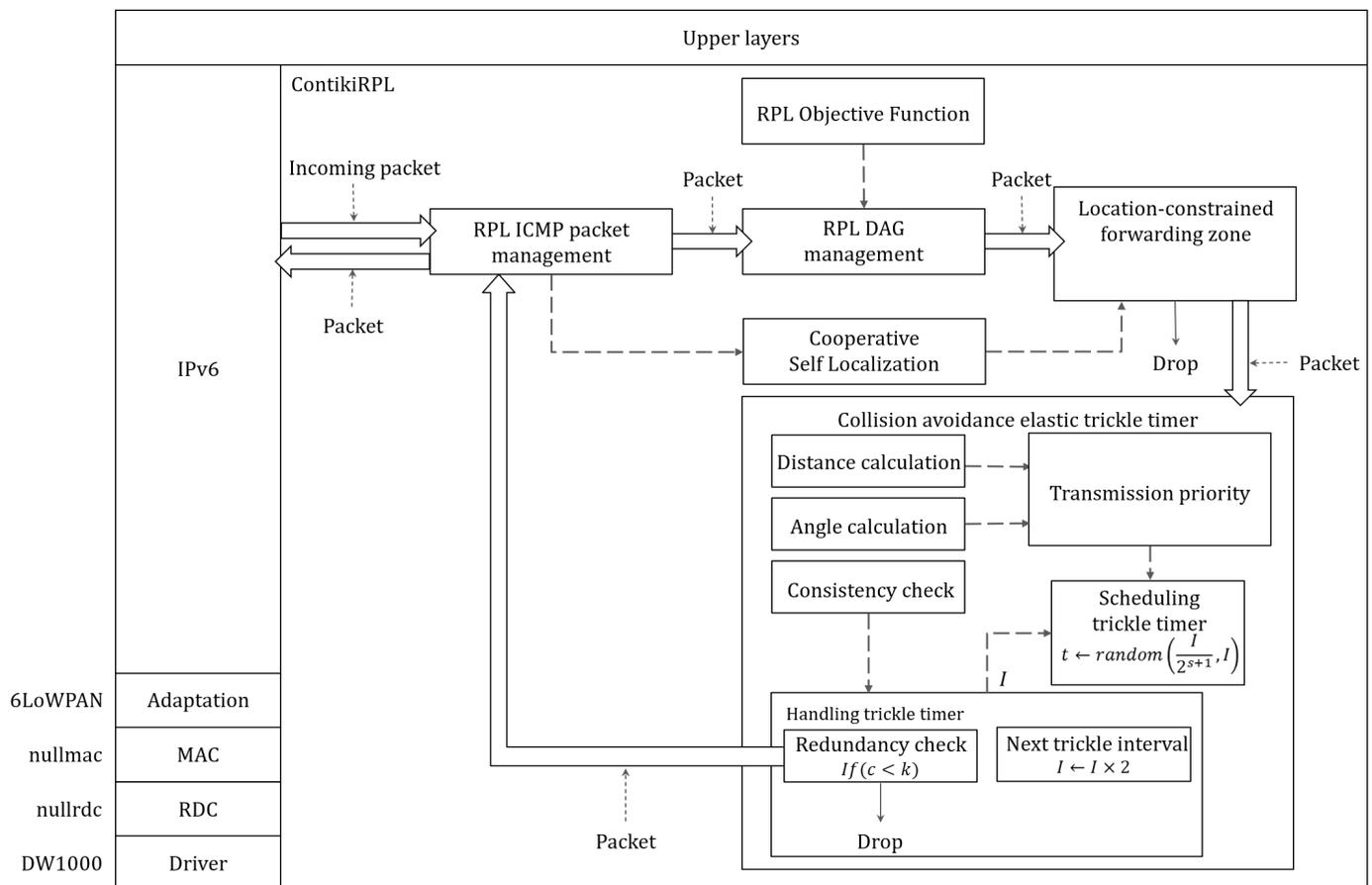


Figure 6. The software architecture of Collision Avoidance Geographic P2P-RPL on Contiki OS.

#### 4. Performance Evaluation

In this section, we evaluate the performance of P2P-RPL, ER-RPL, Location-Aided P2P-RPL, and Collision Avoidance Geographic P2P-RPL using the Cooja network simulator for Contiki Operating System (OS) [41]. The proposed algorithms, Location-Aided P2P-RPL and Collision Avoidance Geographic P2P-RPL, use a location-constrained forwarding zone for a P2P-RPL-based route discovery procedure. Collision Avoidance Geographic P2P-RPL additionally uses the elastic trickle timer algorithm during the route discovery procedure with the location-constrained forwarding zone. The algorithms are implemented based on modules of ContikiRPL library and P2P-RPL implementation [37,38].

Preliminary testbed experiments with the EVB1000 platform were performed for the IR-UWB-based Cooja network simulation parameter settings. We used the default Contiki UDP on top of the nullrdc layer and the Decawave low-level radio driver with radio adaptation codes [42]. The main elements integrated on the EVB1000 were the following: a STM32F105 ARM Cortex M3 microcontroller (with 64 kb RAM memory and 256 kb Flash memory), a DW1000 transceiver (based on standard 802.15.4-2011), and an omnidirectional antenna. The UWB radio configurations are stated in Table 1.

The end-to-end delay measurements were evaluated using Contiki's RTIMER with a resolution of 30.5  $\mu$ s. We changed the hop length between the sender and the destination node from 1 to 4 to assess the performance with various path lengths. The inter-message interval (IMI) was set to 1 s. The average one-hop delay is presented in Table 1.

**Table 1.** The UWB radio configuration parameters.

Parameter	Value
Channel number	5
Channel center frequency	6489.6 MHz
Channel bandwidth	499.2 MHz
Nominal bitrate	6.81 Mbps
Mean PRF	15.6 MHz
Preamble length	128 symbols
Transmitter electronics ( $E_{TX-elec}$ )	33.97 nJ/bit
Transmitter amplifier ( $\epsilon_{amp}$ )	6 pJ/bit/m <sup>2</sup>
Receiver electronics ( $E_{RX-elec}$ )	14.56 nJ/bit
Average one-hop delay	259.25 $\mu$ s

Based on the testbed experiment results, simulation studies were conducted to study the localization and routing performances in multi-hop LLNs. In network configurations, the static nodes and the mobile nodes were deployed in a  $75 \times 75$  m<sup>2</sup> network area. The number of static nodes varied from 25, 36, to 49 and the number of mobile nodes was set to 5. The static nodes were placed in grid deployment with some randomness: the static node's position was  $(10a \pm \Delta x, 10b \pm \Delta y)$ , where  $-10 \leq \Delta x \leq 10$ ,  $-10 \leq \Delta y \leq 10$ . Assuming that node ID is  $U$ , the value of  $a$  and  $b$  were obtained through  $U = \sqrt{N} \times a + b$ , where  $N$  is the number of static nodes.

Five mobile nodes were moving continuously with slow random walking speed and operated as leaf nodes without intermediate route forwarding features in the route discovery procedures [43,44]. The four outermost static nodes were anchor nodes with prior known positions. In this simulation study, we defined the concept of the average node degree for the performance evaluation in terms of the node density. The average node degree was defined as the average number of one-hop neighbors of a node in the network. The higher the average node degree was, the higher the node density was. The average node degree varies from 6, 9, to 12.

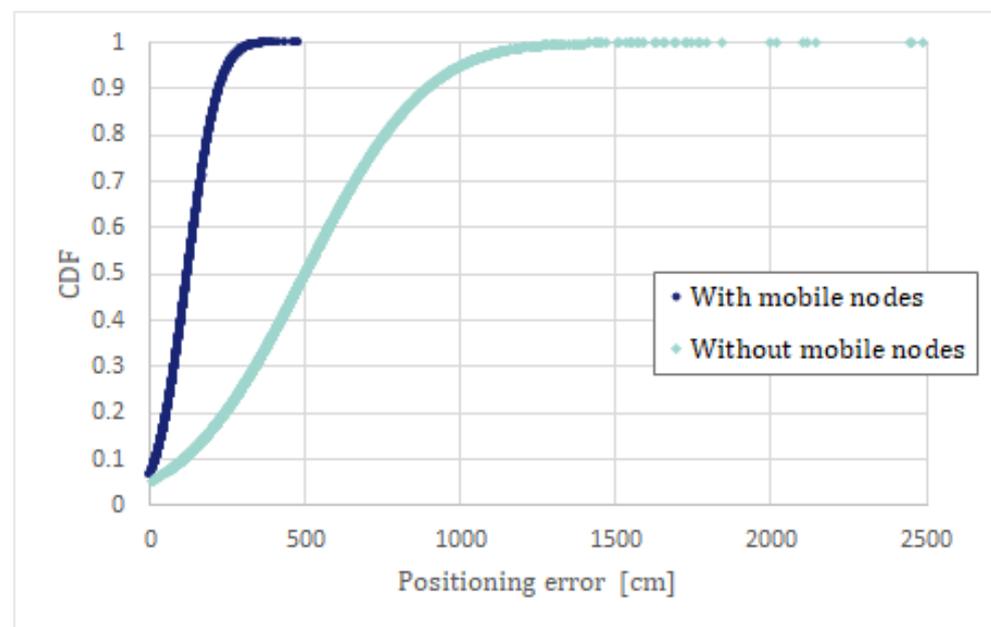
Figure 7 and Table 2 show the CDF of the average estimated errors for the proposed localization algorithm in terms of the average node degree in two cases: with and without mobile nodes. As shown in Table 2, localization errors remained low regardless of the average node degree when mobile nodes existed. The average location errors were 130.89,

123.73, and 121.99 cm when the average node degrees were 6, 9, and 12, respectively. On the other hand, the average node degree greatly affected the localization accuracy in the cases without mobile nodes. With the increase of the average node degree, the location error greatly increased from 1 to 10 m or more.

When the average node degree was set to 12, a simple comparison between with and without mobile nodes is shown in Figure 7. When mobile nodes existed, 90% of the nodes were located within an error of 2.5 m or less; otherwise, only 20% of the nodes were localized with similar accuracy. The comparison shows that the introduction of mobile nodes with a smartphone-based inertial navigation system significantly improved the overall localization performance of the IR-UWB-based multi-hop network.

**Table 2.** The CDF of the average estimated errors for the proposed cooperative self localization algorithm.

CDF	With Mobile Nodes			Without Mobile Nodes		
	6	9	12	6	9	12
0.1	23.75	21.12	19.76	114.48	10.71	105.99
0.2	60.61	56.49	54.76	706.87	280.24	240.60
0.3	87.17	81.69	80.03	1134.93	478.05	337.36
0.4	109.67	103.49	101.79	1500.69	647.07	420.16
0.5	130.92	123.71	122.02	1842.73	804.99	497.64
0.6	152.26	143.84	142.22	2183.95	963.78	575.20
0.7	174.65	165.70	163.33	2549.84	1131.92	657.70
0.8	201.19	190.96	189.03	2978.23	1326.42	754.51
0.9	237.60	225.92	224.00	3570.22	1604.24	888.80
1.0	505.65	569.89	479.80	8748.82	4353.20	2488.74



**Figure 7.** The CDF comparison at the average node degree of 12.

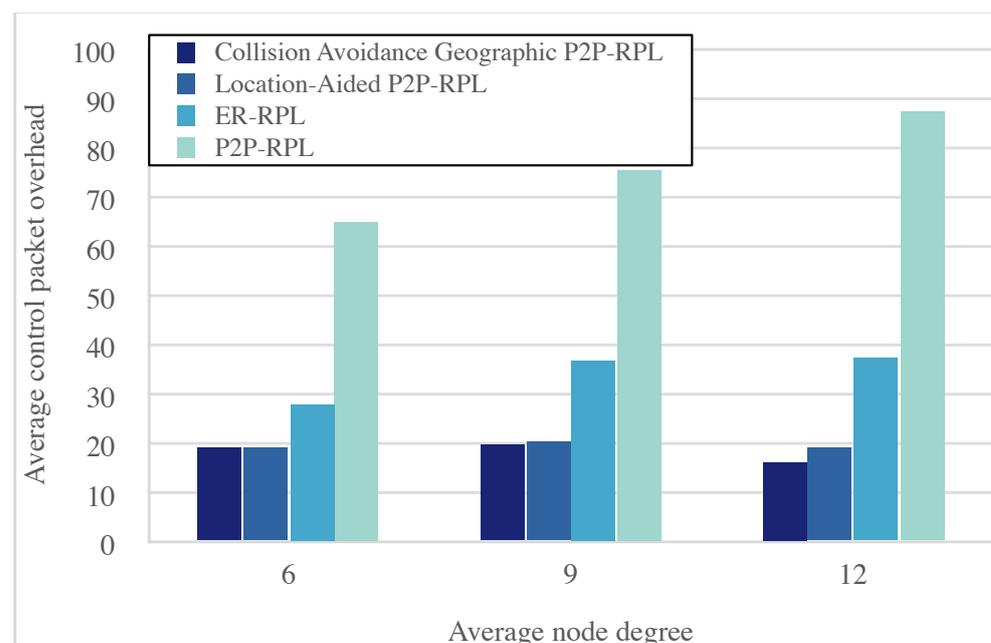
Figures 8–12 show the routing performance results with the increase of the average node degree. In this simulation study, each traffic flow started randomly at 1 s between arbitrarily selected source-destination pairs. The general simulation parameters are stated in Table 3. For each simulation, a different random topology was generated, and the result is the average value from all simulation runs.

The performance metrics to be observed are:

1. The average control packet overhead refers to the average number of control packets generated for a route discovery.
2. The average energy consumption is the average amount of energy consumption that is consumed for a route discovery. The first order radio model was used in this study as we considered only the energy consumption in the transmitting and receiving states [45]. The parameters used in the first order radio model are shown in Table 1.
3. The average hop count refers to the average hop length of the discovered P2P path between the source node and destination node.
4. The Packet Delivery Ratio (PDR) refers to the ratio of the successfully received packet number on the destination node to the total generated packet number by the source node.
5. The route discovery success ratio refers to the ratio of the total number of successful route discoveries to the total number of route discovery attempts.
6. The network convergence time refers to the latency from the instant in which the source node starts its first trickle interval until the instant at which the destination node has joined the temporary DODAG rooted at the source node.

**Table 3.** The simulation model parameters.

Parameter	Value
Simulation area	$75 \times 75 \text{ m}^2$
Radio environment	UDGM with distance loss
Transmission success ratio (%)	100
Reception success ratio (%)	90
Transmission range (m)	20
$I_{min}$	$2^6 \text{ ms}$
$I_{max}$	$2^{10} \text{ ms}$
Redundancy constant ( $k$ )	1
RPL mode of operation	Storing mode
Objective function	MRHOF
Routing metric	ETX



**Figure 8.** Average control packet overhead versus average node degree.

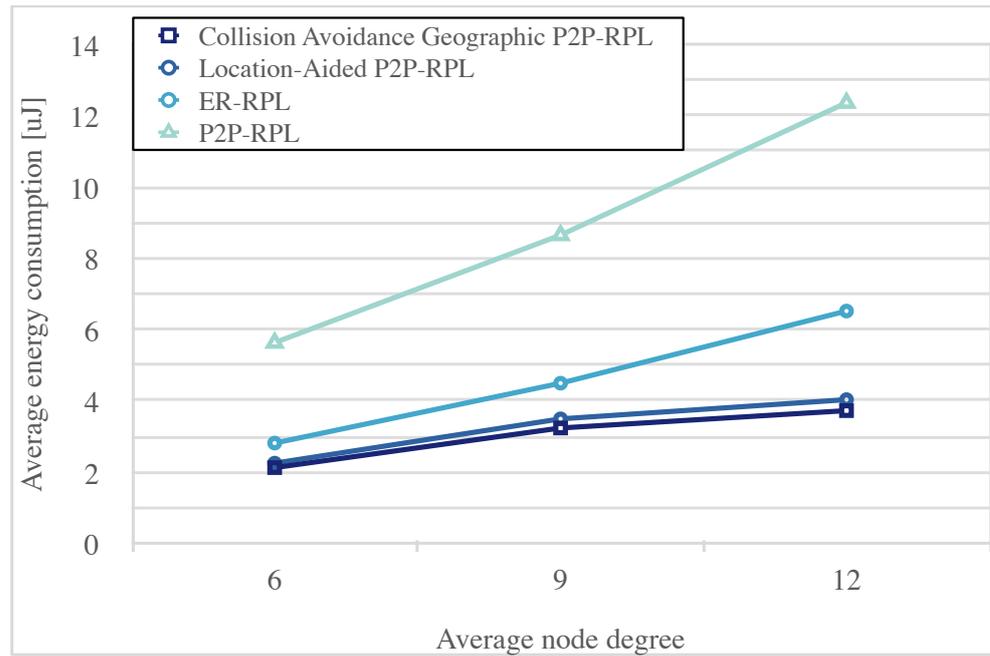


Figure 9. The average energy consumption versus the average node degree.

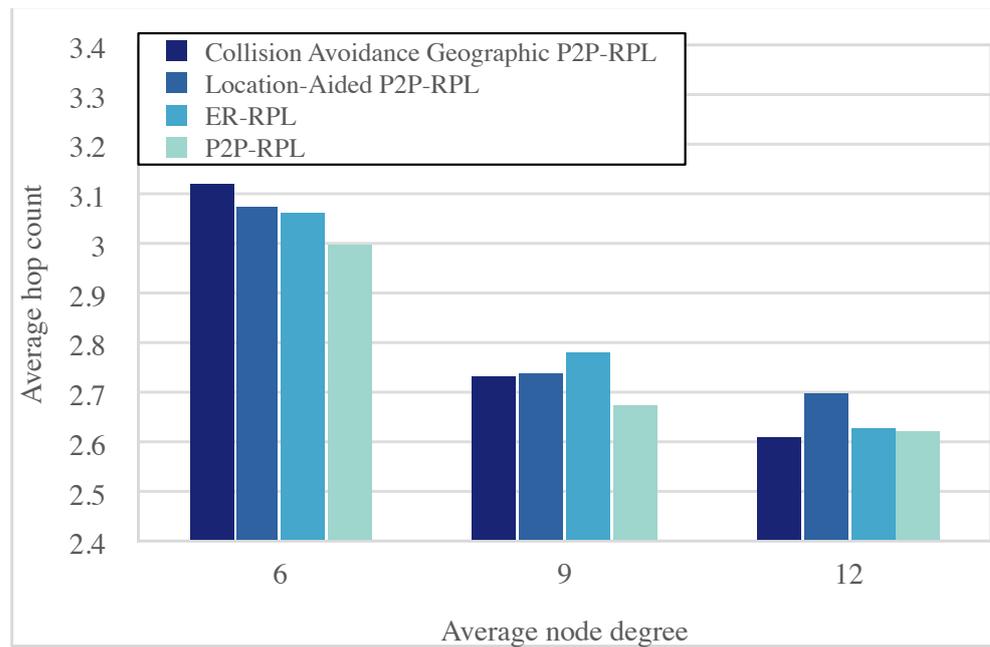
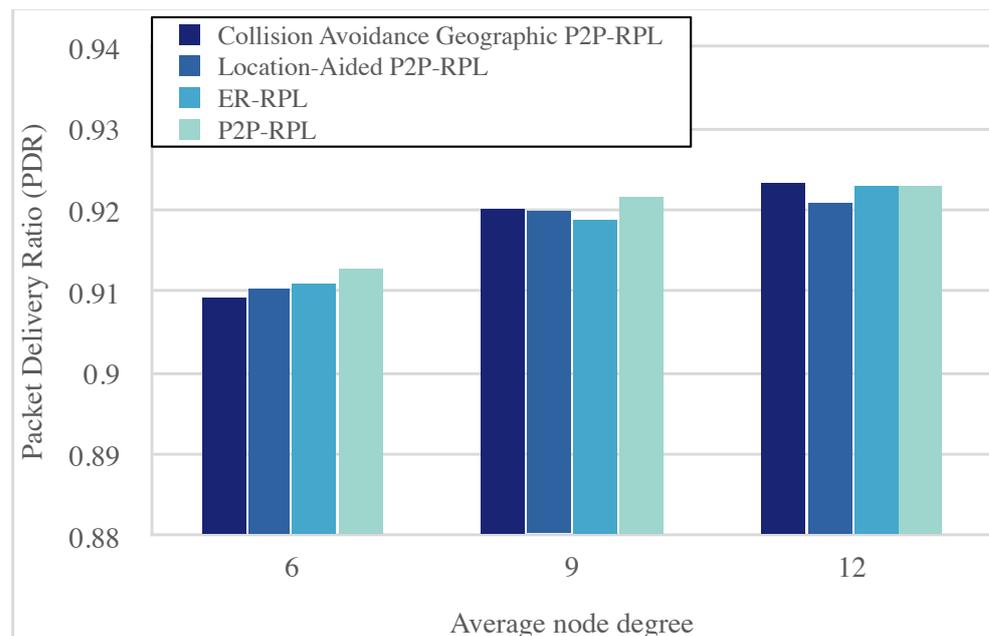
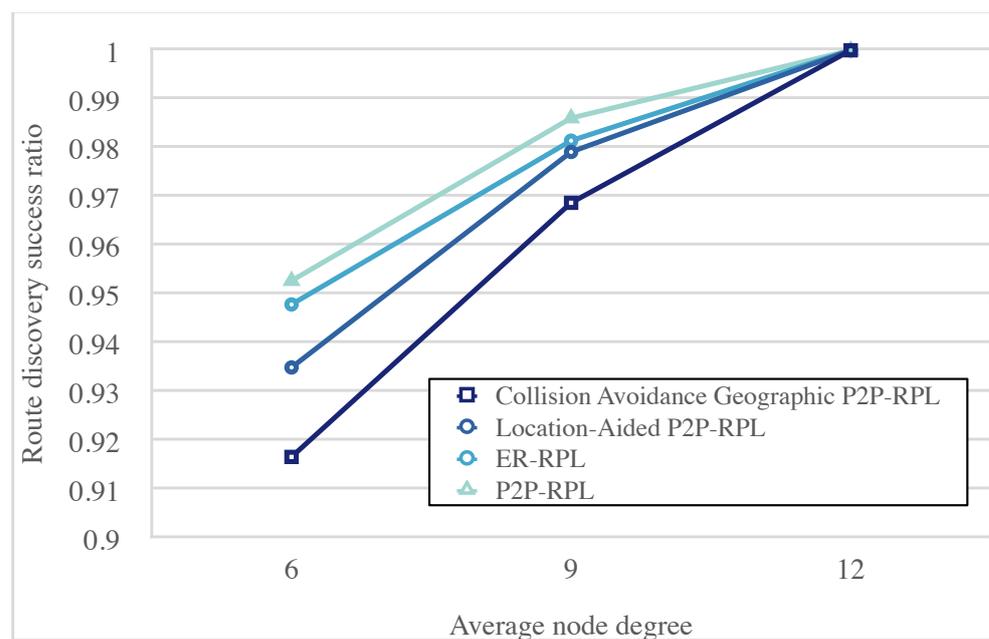


Figure 10. The average hop count versus the average node degree.



**Figure 11.** The packet delivery ratio versus the average node degree.



**Figure 12.** The route discovery success ratio versus the average node degree.

Figure 8 shows that Collision Avoidance Geographic P2P-RPL achieved about 75.7%, 45.9%, and 6.5% less control packet overhead on average compared with P2P-RPL, ER-RPL, and Location-Aided P2P-RPL, respectively. In Collision Avoidance Geographic P2P-RPL and Location-Aided P2P-RPL, only the node inside the location-constrained forwarding zone re-broadcasts a DIO control packet. Collision Avoidance Geographic P2P-RPL and Location-Aided P2P-RPL exploit accurate location information obtained from self-localization to reduce the search space for route discovery. The search space for route discovery is limited to the location-constrained forwarding zone based on the location information of the source node and the destination node.

Especially when the distance between source node and destination node is short, the location-constrained forwarding zone is limited to a small area. This results in signifi-

cantly reduced control packet overhead compared to P2P-RPL and ER-RPL. In ER-RPL, the estimated localization accuracy severely depends on the network configuration and the size of the search space for route discovery. As the average node degree increases, the performance differences with ER-RPL and P2P-RPL become larger.

Collision Avoidance Geographic P2P-RPL and Location-Aided P2P-RPL obtain more accurate self localization information with the increase of the average node degree. The performance difference between Collision Avoidance Geographic P2P-RPL and Location-Aided P2P-RPL comes from the difference in network convergence time. If P2P-DIO reaches the destination node earlier, the destination node sends P2P-DRO earlier and the nodes receiving P2P-DRO stop re-broadcasting a P2P-DIO for temporary DODAG formation.

Figure 9 shows that Collision Avoidance Geographic P2P-RPL achieved around 34.3% less energy consumption in comparison with ER-RPL and was reduced by more than twice compared to P2P-RPL on average. The energy model only considers the transmitting and receiving states; hence, the number of control packets constitutes a significant portion of the energy consumption.

In Collision Avoidance Geographic P2P-RPL, only the nodes inside the location-constrained forwarding zone re-broadcast a DIO control packet. This reduces the number of DIO control packets both transmitted and received. The impact of the average node degree is more significant on P2P-RPL and ER-RPL than on Collision Avoidance Geographic P2P-RPL. As the average node degree increases, more one-hop neighbor nodes exist in the transmission range of the sender node, resulting in an increased number of received control packets.

Figure 10 shows that the average hop count of P2P routes discovered by the Collision Avoidance Geographic P2P-RPL was very close to P2P-RPL when the average node degree was nine or more. P2P-RPL achieved the smallest hop count by discovering the P2P path through the whole network. Collision Avoidance Geographic P2P-RPL and Location-Aided P2P-RPL could not find a shorter route than P2P-RPL because these algorithms only search the location-constrained forwarding zone based on the location information of source and destination nodes. In this simulation network configuration, the network was fully connected. When the average node degree was 6, all algorithms discovered longer routes than when the average node degree was 9 or more. As the average node degree increased, shorter routes were discovered.

The PDR performance under various average node degree is shown in Figure 11. As the average node degree increases, PDR increases. The PDR of Collision Avoidance Geographic P2P-RPL is observed about 90.9% at the average node degree of 6, 92.0% at the average node degree of 9, and 92.3% at the average node degree of 12. Although partial nodes participate in route discovery procedure, ER-RPL, Location-Aided P2P-RPL and Collision Avoidance Geographic P2P-RPL still achieve close performance to P2P-RPL when the average node degree is 9 and 12.

Figure 12 shows the route discovery success ratio in the first route discovery attempt. As the average node degree increased, all algorithms could have a higher probability of finding the P2P route from the source node to destination node. Since P2P-RPL searches the P2P path throughout the entire network, ER-RPL, Location-Aided P2P-RPL, and Collision Avoidance Geographic P2P-RPL with limited search space for route discovery cannot improve the route discovery success ratio better than P2P-RPL.

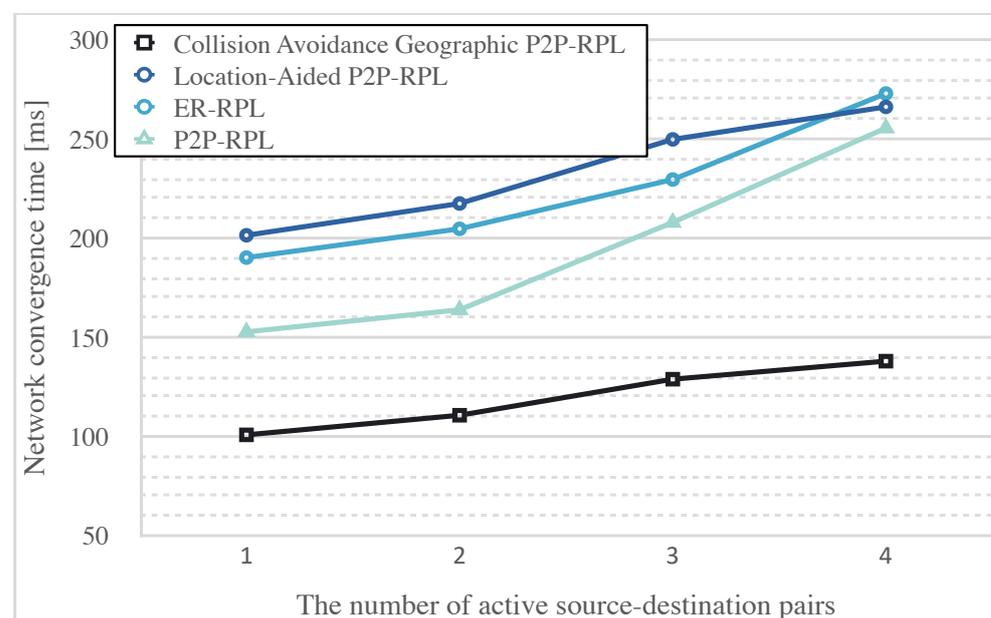
However, at the average node degree of 12, the route discovery success ratio performance of Collision Avoidance Geographic P2P-RPL was close to P2P-RPL with the accurate localization information. To improve the route discovery success ratio, Collision Avoidance Geographic P2P-RPL performed P2P-RPL by flooding the DIO control packets over the whole network if the route discovery failed in the first attempt.

Figures 13 and 14 show the routing performance results under various numbers of active source-destination pairs. The number of active source-destination pairs varied from one to four. The size of network area was  $75 \times 75 \text{ m}^2$ , and the average node degree was set to 12. Figure 13 shows that the network convergence time became longer as the number of active source-destination pairs increased. Due to the flooding nature,

more collisions occurred as the number of active source-destination pairs increased. The network convergence time increased with the number of collisions.

In Collision Avoidance Geographic P2P-RPL, the network convergence time is 38.7%, 46.7%, 48.8% shorter compared to P2P-RPL, ER-RPL, Location-Aided P2P-RPL, on average. Collision Avoidance Geographic P2P-RPL reduces the listen-only period up to 4 times shorter at each hop. A decrease in the listen-only period led to a decrease in the network convergence time. The proposed elastic trickle timer algorithm also allowed nodes to contend in longer intervals, resulting in a decreased probability of collisions.

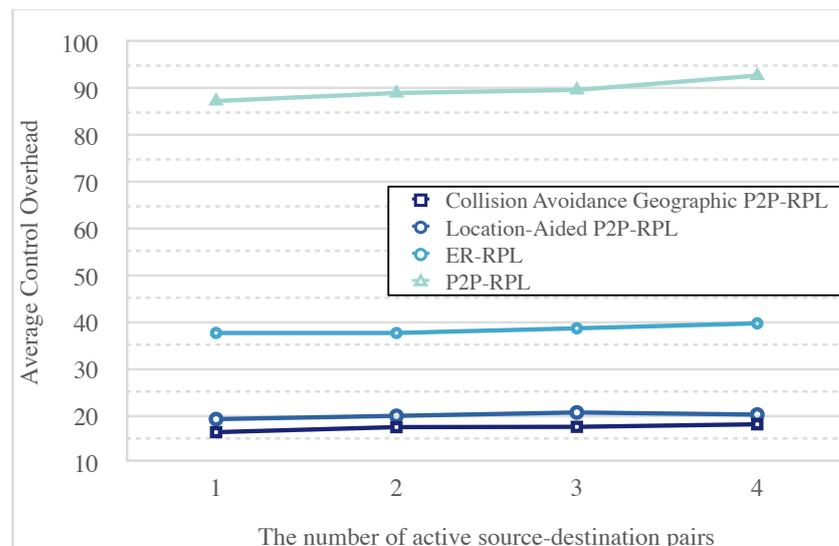
The network convergence time of Location-Aided P2P-RPL and ER-RPL was longer than for P2P-RPL. As shown in Figure 5, the non-deterministic suppression of standard trickle algorithm can cause a relay node to remain silent in the current trickle interval. This delays the DIO control packet in reaching the destination node, especially when high message suppression is performed (e.g.,  $k = 1$ ).



**Figure 13.** The network convergence time versus the number of active source-destination pairs.

Figure 14 shows that, as the number of active source-destination pairs increased, the number of control packet overhead increased. As the number of active source-destination pairs increased, the link losses caused by packet collisions increased. If the node on the shortest path from the source node to the destination node failed to send the DIO control packet to the intermediate nodes, the node retransmitted the DIO control packet in the next interval, thus, resulting in increased control packet overhead. Collision Avoidance Geographic P2P-RPL achieved average 13.0%, 54.8%, and 80.7% less control packet overhead compared with Location-Aided P2P-RPL, ER-RPL, and P2P-RPL, respectively.

The decrease in network convergence time means that P2P-DRO packets sent by the destination node finished the P2P route discovery procedure much faster. This reduced the number of P2P-DIO control packets for the amount of reduced network convergence time. The network convergence time difference of 114 ms between Location-Aided P2P-RPL and Collision Avoidance Geographic P2P-RPL led to additional DIO control packet overhead.



**Figure 14.** The control packet overhead versus the number of active source-destination pairs.

## 5. Conclusions

In this paper, we proposed the Collision Avoidance Geographic P2P-RPL to achieve low routing overheads with short convergence time and energy-efficiency for indoor IR-UWB-based networks. Collision Avoidance Geographic P2P-RPL makes use of the estimated location knowledge of the individual node in cooperative multi-hop IR-UWB self-localization networks. The proposed algorithm achieved energy-efficient P2P data delivery without reducing the networking reliability with a short convergence time.

The proposed algorithm utilizes standard RPL control messages and is compatible with the existing RPL implementations. We implemented Collision Avoidance Geographic P2P-RPL on Contiki OS and provide the experimental results using the Cooja network simulator under various network conditions. Our experimental results show that restricting the network search space for route discovery to the location-constrained forwarding zone significantly reduced the routing control packet overhead and energy consumption.

As the average node degree increased, the positioning accuracy increased, and the routing control packet overhead performance was improved. From various simulation results, the proposed algorithm showed more robust operation characteristics for P2P traffic while supporting a low routing overhead and short convergence time simultaneously. In future work, we aim to extend our work to a 3D environment and consider node mobility to reflect real-life environments more accurately.

**Author Contributions:** Conceptualization, Y.C.; methodology, Y.C.; validation, Y.C.; formal analysis, Y.C. and Y.K.; investigation, Y.C., J.J. and J.P.; resources, Y.C. and J.J.; data curation, Y.C.; writing-original draft preparation, Y.C. and J.J.; writing-review and editing Y.C. and Y.K.; visualization, Y.C.; supervision, Y.K.; project administration, Y.K. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Institutional Review Board Statement:** The study was conducted according to the guidelines of the Declaration of Helsinki, and approved by the Graduate School of Konkuk University on 14 June 2021.

**Informed Consent Statement:** Informed consent was obtained from all subjects involved in the study.

**Acknowledgments:** This paper was supported by Korea Institute for Advancement of Technology(KIAT) grant funded by the Korea Government (MOTIE) (N0002428, The Competency Development Program for Industry Specialist).

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Winter, T.; Thubert, P.; Brandt, A.; Hui, J.; Kelsey, R.; Levis, P.; Pister, K.; Struik, R.; Vasseur, J.P.; Alexander, R. RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks. 2012. Available online: <https://www.hjp.at/doc/rfc/rfc6550.html> (accessed on 25 January 2021).
2. Ghaleb, B.; Al-Dubai, A.; Ekonomou, E.; Alsarhan, A.; Nasser, Y.; Mackenzie, L. M.; Boukerch, A. A Survey of Limitations and Enhancements of the IPv6 Routing Protocol for Low-Power and Lossy Networks: A Focus on Core Operations. *IEEE Commun. Surv. Tutor.* **2019**, *21*, 1607–1635. [[CrossRef](#)]
3. Mahyoub, M.; Hasan Mahmoud, A.S.; Abu-Amara, M.; Sheltami, T.R. An Efficient RPL-Based Mechanism for Node-to-Node Communications in IoT. *IEEE Internet Things J.* **2021**, *8*, 7152–7169. [[CrossRef](#)]
4. Tang, C.; Zhang, Y.; Wu, Y. The P2P-RPL Routing Protocol Research and Implementation in Contiki Operating System. In Proceedings of the Second International Conference on Instrumentation, Measurement, Computer, Communication and Control, Harbin, China, 19 October 2012; pp. 1472–1475. [[CrossRef](#)]
5. Zhao, M.; Kumar, A.; Chong, P.H.; Lu, R. A comprehensive study of RPL and P2P-RPL routing protocols: Implementation, challenges and opportunities. *Peer Peer Netw. Appl.* **2016**, *10*, 1232–1256. [[CrossRef](#)]
6. Goyal, M.; Baccelli, E.; Philipp, M.; Brandt, A.; Martocci, J. Reactive Discovery of Point-to-Point Routes in Low-Power and Lossy Networks. 2013. Available online: <https://www.hjp.at/doc/rfc/rfc6997.html> (accessed on 25 January 2021).
7. Levis, P.; Patel, N.; Culler, D.; Shenker, S. Trickle: A self-regulating algorithm for code maintenance and propagation in wireless sensor networks. In Proceedings of the First USENIX Conference on Networked Systems Design and Implementation (NSDI), San Francisco, CA, USA, 29–31 March 2004.
8. Ghaleb, B.; Al-Dubai, A.; Ekonomou, E. E-Trickle: Enhanced Trickle Algorithm for Low-Power and Lossy Networks. In Proceedings of the 2015 IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing, Liverpool, UK, 26–28 October 2015; pp. 1123–1129. [[CrossRef](#)]
9. Ghaleb, B.; Al-Dubai, A.; Ekonomou, E.; Paechter, B.; Qasem, M. Trickle-Plus: Elastic Trickle Algorithm for Low-Power Networks and Internet of Things. In Proceedings of the 2016 IEEE Wireless Communications and Networking Conference (WCNCW), Doha, Qatar, 3–6 April 2016; pp. 103–108. [[CrossRef](#)]
10. Yassein, M.B.; Aljawarneh, S.; Masa'deh, E.; Ghaleb, B.; Masa'deh, R. A new dynamic trickle algorithm for low power and lossy networks. In Proceedings of the 2016 International Conference on Engineering & MIS (ICEMIS), Agadir, Morocco, 22–24 September 2016; pp. 1–6. [[CrossRef](#)]
11. Yassein, M.B.; Aljawarneh, S.; Masa'deh, E. A new elastic trickle timer algorithm for Internet of Things. *J. Netw. Comput. Appl.* **2017**, *89*, 38–47. [[CrossRef](#)]
12. Vallati, C.; Mingozzi, E. Trickle-F: Fair broadcast suppression to improve energy-efficient route formation with the RPL routing protocol. In Proceedings of the 2013 Sustainable Internet and ICT for Sustainability (SustainIT), Palermo, Italy, 30–31 October 2013; pp. 1–9. [[CrossRef](#)]
13. Zhao, M.; Ho, I.W.; Chong, P.H.J. An Energy-Efficient Region-Based RPL Routing Protocol for Low-Power and Lossy Networks. *IEEE Internet Things J.* **2016**, *3*, 1319–1333. [[CrossRef](#)]
14. Tseng, Y.C.; Ni, S.Y.; Chen, Y.S.; Sheu, J.P. The Broadcast Storm Problem in a Mobile Ad Hoc Network. *Wireless Netw.* **2002**, *8*, 153–167. [[CrossRef](#)]
15. Macker, J. Simplified Multicast Forwarding. 2011. Available online: <https://www.hjp.at/doc/rfc/rfc6621.html> (accessed on 25 January 2021).
16. Levis, P.; Clausen, T.; Hui, J.; Gnawali, O.; Ko, J. The Trickle Algorithm. 2011. Available online: <https://www.hjp.at/doc/rfc/rfc6206.html> (accessed on 25 January 2021).
17. Djamaa, B.; Senouci, M.R.; Bessas, H.; Dahmane, B.; Mellouk, A. Efficient and Stateless P2P Routing Mechanisms for the Internet of Things. *IEEE Internet Things J.* **2021**. [[CrossRef](#)]
18. Zamalloa M.Z.; Seada, K.; Krishnamachari, B.; Helmy, A. Efficient Geographic Routing over Lossy Links in Wireless Sensor Networks. *ACM Trans. Sens. Netw.* **2008**, *4*. [[CrossRef](#)]
19. Barriuelo, C.H.; Denardin, G.W.; Campos, A. A geographic routing approach for IPv6 in large-scale low-power and lossy networks. *Comput. Electr. Eng.* **2013**, *45*, 182–191. [[CrossRef](#)]
20. Kuhn, F.; Wattenhofer, R.; Zollinger, A. Worst-case optimal and average-case efficient geometric ad-hoc routing. In Proceedings of the International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc), Annapolis, MD, USA, 1–3 June 2003; pp. 267–278. [[CrossRef](#)]
21. Chakraborty, M.; Spanò, A.; Cortesi, A. Geographic Location Based Dynamic and Opportunistic RPL for Distributed Networks. In Proceedings of the International Conference on Computer Information Systems and Industrial Management, Belgrade, Serbia, 19–21 September 2019; pp. 120–131. [[CrossRef](#)]
22. Xu, C.; Xiong, Z.; Zhao, G.; Yu, S. An Energy-Efficient Region Source Routing Protocol for Lifetime Maximization in WSN. *IEEE Access* **2019**, *7*, 135277–135289. [[CrossRef](#)]
23. Simic, S.; Sastry, S. *Distributed Localization in Wireless Ad Hoc Networks*; Technical Report UCB/ERL; EECS UC BERKELEY: Berkeley, CA, USA, 2002; Volume 2, pp. 1–13.

24. Galstyan, A.; Krishnamachari, B.; Lerman, K.; Pattem, S. Distributed online localization in sensor networks using a moving target. In Proceedings of the Third International Symposium on Information Processing in Sensor Networks, Berkeley, CA, USA, 26–27 April 2004; pp. 61–70. [\[CrossRef\]](#)
25. Ruiz, A.R.J.; Granja, F.S. Comparing Ubisense, BeSpooon, and DecaWave UWB Location Systems: Indoor Performance Analysis. *IEEE Trans. Instrum. Meas.* **2017**, *66*, 2106–2117. [\[CrossRef\]](#)
26. Decawave Ltd. *APS006 Part 2: Non Line Of Sight Operation and Optimizations to improve performance in DW1000 Based Systems; Version 1.5*; Decawave Ltd.: Dublin, Ireland, 2018.
27. Jung, J.; Choi, Y.; Kwon, Y. Location-Aware Point-to-Point RPL in Indoor IR-UWB Networks. *Electronics* **2020**, *9*, 861–877. [\[CrossRef\]](#)
28. Arulampalam, M.S.; Maskell, S.; Gordon, N.; Clapp, T. A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. *IEEE Trans. Signal Process.* **2002**, *50*, 174–188. [\[CrossRef\]](#)
29. Crassidis, J.L.; Markley, F.L. Unscented Filtering for Spacecraft Attitude Estimation. *J. Guid. Control Dyn.* **2003**, *26*, 536–542. [\[CrossRef\]](#)
30. Kang, W.; Han, Y. SmartPDR: Smartphone-based pedestrian dead reckoning for indoor localization. *IEEE Sens. J.* **2015**, *15*, 2906–2916. [\[CrossRef\]](#)
31. Ho, N.-H.; Truong, P.H.; Jeong, G.-M. Step-Detection and Adaptive Step-Length Estimation for Pedestrian Dead-Reckoning at Various Walking Speeds Using a Smartphone. *Sensors* **2016**, *16*, 1423. [\[CrossRef\]](#)
32. Kos, A.; Tomazic, S.; Umek, A. Evaluation of Smartphone Inertial Sensor Performance for Cross-Platform Mobile Applications. *Sensors* **2016**, *16*, 477. [\[CrossRef\]](#)
33. Diebel, J. Representing attitude: Euler angles, unit quaternions, and rotation vectors. *Matrix* **2006**, *58*, 1–35.
34. Li, T.; Bolic, M.; Djuric, P.M. Resampling Methods for Particle Filtering: Classification, implementation, and strategies. *IEEE Signal Process. Mag.* **2015**, *32*, 70–86. [\[CrossRef\]](#)
35. Li, T.; Sattar, T.P.; Sun, S. Deterministic resampling: Unbiased sampling to avoid sample impoverishment in particle filters. *Signal Process.* **2012**, *92*, 1637–1645. [\[CrossRef\]](#)
36. Contiki-ng (New Development). Historical Contiki-os git repo. Thingsquare. Available online: <http://www.contiki-os.org> (accessed on 25 January 2021).
37. The Contiki Open Source OS for the Internet of Things. Available online: <https://github.com/contiki-os/contiki> (accessed on 25 January 2021).
38. Available online: <http://contiki-p2p-rpl.gforge.inria.fr> (accessed on 25 January 2021).
39. Thubert, P. Objective Function Zero for the Routing Protocol for Low-Power and Lossy Networks (RPL). 2012. Available online: <https://www.hjp.at/doc/rfc/rfc6552.html> (accessed on 25 January 2021).
40. Gnawali, O.; Levis, P. The Minimum Rank with Hysteresis Objective Function. 2012. Available online: <https://www.hjp.at/doc/rfc/rfc6719.html> (accessed on 25 January 2021).
41. Tsiftes, N.; Eriksson, J.; Finne, N.; Österlind, F.; Höglund, J.; Dunkels, A. A Framework for Low-Power IPv6 Routing Simulation, Experimentation, and Evaluation. *ACM SIGCOMM Comput. Commun. Rev.* **2010**, *40*, 479–480. [\[CrossRef\]](#)
42. DecaWave EVB1000/DWM1001 Contiki Port with Integrated Glossy and Crystal. Available online: <https://github.com/d3s-trento/contiki-uwb> (accessed on 25 January 2021).
43. Korbi, I.E.; Ben Brahim, M.; Adjih, C.; Saidane, L.A. Mobility Enhanced RPL for Wireless Sensor Networks. In Proceedings of the 2012 Third international conference on the network of the future (NOF), Gammarth, Tunisia, 21–23 November 2012; pp. 1–8. [\[CrossRef\]](#)
44. Cobârzan, C.; Montavont, J.; Noël, T. Analysis and performance evaluation of RPL under mobility. In Proceedings of the 2014 IEEE Symposium on Computers and Communications (ISCC), Madeira, Portugal, 23–26 June 2014; pp. 1–6. [\[CrossRef\]](#)
45. Heinzelman, W.R.; Chandrakasan, A.; Balakrishnan, H. Energy-efficient communication protocol for wireless microsensor networks. In Proceedings of the 33rd Annual Hawaii International Conference on System Sciences, Maui, HI, USA, 4–7 January 2000; p. 10. [\[CrossRef\]](#)