The following publication M. Zhao, I. W. -H. Ho and P. H. J. Chong, "An Energy-Efficient Region-Based RPL Routing Protocol for Low-Power and Lossy Networks," in IEEE Internet of Things Journal, vol. 3, no. 6, pp. 1319-1333, Dec. 2016 is available at https://doi.org/10.1109/JIOT.2016.2593438.

An Energy-efficient Region-based RPL Routing Protocol for Low-Power and Lossy Networks

Ming Zhao, *Student Member, IEEE,* Ivan Wang-Hei Ho and Peter Han Joo Chong, *Member, IEEE*

***Abstract*—Routing plays an important role in the overall architecture of the Internet of Things. IETF has standardized the RPL routing protocol to provide the interoperability for Low-Power and Lossy Networks (LLNs). LLNs cover a wide scope of applications, such as building automation, industrial control, healthcare and so on. LLNs applications require reliable and energy-efficient routing support. Point-to-point (P2P) communication is a fundamental requirement of many LLNs applications. However, traditional routing protocols usually propagate throughout the whole network to discover a reliable P2P route, which requires a huge energy consumption. Again, it is challenging to achieve both the reliability and energy-efficiency simultaneously, especially for LLNs. In this paper, we propose a novel energy-efficient region-based routing protocol, called ER-RPL, which achieves energy-efficient data delivery without compromising the reliability. In contrast of traditional routing protocols where all the nodes are required for route discovery, the proposed scheme only requires a subset of nodes to do the job, which is the key of energy saving. Our theoretical analysis and extensive simulation studies indicate that ER-RPL has a great performance superiority over two conventional benchmark protocols, RPL and P2P-RPL.**

***Index Terms*—Low-power and Lossy Networks (LLNs), RPL, region-based, energy-efficiency, reliability, Point-to-point (P2P) communication.**

I. Introduction

MACHINE-to-Machine (M2M) communications [1] aim to achieve ubiquitous communication among intelligent devices for application control and monitoring, which have attracted both academia and industry in recent years. Motivated by the great potential of M2M communications, many standardization activities, such as IETF, 3GPP, and IEEE have defined protocol stacks to enable the design and implementation of M2M communications [2]. M2M devices are usually battery powered and operate in harsh environment (e.g., heat, dust, and moisture weather). LLNs are used to categorize the dynamic and lossy environment where a large group of highly constrained devices are interconnected by unreliable wireless links. Routing plays a crucial role to provide the interoperability among network components. The IETF Routing Over Low-power and Lossy network (ROLL) working group standardizes the IPv6 Routing Protocol for Low Power and Lossy Networks (RPL) [3], [4], [5], which targets highly constrained nodes and large scale networks for LLNs applications. Point-to-point (P2P) communication is a fundamental requirement in lots of LLNs applications.

M. Zhao and P. H. J. Chong are with the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore.
E-mail: {MZHAO003, EHJCHONG}@ntu.edu.sg
Ivan W. H. Ho is with the Department of Electronic and Information Engineering, The Hong Kong Polytechnic University, Hong Kong.
E-mail: ivanwh.ho@polyu.edu.hk

LLNs applications require efficient P2P routing support. However, achieving high reliability and consuming less energy at the same time, is a challenging issue. The routing paths between arbitrary M2M devices are not provided by default due to the resource constraints. Traditional routing protocols, such as Lightweight On-demand Ad hoc Distance-vector Routing (LOADng) [6] and P2P-RPL [7], disseminate the route discovery messages throughout the network to find the optimal P2P route, but they consume a large amount of overhead and a great energy cost. In this paper, we propose an energy-efficient region-based routing protocol (ER-RPL), which achieves energy-efficient P2P communication without compromising the reliability. In contrast to traditional routing protocols, where all nodes in the network participate in the route discovery process, ER-RPL only requires a subset of nodes in some regions to join the route discovery. A nearly optimal route in terms of reliability can be discovered with a great energy conservation.

Our proposed ER-RPL uses the region information to support an efficient P2P communication. For static networks, such as M2M networks and wireless sensor networks (WSNs), the area where a node resides is a piece of important information. Many LLNs applications exploit this region feature. For example, in automatic control systems, the control command can be sent to all devices in one level or a region/room of a building. In event-triggered applications, all sensors within an area can capture this event, but which sensor has collected the information may not be important. The region information can be used to efficiently discover the routing paths. Overall, the key contributions of this paper are summarized as follows:

1) We propose a novel scalable routing protocol called ER-RPL, to achieve both reliable and energy-efficient data delivery for static networks. Significant control overhead reduction is achieved because only a portion of nodes in the network participate in the route discovery.
2) We propose a P2P traffic model with the routing decision making for the lossy networks.
3) We evaluate the ER-RPL protocol through both theoretical analysis and extensive simulation.
4) Another two conventional routing protocols are used as the benchmarks in the simulation study. In comparison with the benchmarks, ER-RPL can achieve more reliable data delivery with a greater energy conservation.

The rest of the paper is organized as follows. In Section II, we introduce the existing LLNs routing protocol and discuss the challenges of designing efficient routing protocol for LLNs. Section III describes the proposed protocol ER-RPL.

Theoretical analysis is presented in Section IV. Simulation and performance evaluation are conducted in Section V. Finally, this paper is concluded in Section VI.

II. Background

In recent years, LLNs applications has emerged as the predominant paradigm for M2M communications. LLNs applications cover a wide range of scenarios, including but not limited to, building automation, industrial control, urban environment and home automation [8], [9], [10]. LLNs applications essentially require reliable and energy-efficient routing to support the connectivity of network utilities with tighter control and energy conservation. However, nodes operating in LLNs usually have limited battery power and communicate via dynamic and lossy wireless medium. It is inherently challenging to achieve the reliability and the energy-efficiency at the same time, especially for LLNs. It has been shown that traditional routing protocols cannot provide an efficient routing support for LLNs [11]. For P2P communication, three main routing techniques have been proposed: proactive routing, on-demand (reactive) routing and geographic routing. In this section, we first introduce several conventional LLNs routing protocols and analysis their limitations. Then we summarize suits of main challenges of designing routing protocols for LLNs, and present the routing solutions that are implemented in our proposed protocol.

A. *Routing Protocols for Low-Power and Lossy Networks*

RPL is a proactive IPv6-based distance-vector routing protocol. RPL can establish a Destination Oriented Directed Acyclic Graph (DODAG) at a high speed with the trickle algorithm [12]. According to applications' objectives, RPL uses different routing metrics to support LLNs applications. The root node serves as a transit point bridging the DODAG with the IPv6 network. The formation of a DODAG is initiated by the root that periodically originates DODAG Information Object (DIO). RPL is designed with optimization to support multipoint-to-point (MP2P). RPL chooses the best next hop as the preferred parent to the root given the particular objective function [13]. Although RPL support the routing for generic traffic pattern. For P2P communication, RPL needs to pre-establish routes and can only route along pre-established DAGs. The source node has to send the packet upwards until it reaches the ancestor node of the destination node. Then the common ancestor node will deliver the packet downwards towards the destination node. In the non-storing mode of RPL, the common ancestor has to be the root. Hence, the packet delivery usually suffer from longer delay and lossy links. Additionally, the root becomes a bottleneck when the traffic becomes heavy.

A reactive P2P route discovery mechanism with RPL is defined as P2P-RPL [7], [14]. In P2P-RPL, a temporary DAG rooted at the source node is built to facilitate the end-to-end traffic transmission in LLNs when there is no available route expect for the basic RPL strategy. A P2P Route Discovery Option (P2P-RDO), which is piggybacked in DIO, is used for the route discovery in P2P-RPL. The source node originates

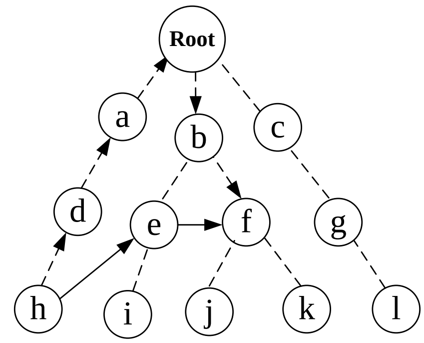


Fig. 1 Communication routes between node *h* and node *f*, provided by RPL with the dotted arrows and by P2P-RPL with solid arrows.

the route discovery message and disseminates throughout the whole network with the trickle algorithm. The same DODAG formation mechanism as RPL is used to set up a temporary DODAG for P2P route discovery. The lifetime of the DODAG is strictly restricted by the lifetime of the route request. Once the destination node receives this temporary DODAG information, it replies a P2P Discovery Reply Object (P2P-DRO) to the source through the discovered route. The reverse route will be used for the P2P transmission. Fig. 1 shows the communication routes from node h to f in RPL and P2P-RPL. P2P-RPL can usually find better P2P route than RPL. But sometimes the route discovered by P2P-RPL may not be much better than the existing route in RPL, which is through the DAG root from the source to the destination. The improvement of route quality depends on the network topology, such as the distance between the source and the destination. Moreover, the cost associated with the P2P route discovery is very expensive in LLNs, because all nodes in the network needs to participate in temporary DODAG formation for the route discovery, which leads to significant cost in energy-constraint networks.

LOADng is a reactive routing protocol derived from Ad hoc On-demand Distance Vector Routing (AODV) [15] for LLNs. The basic operation of LOADng is similar to that of AODV, but it makes lots of simplifications. During the route discovery stage, route request (RREQ) messages will be distributed throughout the network. If the destination receives the RREQ message, it will reply a route-reply (RREP) message through the reverse path. LOADng supports more traffic patterns, such as P2P, point-to-multipoint (P2MP), and MP2P. But it shows disadvantage in terms of very much overhead for MP2P traffic pattern compared with RPL [16]. The performance comparison and detail analysis of LOADng with RPL have been studied with different traffic patterns [17], [18]. Because both P2P-RPL and LOADng disseminate the whole network for the best route discovery given particular objective, so LOADng is not addressed in the simulation study.

Geographic routing relies on nodes' locations (either real coordinates or virtual coordinates) instead of nodes' addresses to forward data packets in a greedy manner [19], [20]. Nodes choose the node closest to the destination as their next hop to deliver data packet towards the destination. The location/coordinate is known by each node or partial nodes either as a priori knowledge or through a self-configuration localization scheme. Geographic routing has the advantages of a low overhead and the scalability feature, but it does not take the lossy nature of wireless links into consideration when it chooses the next hop. Consequently, geographic routing usually cannot cope well with the lossy wireless medium and provide reliable data delivery support for LLNs. In addition, some geographic routing protocols require nodes to maintain many states and have to exchange the one-hop or even two hop neighbour table periodically [20], which is very costly for a resource constrained network.

B. Routing Challenges in Low-Power and Lossy Networks

Efficient support for generic traffic patterns. LLNs applications require stable routing support for generic traffic patterns, such as MP2P, P2P, and P2MP with heterogeneous node capabilities [11]. However, devices in LLNs applications have limited memory. In order to keep the routing table size small, the routing path between two arbitrary nodes is usually not provided by default. Thus the route discovery is required when there is no available route between a source and destination pair.

Reliable routing in dynamic and lossy environment. It is extremely crucial and challenging to achieve reliable routing in dynamic and lossy environment. Data transmission suffer from link loss in LLNs. In addition to the harsh environment, the channel fading and co-channel interference add more uncertainties to the wireless channels. Unreliable wireless channels usually cause more retransmissions, which results in higher energy consumptions and longer channel occupancy time. It is vital to discovery the reliable routing path for data delivery in LLNs.

Energy-efficient route discovery. In LLNs, a large number of energy constraint nodes may be potentially inaccessible due to the constraints of physical environment in practice. It is critical to conserve power and prolong the lifetime of the network, so that superior network performances can be attained by maintaining a desired network connectivity. The expenditure of energy results from transmitting and receiving both data packets and control packets. Control packets are used for topology construction in proactive routing and route discovery in reactive routing. For nodes suffering from severe energy constraints, the transmission and receiving of control packets are very expensive in terms of energy consumption. In addition, nodes in LLNs are usually battery-powered and have to run complex computational processes. Traditional routing protocols disseminate the route discovery messages throughout the whole network in all directions, which incurs very much overhead. Consequently, routing overhead should be minimized so as to conserve energy in the design of routing protocols for LLNs.

Link asymmetry in real-world scenarios. Empirical studies have shown that wireless links have asymmetric nature. It is mainly because the transmitter power and receiver sensitivity are different from nodes to nodes [22]. Some other factors, such as the reflectors, absorbers, etc, also result in link asymmetry. The asymmetric nature of wireless links has significant impact on the routing protocols' performance, especially for LLNs. Protocols without considering the link asymmetry sometimes fail when the link asymmetry is encountered in the real deployments [23]. Therefore, the asymmetry properties of wireless channels needs to be considered in the protocol design.

Scalability support for the large scale network. LLNs are normally large scale and require scalability support of routing protocols. Routing protocols for LLNs have to be scalable so as to accommodate large and increasing number of nodes. Scalability is one of the most important criteria for the design of LLNs routing protocols [21].

To address the above challenges and limitations of existing LLNs routing protocols, ER-RPL consists of the following four major components:

1) ER-RPL is designed with the capability to support the generic traffic patterns, because it takes advantage of the existing DODAG structure of RPL in addition to its efficient support for P2P route discovery.
2) ER-RPL exploits the region feature of static networks. Only a portion of nodes are required to participate in the route discovery for a source-destination node pair. In addition to the region-based route discovery, Region-to-Region (R2R) routing without route discovery is implemented as an enhancement. These designs result in a great reduction of control overhead. In this way, reliable routing paths can be discovered in an energy-efficient way.
3) A distributed Self-regioning algorithm is proposed for nodes to obtain their region codes (RCs). Meanwhile, the region-based route discovery also works in a decentralized manner. Therefore, ER-RPL has the capability to support network scalability.
4) The asymmetric nature of wireless links are considered in the protocol design. ER-RPL is robust to different wireless channel conditions

III. Proposed Protocol: ER-RPL

Based on the key challenges and insights described in the Section II, we propose a hybrid of proactive and reactive routing protocol, namely ER-RPL, to achieve reliable data delivery in an energy-efficient manner. We first provide the system model of ER-RPL, and then present the key stages in ER-RPL for efficient P2P data delivery.

A. System Model

1) Preliminary: We consider that n number of stationary nodes are densely deployed on an area with the size of G. Mobile nodes are out of the scope of this paper. All nodes' transmission range is R. In the network, we consider a set of nodes with location-awareness capability (e.g., with GPS), which are called Reference Nodes (RNs), as shown in Fig. 2. Assume that N denotes the number of RNs in the network, where $N \ll n$. The set of all RNs in the network is denoted by $\Omega = \{RN_1, RN_2, \ldots, RN_N\}$. With the help of RNs, the network area are segmented into several regions. The nodes without position knowledge are regarded as normal nodes ("nodes" in the following of this paper). The objective of protocol design is to energy-efficiently discover reliable

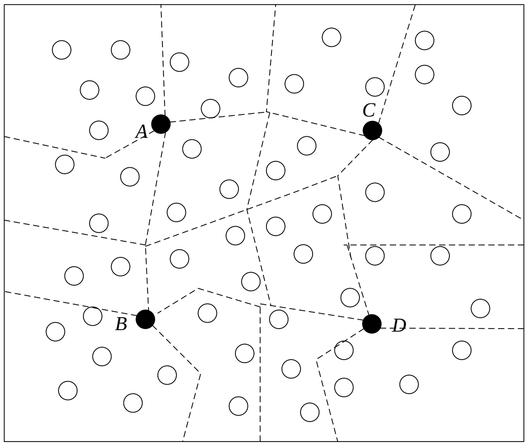


Fig. 2 An example of reference node and the regions in the network

Table I Summary of notations

Symbol	Definition
n	number of nodes
N	number of reference nodes
G	network size
Ω	The set of all RNs
ρ	network density
R	the transmission range of node and RN
λ	the number of traffic flows
I_{max}	the maximum time interval size
p_{ij}	the successful delivery probability from node i to j
E_{ij}	one hop ETX from node i to j
C_I	the *Coordinate* of RN I
h_{iI}	the hop count of arbitrary node i to RN I
$RNM(i,j)$ or RN_{ij}	the entry in the ith row and jth column of the matrix RNM
$RCM(m,n)$ or RC_{mn}	the entry in the mth row and mth column of the matrix RCM

routes for nodes without location-awareness capability so as to support reliable and energy-efficient P2P communication. An example of P2P communication can be found in a remote control application. A motion sensor (node *s*) suddenly needs to communicates with a lamp module (node *d*). Node *s* and node *d* are normal nodes in the network and do not have the location-awareness capability. The source node *s* needs to send the control command to the destination node *d* via multihop routing. With ER-RPL, a reliable route between node *s* and node *d* can be discovered in an energy-efficient manner.

Our proposed ER-RPL makes use of a small amount of location-aware RNs, but it differs from the geographic routing in three fundamental ways. 1) Different from geographic routing, ER-RPL establishes the best quality route in terms of reliability in a reactive manner. 2) Nodes do not have the unique geographic coordinates knowledge. The region information is only used to determine whether the node should participate the route discovery or not. 3) In ER-RPL, the data delivery relays on the node's address and routing table instead of the virtual or real coordinates. These features make our proposed ER-RPL essentially different from geographic routing.

Two scenarios are considered in our study. In the first scenario, the node are nearly uniformly distributed so that the value of ρ is a constant. In the second scenario, the value of ρ is likely to change due to the irregular node deployment. Table I shows some notations to be used in this paper.

2) P2P Traffic Modelling in LLNs: Packets drop takes place in both node level and link level [22]. The link level channel contention and node level resource limitation will affect the communication quality. In this research study, we assume that there is enough buffer space at the relay node, hence the packet drop rate at an arbitrary node *i* is approximately 0.

Assume that the link loss is independent and identically distributed (i.i.d). The successful delivery probability from node *i* to node *j* can be denoted by p_{ij}. We assume that wireless links have bidirectional readability and have asymmetric nature, which means p_{ij} may not be equal to p_{ji} for the delivery of packets with the same size. Retransmission is implemented in the Medium Access Control (MAC) layer to improve the network reliability. The Expected Transmission count (ETX) [24] is used to measure the quality of wireless links. One hop ETX means the average number of transmissions required to successfully deliver one packet to the next hop, which is defined as $E_{ij} = 1/p_{ij}p_{ji}$, where p_{ij} is the probability of successful delivery of a data packet from node i to i, and p_{ji} in this equation refers to the probability of successful delivery of an ACK from node j to node i. Because the ACK usually has very small size, which can be recovered with the strong coding techniques. Hence, p_{ji} for ACK is approximate to one. In this study, we assume ACK does not suffer from link loss, so we can get $E_{ij} = 1/p_{ij}$. The aggregate ETX is used to select the best route for a source-destination node pair.

The traffic load from source node *i* to destination node *j* is denoted by L_{ij}, where $i \neq j$. Then the traffic load distribution matrix of the network is

$$L = \begin{bmatrix} L_{00} & \cdots & L_{0N} \\ \vdots & \ddots & \vdots \\ L_{N0} & \cdots & L_{NN} \end{bmatrix}, \tag{1}$$

where for $i \in n, L_{ii} = 0$.

For P2P communication, a node needs to choose the next hop among its neighbouring nodes for packet forwarding. This routing decision has significant impact on the network performance. To elucidate this, let δ_{ij} be the routing choice made by node *i* to node *j*, where $j \in N(i)$ and $N(i)$ denotes node i's neighbour node set. If δ_{ij} is equal to 1, it means that node *i* chooses node *j* as its next hop. If the value of δ_{ij} is 0, it means that node *i* does not choose node *j* as its next hop. We use $r_{i,s}$ to denote the traffic generation rate at node *i*, and use $r_{i,d}$ to denote the arrival rate of the traffic destinated at node *i*. Assume that the traffic receiving rate of node *i* from node *j* is $f_{j,i}$, and the traffic departure rate at node *i* to node *j* is $f_{i,j}$. In this way, the P2P traffic model for node *i* can be formulated as

$$r_{i,s} + \sum_{j \in N(i)} \delta_{ji} f_{ji} - \sum_{j \in N(i)} \delta_{ij} f_{ij} p_{ij} = r_{i,d}, \tag{2}$$

where $0 \leq f_{ji} \leq C_{ji}, \forall i,j \in n$ and $i \neq j$.

Let us define the network running time as *T*. At time t, the traffic generation rate of node *i* is $r_{i,s,t}$. The traffic arrival rate of node i as destination is represented by $r_{i,d,t}$. The duration of traffic flow is Δt. In this way, the traffic load of node i is shown in Eq. (3) and (4).

$$\int_0^T r_{i,s,t}\Delta t_i = L_{i0} + L_{i1} + ... + L_{iN} = \sum_{0\le j\le N} L_{ij}, \quad (3)$$

$$\int_0^T r_{i,d,t}\Delta t_i = L_{01} + L_{1i} + ... + L_{Ni} = \sum_{0\le j\le N} L_{ji}. \quad (4)$$

3) Energy Model: We model nodes with four basic states (transmit, receive, idle, sleep) and a transition state among them [25]. The energy consumption in transmitting and receiving states is the key issue to address in this research study. We use the First Order Radio Model [26], which has been widely used to measure the energy dissipation for wireless sensor networks (WSNs) [27], [28]. In this model, the radio consumes E_{elec} to run the transmitter or receiver circuitry. The transmitting amplifier is ϵ_{amp}, which is used to achieve the acceptable signal-to-noise (SNR) ratio. We assume the propagation loss exponent is 2. In this way, the energy consumption for transmitting a l-bit message with a transmission range R is modelled as

$$E_{tx}(l, R) = lE_{elec} + lR^2\epsilon_{amp}, \quad (5)$$

The energy consumption of the receiver is modeled as

$$E_{rx}(l, d) = lE_{elec}. \quad (6)$$

B. Overview of ER-RPL

ER-RPL inherits the mechanism from RPL to pre-establish the DODAG such that it can support the multipoint-to-point (MP2P) with the optimized topology. Additionally, ER-RPL discovers the best P2P route with the region information in a reactive manner. Both the RNs and nodes are stationary in this work. In ER-RPL, each RN belongs to an area, and that area is divided into a configurable number of non overlapping regions. The region number associated with each RN can be configured with the Self-regioning algorithm. Then each node starts to estimate which region it resides in. During the route discovery for a source-destination node pair, ER-RPL requires a node to make a judgement whether it should participate or not with the region knowledge of the source and destination nodes. In this way, the route discovery is only performed among a subset of nodes in the network, which leads to significant control overhead reduction. Basically, ER-RPL includes two main stages.

a). Network initialization stag: Each RN computes its *Coordinates*, which is defined as the average distance per hop count in this paper. Then, each node estimates the distances to all RNs based on RNs' *Coordinates* values and their hop counts to the corresponding RNs. A region code (RC), which represents the particular region a node resides in the network, is obtained by a node with the distributed Self-regioning algorithm.

b). Route discovery stage: According to the Region Codes (RC) of source node and destination node, the route discovery is only performed among a subset of nodes in the network with the region-based route discovery. Besides, the Region-to-Region (R2R) routing without route discovery is designed as an enhancement to the region-based route discovery.

Details of these two stages are discussed in the following sections.

C. Network Initialization Stage

A series of positioning algorithms have been proposed, which are well known as the Ad Hoc Positioning System (APS). APS includes six algorithms: DV-Distance, DV-Hop, Euclidean, DV-Bearing, DV-Coordinate, and DV-Radial [29]. Different from the work in [29], where the *Coordinates* is defined as a correction factor, our work defines the *Coordinates* as the average distance per hop of RNs. ER-RPL uses an enhanced DV-Hop algorithm, which uses the RNs' *Coordinates* and the number of hops to RNs to estimate the distances between a node to the RNs. It is worthy to highlight that the number of RNs in a network is known in advance by all RN nodes. RNs play a critical role in the network initialization stage, but they do not perform any task during the reactive P2P route discovery and data delivery stage.

1) Coordinates of Reference Node: Each RN computes its *Coordinates* value based on its relative distances and hop counts towards other RNs in the network. Upon receiving the topology formation information initiated by the root, each RN serves as a temporary "root" and builds up a temporary DODAG using the Minimum Hop Count [30] as the routing metric. The construction of temporary DODAG is achieved by disseminating the Region Formation Objec (RFO) message, and the regions are initialized during this process. Fig. 3 shows the packet structure of the RFO message. The *DODAG_RN* is the IP address of the RN. The *Rank* is used to update the hop distance to this RN. The *Position* carries the geographical position of this RN. There are two stages of the RFO operation, which is indicated by the *CFlag*. When the *CFlag* is set to be zero by the RN, it indicates that the current stage is the *Coordinates Computation* stage of this RN. When the *CFlag* is set to be one, it is in the *Euclidean Distance Calculation* stage. The *Coordinate* field records the *Coordinates* of the corresponding RN. The temporary DODAG rooted at RN *I* is denoted as *DODAG_RN*$_I$, where *I* is an arbitrary RN in the network area. A node will first check the *CFlag* field when it receives an RFO message from its one hop neighbour node.

During the *Coordinates Computation* stage, a node will join the temporary DODAG, choose the next hop, update its rank, and broadcast its status with an RFO message. Similar to RPL basic DODAG set up, the rank is monotonically increased based on next hop towards that RN. The RFO operation also follows the Trickle Algorithm. In this way, the information will be dismissed throughout the whole network. Based on the control message exchange and rank value update, each RN is able to collect the information including other RNs' positions and the distances between other RNs in the network. N temporary DODAGs will be constructed with

DODAG_RN	Rank	CFlag	Coordinate	Position

Fig. 3 The Packet Structure of RFO Message

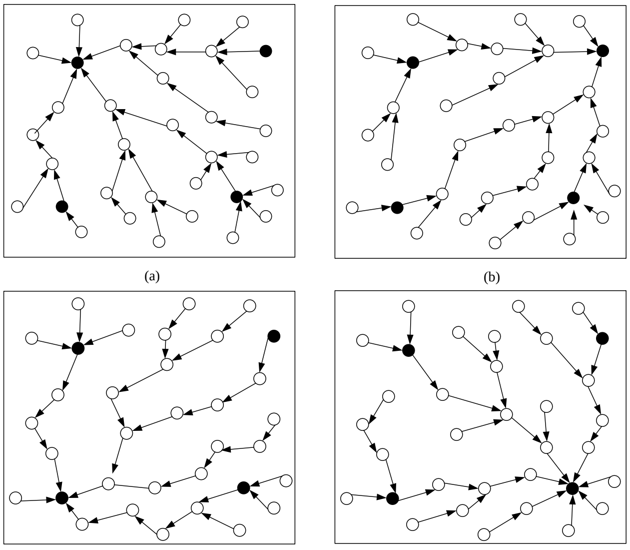


Fig. 4 Four temporary DODAGs rooted at each RN are built during network initialization stage.

N RNs in the network, as shown in Fig. 4. Assume that the position of RN I is (x_I, y_I). The Euclidean distance between two arbitrary RN I and RN J, is calculated using $d_{IJ} = \sqrt{(x_I - x_J)^2 + (y_I - y_J)^2}$. We use h_{IJ} to denote the hop count of RN I to RN J. When an RN joins the DODAG rooted by another RN, it records the position information as well as the hop count associated with that RN. Once an arbitrary RN I has the distance and hop count information of the other $(N-1)$ RNs, its *Coordinates* can be calculated through

$$C_I = \frac{\sum_{J \in \Omega, J \neq I} d_{IJ}}{\sum_{J \in \Omega, J \neq I} h_{IJ}}. \tag{7}$$

After a RN get its *Coordinates* value, it update *CFlag* field as one and broadcast the RFO message with its *Coordinates* value.

During the *Euclidean Distance Calculation* stage, a node is able to get each RN's *Coordinates* value and the hop count to that RN through joining the temporary DODAGs, the information of which is used for the Self-regioning algorithm. The *Coordinates* measurement of RN does not depend on the transmission range of nodes, the advantage of which is the simplicity. However, the accuracy depends on the isotropy property of the network deployment. The network graph of an isotropic network has similar distribution in different directions. So the estimation of distance per hop works more accurately when nodes are deployed evenly.

2) Distributed Self-regioning Strategy: Based on the update of RFO message during the *Euclidean Distance Calculation* stage, a node is able to collect the information of each RN's *Coordinates* value and the hop count associated with that RN. With that information, the distances of a node to all RNs can be calculated. For an arbitrary node *i*, the Euclidean distances to every RN in the network are denoted by $\{d_{iI}\}$, where $I \in \Omega$. The distance between node *i* to RN *I* can be estimated with

$$d_{iI} = C_I h_{iI}, \tag{8}$$

Through comparing its estimate distances to all RNs, a node can find out its closest RN. We use $RegionNumber_{RN}$ to represent the particular region under one RN. There are multiple regions covered by each RN. When the area of one RN is segmented into four regions, whose locations are the upper left segment, lower left segment, upper right segment, and lower right segment, the region number will be *Region* I, II, III, IV, respectively, as shown in Fig. 5. In Fig. 5(a), d_{iA} is the minimum one among the distances to four RNs $(d_{iA}, d_{iB}, d_{iC}, d_{iD})$, so node *i* is inferred to reside in the region under RN *A*, which is denoted by $i \subseteq R_A(I, II, III, IV)$. Then a Self-regioning algorithm using the Law of Cosines identifies which region the node belongs to, and a RC will be assigned to the node accordingly. Fig. 5 shows an example of the distributed Self-regioning algorithm with four RNs in the network. The Law of Cosines is used to compute the values of $\angle\alpha$, $\angle\beta$, and $\angle BAC$ as follows.

$$\angle\alpha = \angle iAB = \arccos \frac{d_{iA}^2 + d_{AB}^2 - d_{iB}^2}{2 d_{iA} d_{iB}} \tag{9}$$

$$\angle\beta = \angle iAC = \arccos \frac{d_{iA}^2 + d_{AC}^2 - d_{iC}^2}{2 d_{iA} d_{iC}} \tag{10}$$

$$\angle BAC = \arccos \frac{d_{AB}^2 + d_{AC}^2 - d_{BC}^2}{2 d_{AB} d_{AC}} \tag{11}$$

Through comparing the value of $|360-\alpha-\beta-BAC|, |\beta-\alpha-BAC|, |\alpha-\beta-BAC|, |\alpha+\beta-BAC|$, node *i* is considered to reside in IV_A because $|\alpha+\beta-BAC|$ gets the minimum value. The region code (RC) of IV_A, refers to the lower right segment of RN *A*, will be assigned to node *i*. The RC is very important in ER-RPL, which is a 8-bits code. The first four bits are represented by the region number and the last four bits are represented by the RN's ID. The ID of an RN is an unique integer. For example, 0000, 0001, 0010, and 0011 can be used to represent *Region* I, II, III, and IV, respectively. If node *i* is located at *Region* IV_A and the ID of RN A is 5, which is 0101 in binary, then node *i*'s RC will be 00110101, as shown in Fig. 5.

With the planned deployment of *N* RNs in the network, a 2D tessellation with *N* cells can be formed. Each cell contains one RN. The number of RNs is a composite number, $r \times t = N$, $\forall r, t \in \mathbb{Z}_+$, where *r* is the row number and *t* is the column number of the tessellation.

Definition 1: The reference node map, $RNM = (RN_{ij})$, is defined as a $r \times t$ matrix with RNs' ID, where $0 \leq i \leq r$, $0 \leq j \leq t$.

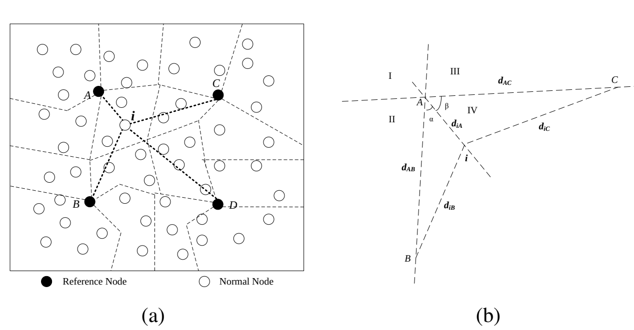


Fig. 5 The distributed Self-regioning algorithm: (a) Step I. (b) Step II.

When four regions are associated with each RN, four RCs can be generated from every element of a Reference Node Map (RNM). In this way, the Region Code Map (RCM), which consists of all RCs in the network, can be computed based on the RNM according to the *Def.* 2.

Definition 2: Based on the RNM, the RCM is defined as a $2r \times 2t$ matrix. $RCM = (RC_{mn})$. Each element of RNM, RN_{ij}, is used to generate four elements of RCM, which are $RC_{2i,2j}$, $RC_{2i,2j+1}$, $RC_{2i+1,2j}$, and $RC_{2i,2j+1}$.

The RNM represents RNs' relative position of the cells. In the network initialization period, nodes obtain the geographical location of each RN to build the RNM. In Fig. 5(a), the ID of RN *A*, *B*, *C*, and *D* in binary are 0101, 0110, 0111, and 1110, respectively. The RNM is

$$RNM = \begin{bmatrix} 0101, 0110 \\ 0111, 1110 \end{bmatrix}, \quad (12)$$

The *Region* I, II, III, and IV are represented by 0000, 0001, 0010, and 0011, respectively. Therefore, the RCM of this network is

$$RCM = \begin{bmatrix} 00000101, 00100101, 00000111, 00100111 \\ 00010101, 00110101, 00010111, 00110111 \\ 00000110, 00100110, 00001110, 00101110 \\ 00010110, 00110110, 00011110, 00111110 \end{bmatrix}, \quad (13)$$

The pseudo-code of the distributed Self-regioning algorithm is shown in Algorithm 1.

3) Rational of Reference Node and Region: It is important to notice that the number of regions per RN covers may vary. There is no overlapping regions, which means one region can only belong to one RN. More regions can be created if more RNs are involved in the Self-regioning algorithm. Besides, if a node has equal distance to two or more RNs, an additional RN needs to get involved in the Self-regioning algorithm. Algorithm 1 includes the equal-distance case and addresses how to select the additional node.

Consider each region is with the same node density. After a node finds out its closest RN, four regions per RN can be formed if the node uses another two RNs for the Self-regioning algorithm. And the shape of each region is approximately a rectangle. Similarly, six regions per RN can be formed if three neighbour RNs are used, and the region shape is close to a hexagon. Octagonal regions will be formed if another four neighbour RNs are used for the Self-regioning algorithm. After identifying the closest RN, if another ε neighbour RNs are involved in the Self-regioning algorithm, each RN can have 2ε regions. In this way, there are $2\varepsilon N$ regions when there exists *N* RNs in the network.

Generally speaking, with more numbers of RNs used in the Self-regioning algorithm, there will be more regions in the network. ER-RPL explores the optimal P2P route among only a subset of regions in the network. When the network is segmented to more regions, each region will have a smaller size and there tends to be a smaller subset of nodes participating in the route discovery on average. In this way, a greater reduction of control overhead can be achieved with an increased number of RNs used in the Self-regioning algorithm. The impact of the number of regions on the reduction of overhead that can be achieved by ER-RPL is analysed in Section IV.

Algorithm 1: Distributed Self-regioning Algorithm.

Data: $\{d_{iI}\}$, $\{d_{IJ}\}$, where $i \in N$, RN $I, J \in K$, and $I \neq J$. RN *A*, *B*, *C*, and *D* is $RN_{i,j}$, $RN_{i+1,j}$, $RN_{i,j+1}$, and $RN_{i+1,j+1}$ element of RNM, respectively.

if $\{d_{iA}\} = min\,\{d_{iI}\}$ **then**
  $i \subseteq R_A(I, II, III, IV)$;
  if $(d_{iA} + d_{iB} > d_{AB})$ **then**
    if $(d_{iA} + d_{iC} > d_{AC})$ **then**
      return $compare\_angles(\angle iAB, \angle iAC, \angle BAC)$
    else if $\angle iAB > \angle BAC$ **then**
      return III_A
    else
      return IV_A
  else if $(d_{iA} + d_{iC} > d_{AC})$ **then**
    if $\angle iAC > \angle BAC$ **then**
      return II_A
    else
      return IV_A
  else
    return IV_A
else if $\{d_{iA}\}$=$\{d_{iB}\}$=*min*$\{d_{iI}\}$ **then**
  $i \subseteq R_A(II, IV) \bigcup R_B(I, III)$;
  if $d_{iC} < d_{iD}$ **then**
    if $\angle iAC > \angle BAC$ **then**
      return II_A
    else
      return IV_A
  else if $\angle iAC > \angle BAC$ **then**
    return I_B
  else
    return III_B
else if $\{d_{iA}\}$=$\{d_{iB}\}$=$\{d_{iC}\}$=*min*$\{d_{iI}\}$ **then**
  return IV_A $(d_{DA} > d_{DB}, d_{DA} > d_{DC})$
else
  return IV_A or III_B or II_C or I_D
Function *compare_angles*(α, β, γ)
  δ=min $\{|360 - \alpha - \beta - \gamma|, |\beta - \alpha - \gamma|, |\alpha - \beta - \gamma|, |\alpha + \beta - \gamma|\}$
  if $\delta = |360 - \alpha - \beta - \gamma|$ **then**
    return *Region* I
  else if $\delta = |\beta - \alpha - \gamma|$ **then**
    return *Region* II
  else if $\delta = |\alpha - \beta - \gamma|$ **then**
    return *Region* III
  else
    return *Region* IV

D. Reactive P2P Route Discovery

In this section, we present the routing design of the proposed protocol. ER-RPL comprises three main components for the reliable and energy-efficient data delivery. 1) During the region-based route discovery, a submatrix of the region code map (RCM) is computed based on the source node's region code (SRC) and the destination node's region code (DRC). Upon receiving a route request, a node can make a decision whether it should join or ignore a route discovery process by checking its RC's existence in that submatrix of the RCM. Then the temporary DODAG is formed among only a portion of nodes in the network. 2) The Region-to-Region (R2R) routing without route discovery is designed as an enhancement. The R2R route refers that there is an available route for any node in one region to any node in another region. The R2R route needs to go through an intermediate

Type	Start Point Address	End Point Address	Sequence Num	Lifetime
Source Region Code	Destination Region Code	Accumulated ETX	Hop	

Fig. 6 Packet structure of MRO.

node, which is located in one of these two regions. The R2R routes are enable during the a route discovery for one source-destination node pair, which are the routes between nodes in the source node's region and nodes in the destination node's region. Besides, the route adjustment on-the-fly is implemented to shorten the R2R routes. 3) Due to the limited transmission range and sparse node deployment, nodes in some regions maybe not able to provide the connectivity to their neighbour regions, which is defined as the dead zone problem. We propose an Adaptive Region Selection (ARS) solution for the dead zone problem.

1) Region-based P2P Route Discovery: Given a source-destination node pair, a submatrix of RCM is generated and defined as the IRCM, which composes a subset of the RCs in the network. The IRCM is used to determine whether a node should join the route discovery or not. Nodes, whose RC is the element of IRCM, are supposed to participate the route discovery for the source-destination node pair. Compared to the network size, ER-RPL selects a smaller area for the route discovery with the help of IRCM. The size of that area is denoted by g, where $g \leq G$. The regions of the source and destination nodes are located in the diagonal of the selected area. The shortest (most reliable) routing path for the source-destination pair will be chosen for the data delivery. Assume that RC_{m_s,n_s} and RC_{m_d,n_d} denote the SRC and DRC in the RCM, respectively. Then the IRCM is a $(|m_s-m_d|+1)\times(|n_s-n_d|+1)$ matrix, and the $IRCM(u,v)$ has equal entry to $RCM(u+min(m_s,m_d), v+min(n_s,n_d))$, where $0 \leq u \leq (|m_s-m_d|+1)$, $0 \leq v \leq (|n_s-n_d|+1)$.

With the RCM shown in Eq. (13). If the SRC is 00110101 and the DRC is 00011110, the *IRCM* is

$$IRCM = \begin{bmatrix} 00110101, 00010111 \\ 00100110, 00001110 \\ 00110110, 00011110 \end{bmatrix}. \quad (14)$$

The Message Request Object (MRO) is designed as the control message for ER-RPL, the packet structure of which is shown in Fig. 6. The *Start Point Address* and *End Point Address* record the source and destination IP addresses, respectively. The *Lifetime* indicates how long the temporary DODAG will exist. But if the source node receives a hint from the upper layer, the lifetime can be extended or shortened. The end-to-end cost in terms of ETX is recorded in the *Accumulated ETX* field. The hop count towards the destination is stored in the *Hop* field. Different types of MRO messages are distinguished through the *type* field. MRO(0) denotes the Message Request Object with type 0. MRO(1) denotes the Message Reply Object with type 1. MRO(2) denotes the Region-to-Region MRO with type 2. MRO(3) denotes the Region-based DAG MRO with type 3.

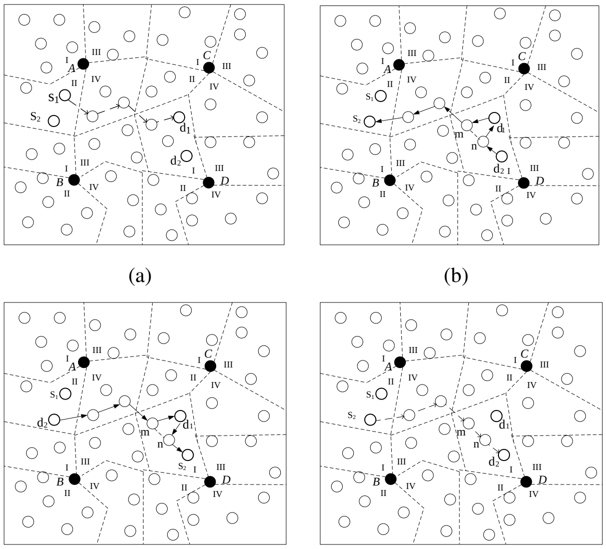
For a P2P route request, the source node will send a MRO(0) through the existing DAG (via the root) to the destination node. MRO(0) records the accumulated cost and hop count along this path. Upon receiving MRO(0), the destination node can decide whether to use this existing route or explore a better route based on the route constraints[1]. If the existing route's cost is within the route constraints, MRO(1) is sent from the destination node to the source node. Otherwise, this route is deemed as unsatisfactory. Then the destination node will check whether the routes between the source node's region and the destination node's region are available or not. If the R2R route has been enabled, there exists a route from the destination node to the source node through an intermediate node. The source node will send the MRO(2) to the intermediate node through the best quality path. That intermediate node has the routing path to any node in the source node's region, and it will forward the MRO(2) to the source node through the optimal routing path. Once the source node receives the MRO(2) , the reverse route will be used for data delivery. However, if the existing route is unsatisfactory and there is no R2R route available, the destination node will initiate the temporary DODAG construction as the root with MRO(3), in which the SRC and the DRC are piggybacked.

Once a node receives the MRO(3) from its neighbour nodes, a node computes the IRCM based on the SRC and DRC first. Then a node checks whether its RC exists in the IRCM or not. If it is so, the node will join the route discovery with the following steps. 1) The node temporarily records the sender's ID with its end-to-end cost and hop count corresponding to the destination node in its local table. 2) The node chooses the neighbour node that gives the minimum accumulated cost to the destination as its preferred parent. 3) The node computes its cost towards the destination, update its hop count and broadcast the MRO(3) to its neighbours. In this way, a temporary DODAG is constructed in the way that nodes always select the best quality path towards the destination. During route discovery, nodes in a subset of regions choose the best route to the destination given particular routing metrics. Once the source node receives the route discovery message, it will send data to the destination node through the optimal route discovered by the DODAG construction.

Assume that *M* denotes the total number of regions, which is equivalent to ($2r \times 2t$), and *K* denotes the number of the subset of regions that involve in the route discovery, which is equivalent to $(|m_s-m_d|+1)\times(|n_s-n_d|+1)$. The complexity of our proposed protocol is $\mathcal{O}(M+M+K)$, where $K < M$. As *N* is very small ($N \ll n$), the complexity of our algorithm is low, so that our proposed algorithm is comparable with the existing protocols, such as RPL and P2P-RPL.

2) Region-to-Region Routing without route discovery: During the region-based route discovery process, the best quality path towards the destination is selected by each node in the regions identified by the IRCM. Meanwhile, nodes in the same region with the source or destination node, send their routing information (hop-by-hop or source routing) to the destination node. In this way, the downwards route from the temporary root (destination node in this routing pair) to all nodes in the source node's region and destination node's region

[1]The work in [31] describes how to measure the P2P routing metrics in detail.



(a) (b)

(c) (d)

Fig. 7 Region-to-Region routing without route discovery. The solid arrow is used to indicate the route which MRO(2) travels along. The dash arrow indicate the route for data delivery. (a) The routing between *Region* II_A and *Region* I_D is enabled. (b) R2R routing is used for s_2 and d_2. (c) The intermediate node, d_1, can locates either in s_2's region or d_2's region. (d)R2R routing with Route Adjustment on-the-fly.

will be enabled. The DODAG root, which is the destination node in this P2P pair, is used as an intermediary to support the communication between the source node's region to the destination node's region. Besides, in order to enable the R2R routing, every node keeps a R2R table, each row of which stores one node id, two RCs and the lifetime for this row. In order to keep the R2R table small, only nodes with SRC or DRC create the entry. Upon receiving the MRO(3) that carries the SRC and DRC, a node will store the SRC and DRC pair corresponding to the destination node (temporary root) in its R2R table. Upon receiving a P2P routing request, the destination node will examine its R2R routing table before the route discovery. If the route to source node's region is indicated as available, the destination node will send a reply MRO(2) via the corresponding DODAG root to the source node.

Fig. 7 illustrates an example for the R2R routing with ER-RPL. As shown in Fig. 7(a), the routing between nodes in *Region* II_A and *Region* I_D are enabled during the route discovery from destination node d_1 to source node s_1. All nodes in *Region* II_A and *Region* I_D update their route information to node d_1 and store the information of (d_1, I_D, II_A) with its expire time in this R2R table. Node d_1 will temporarily record the routes to all nodes in *Region* IV_A and *Region* I_D. When there is another source-destination node pair, $s_2 - d_2$ in Fig. 7(b), the destination node d_2 checks its R2R table, in which the entry (d_1, I_D, II_A) indicates that there is an existing route to nodes in *Region* I_D via the intermediary node d_1. Instead of route discovery, the destination node d_2 replies with a MRO(2) to the node d_1. With the routing information to s_2, d_1 will forward the MRO(2) to node s_2. Once node s_2 receives the MRO(2), the reverse route of MRO(2) will be used for data delivery.

The R2R routing without route discovery can generally be applied no matter the intermediate node is located in the source or destination node's region. The example can be found in Fig. 7(b) and 7(c). In both cases, the entry for the source node region in R2R is (d_1, I_D, II_A), and node d_1 performs as the intermediate node. Although the routing paths from the source and destination node to the intermediate node are the best quality paths, the R2R route may not be the optimal one due to the existence of the intermediary. There occurs some extra hops within one region when adopting R2R routing. But it is lesser significant when the route tends to be long between the source region and the destination region. With a fixed network size, the number of nodes in each region is lesser when there are more numbers of regions. In this way, the sub-optimal route will be closer to the optimal route and the R2R table size become smaller. However, to get more regions in the network, it requires more number of RNs leading to an increased computation complexity.

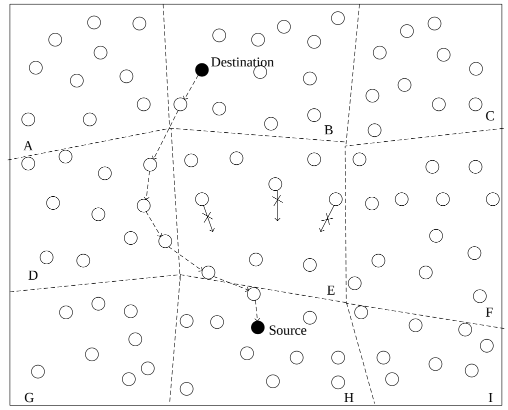


Fig. 8 Illustration of Dead Zone Problem.

Route Adjustment On-the-fly: We propose a route adjustment "on-the-fly" scheme to shorten the R2R route and alleviate the sub-optimal routing path effect. When a node receives or overhears a MRO(2), it always chooses the one that provides smaller hop count as the next hop to the destination node. Without bringing any additional control message, this scheme allows a node to choose its preceding node instead of the one it receives the MRO(2) from. There is no additional control messages in this scheme.

In Fig. 7(b), node *m* overhears the MRO(2) from node *n*, whose hop count to d_2 is one. After node *m* receiving the MRO(2) from node d_1, whose hop count to d_2 is two. Node *m* will choose node *n* as the next hop, so that its hop count to d_2 is two. Comparing 7(b) with 7(d), the hop count is effectively reduced with this scheme.

3) Adaptive Region Selection for the Dead Zone Problem: In the previous section, we assume that the density of nodes in each region is approximately a constant. However, due to the irregular node deployment, the node density may vary from region to region. Let ρ_A denote the node density of *Region* A. It is known that a sufficient density is a critical requirement to ensure almost surely (a.a.s.) network connectedness [32].

Let us consider *n* nodes placed uniformly and independently in a 2D network area. For the sake of simplicity, the region shape is square with a size of $L_I \times L_I$. Then each region is considered to be divided into square size cells, each edge of which is $\sqrt{2}R/2$. For region I, the node density required for a.a.s. connectedness is defined in the following theorem:

Theorem 1: $\rho_I \geq \frac{\alpha ln L_I}{r^2}$, for some $\alpha > 0$, $lim_{L_I \to \infty} P_{conn}(L_I) = 1, P_{conn}(L_I)$ denotes the probability that *Region* I is connected.

This theorem provides the necessary conditions to ensure asymptotically a.a.s. region connectivity. A similar theorem is proved in [27]. So the proof is omitted here. Although this theorem theoretically provides the required density for the network connectedness, an effective routing solution to check the connectedness is also needed in practice.

As we know, the region connectivity is impaired if the network density is too low. When some regions are with sparse node distribution, these regions may not able to provide the connectivity for their neighbor regions. The dead zone problem arises when the P2P route is discovered among such regions. In Fig. 8, when the destination node needs to discover a route to the source node, nodes in *Region* B, E, and H will participate in the route discovery process. Due to the limited transmission range and sparse node deployment, nodes in *Region* E are not able to establish a route for *Region* B and *Region* H. Therefore, *Region* E is considered as a dead zone for *Region* B and *Region* H. However, the routing path can be successfully provided, if nodes in *Region* D join the route discovery process.

We propose an Adaptive Region Selection (ARS) scheme to provide an efficient solution for the dead zone problem. During the network initialization stage, nodes update their RCs and one hop neighbor information through their preferred parent nodes to the root, so that the root node has a global knowledge of the network topology. A similar approach is defined in RPL [3], where the Destination Advertisement Object (DAO) is used to propagate upwards along the DAG to the root, so that the downwards route from the root is enabled. The root usually has a strong computation capability and large memory space, which can store RNs' *Coordinates*, nodes' RCs and their one hop neighbour list. Upon receiving the MRO(0) for a P2P routing request, the root will compute the IRCM and perform a neighbor list checking, so that it can determine whether the source node and destination node are reachable within the regions from IRCM. If the root detects that those regions cannot provide the connectivity, it will use a flag to indicate the dead zone problem in MRO(0). Upon receiving the MRO(0) with the dead zone indication, the destination node will piggyback this information in the route discovery message (MRO(3)). Instead of IRCM, the Expanded IRCM (EIRCM) will be used in this case.

Definition 3: Assume that SRC and DRC are the RC_{m_s,n_s} element and the RC_{m_d,n_d} element in the RCM, respectively, where the RCM is an $2r \times 2t$ matrix. The EIRCM is defined as a $(m_{dd} - m_{ss} + 1) \times (n_{dd} - n_{ss} + 1)$ submatrix of RCM, where m_{ss}, n_{ss}, m_{dd}, and n_{dd} can be computed according to Eq. (15). $EIRCM(u, v)$ has entry equal to $RCM(m, n)$, where $m = u + m_{ss}$, and $n = v + n_{ss}$.

$$\begin{aligned} m_{ss} &= \begin{cases} min(m_s, m_d) - 1, & \text{if } min(m_s, m_d) \neq 0 \\ 0, & \text{otherwise}, \end{cases} \\ n_{ss} &= \begin{cases} min(n_s, n_d) - 1, & \text{if } min(n_s, n_d) \neq 0 \\ 0, & \text{otherwise}, \end{cases} \\ m_{dd} &= \begin{cases} max(m_s, m_d) + 1, & \text{if } max(m_s, m_d) \neq m \\ m, & \text{otherwise}, \end{cases} \\ n_{dd} &= \begin{cases} max(n_s, n_d) + 1, & \text{if } max(n_s, n_d) \neq n \\ n, & \text{otherwise}. \end{cases} \end{aligned} \quad (15)$$

During the region-based route discovery process with ARS, a node will compute EIRCM instead of IRCM after receiving MRO(3) with the dead zone problem indication. If a node finds its RC is an element in the EIRCM, it will join this temporary DODAG for the P2P route discovery. In this way, more regions are involved in the route discovery process so that it greatly increases the chance for identifying an available routing path from the source node to the destination node.

IV. Model Validation

The trickle algorithm is used in RPL, P2P-RPL and ER-RPL to achieve both rapid propagation and low maintenance overhead. Three key parameters are used to define the interval of broadcasting control messages [12]; the minimum time interval size I_{min}, the maximum time interval size I_{max}, and the redundancy counter *c*. The current communication interval is denoted as *I*. When a node starts a trickle timer, it resets *c* to 0 and the control messages update time, which is denoted by *t*, to a random value between $I/2$ to I. Whenever a node hears a transmission that is "consistent" [2], it increments the counter *c*. The trickle algorithm doubles the current communication interval, *I*, when the interval *I* expires. Any "inconsistent" message will cause trickle timer to reset. Until the interval length is greater than I_{max}, *I* is set to be I_{max}. In other words, the time interval increases during the network convergence time. When the interval *I* reaches I_{max}, the transmission rate of control messages, f_c, follows

$$f_c \approx \frac{1}{I_{min} 2^{I_{max}}}. \quad (16)$$

Assume that the number of traffic flows is denoted by λ. In P2P-RPL, all nodes will participate in the route discovery. When λ traffic flows require the route discovery, where $\lambda < n(n-1)/2$, the control overhead of P2P-RPL for time *t* is

$$O_{p2p-rpl} = n\lambda f_c t = \frac{n\lambda t}{I_{min} 2^{I_{max}}}. \quad (17)$$

The route discovery may not be necessary under certain circumstances. Nodes can have direct communication to their one-hop neighbours. By assuming that nodes' one-hop neighbours number is $\pi R^2 \rho$, the ratio of one-hop data delivery is

$$\varphi_{nb} \approx \frac{C_n^1(\pi R^2 \rho - 1)}{n^2} = \frac{\pi R^2 \rho - 1}{n}. \quad (18)$$

[2] A control message (DIO, P2P-DRO, and MRO(3) in RPL, P2P-RPL, and ER-RPL, respectively) from a sender with a lesser DODAG rank that does not cause any changes to the recipient's parent set, preferred parent, or rank is considered consistent. Otherwise, the control message is inconsistent.

Nodes' routing tables usually contain some valid routes. The ratio of the number of valid routes to the number of nodes in the network is denoted as φ_{rt}. The value of φ_{rt} is affected by many factors, such as the routing table size, traffic flows, and the routing lifetime. The number of one-hop neighbours can be approximately considered as the number of valid routing entries due to the limited memory space of nodes. With the i.i.d. source-destination node pairs, the control overhead of P2P-RPL for λ traffic flows during time t is

$$O_{p2p-rpl} \approx \frac{(1-\varphi_{rt})n\lambda t}{I_{min}2^{I_{max}}} \approx \frac{(1-\varphi_{nb})n\lambda t}{I_{min}2^{I_{max}}}. \tag{19}$$

In ER-RPL, only a subset of nodes in the networks will participate in the route discovery. For simplicity, we consider a dense network, in which the node density, ρ, of each region is the same. We also assume the route discovery is needed to support the communication between two arbitrary nodes in the network. For a network containing k^2 regions, the ratio of nodes that participate in route discovery and consume overhead to total nodes is denoted as $\varphi(k, \rho, n)$, where $0 < \varphi(k, \rho, n) \leq 1$. For a $k \times k$ RCM, the SRC is $RCM(m, n)$, and the DRC is $RCM(a, b)$, where $a \in [0, k-1], b \in [0, k-1]$. Then IRCM is a $(|m-a|+1)\times(|n-b|+1)$ matrix. In this case, the value of $\varphi(k, \rho, n)$ is $(|m-a|+1)(|n-b|+1)/k^2$. When the source node and destination node are i.i.d., $\varphi(k, \rho, n)$ will no longer depend on ρ and n as the number of traffic flows λ tends to ∞, which is represented by $\varphi(k)$. In this case, we can have

$$\varphi(k) \approx \frac{\sum_{a=0}^{k-1}\sum_{b=0}^{k-1}\sum_{m=0}^{k-1}\sum_{n=0}^{k-1}(|m-a|+1)(|n-b|+1)}{k^6}. \tag{20}$$

Similarly, it is not always necessary to discover routes because of the one-hop neighbours and valid routes in the routing table. The expectation value of $\varphi(k, \rho, n)$ is $\varphi(k)$, which does not consider the valid routing entries in the routing table. So we can get the following,

$$\varphi(k, \rho, n) \approx \varphi(k) + \varphi_{rt} \approx \varphi(k) + \varphi_{nb}. \tag{21}$$

So theoretically, the control overhead for ER-RPL based on the previous conditions is

$$O_{er-rpl} \approx \frac{\varphi(k, \rho, n)n\lambda t}{I_{min}2^{I_{max}}}. \tag{22}$$

V. Network Simulation

A. Experimental Setup

We implement RPL, P2P-RPL and ER-RPL from scratch on Network Simulator 3 (NS3) [33]. The error erasure channel is used to model the lossy environment. Each wireless link is assigned with a random probability p_{ij}, where $0.3 \leq p_{ij} \leq 0.8$. Many research studies have provided solutions to measure the wireless link quality [34], [35], which is out of the scope of this paper. We assume that the link layer will handle this issue. Four RNs are used in this study. Each RN will have four regions with the distributed Self-regioning algorithm. Both symmetric links ($p_{ij} = p_{ji}$) and asymmetric links ($p_{ij} \neq p_{ji}$) are considered, which are denoted by *s* and *as*, respectively, in the performance analysis section. For instance, for a particular protocol, ER-RPL$_s$ and ER-RPL$_{as}$ denote the routing performance with symmetric and asymmetric links, respectively. RPL is in non-storing mode. Our simulation studies are conducted with the IEEE 802.11. It is worth to highlight that IEEE 802.11 is usually not considered as the best candidate for LLNs, while IEEE 802.15.4 [4] is viewed as the optimal choice for LLNs. But the routing protocols' general behavior, such as path-length, packet delivery, control overhead, etc, can be concluded from the simulation study with IEEE 802.11 [16], especially for networks with a low data rate. A detail performance comparison of IEEE 802.11 and IEEE 802.15.4 can be found in the work of [36].

Table II Parameter setting for our simulation.

Parameter	Value
Number of nodes	100
Number of reference nodes	4
Retransmission limit	5
Communication range	35m
Packet size	512 bytes
Traffic rate	4 pkt/s, CBR flow
Routing metric	ETX
Transmitter electronics (E_{elec})	50nJ/bit
Transmit amplifier (ϵ_{amp})	100pJ/bit/m^2
The Minimum time interval size (I_{min})	50ms
Redundancy constant (c)	∞

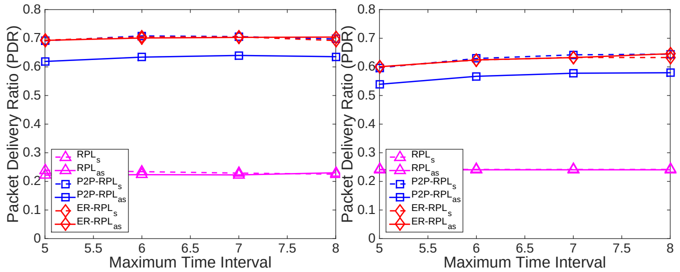
We focus on the P2P routing performances in two scenarios. In Scenario I, the maximum time interval I_{max} are varied from 5 to 8 and traffic flows are set to 30. In Scenario II, the number of P2P traffic flows are varied from 10 to 40, and I_{max} is set to 8. In both scenarios, two types of node deployment strategies are studied. Without loss of generality, 100 nodes are randomly deployed in a $180m \times 180m$ network area with the random planar deployment strategy, and we also study the 10×10 grid deployment with some randomness for the constant node density scenario. The node's position is $(20b \pm \Delta x, 20r \pm \Delta y)$, where $-5 \leq \Delta x \leq 5, -5 \leq \Delta y \leq 5$. Assume that the node ID is U, the value of b and r are obtained through $U = 10b + r$, where b, r is $\mathbb{N}$, and $r < 10$.

In this simulation study, the source and destination node pair is randomly selected for each traffic flow. Each traffic flow lasts for 90 seconds and then another P2P flow starts. In this way, the number of concurrent traffic flows are kept the same but there are different source and destination routing pairs generated during the simulation. Other parameters are displayed in Table II. The simulation time is 1000s. A different random topology is generated in each run and each data of the results is the average value from ten runs.

B. Performance Metrics

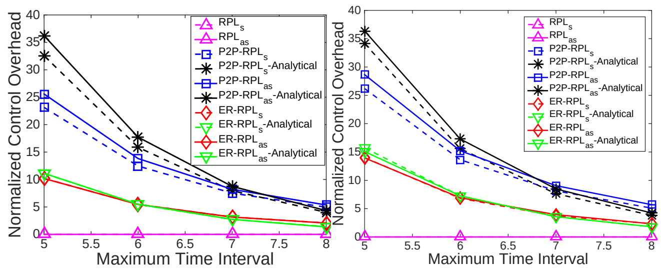
The following metrics are adopted to evaluate the performance of the proposed protocol and the benchmarks.

1) *Packet delivery ratio* refers to the ratio of the number of packets successfully delivered to the destination nodes to the number of packets generated by the source nodes. This metric illustrates the routing reliability.
2) *Normalized routing control overhead* refers to the ratio of the number of control messages to the number of data successfully delivered to the destination nodes.



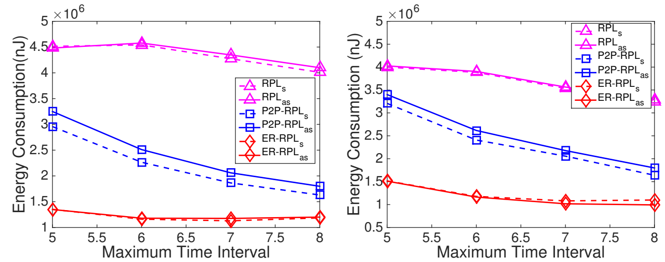
(a) (b)

Fig. 9 Scenario I: The PDR comparison. (a) grid with randomness deployment (ρ is a constant). (b) random planar deployment (ρ is not a constant).



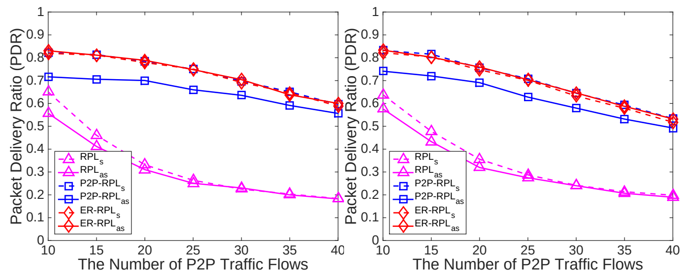
(a) (b)

Fig. 10 Scenario I: Control overhead comparison of simulation vs theoretical results. (a) grid with randomness deployment (ρ is a constant). (b) random planar deployment (ρ is not a constant).



(a) (b)

Fig. 11 Scenario I: Energy consumption per successful data delivery comparison. (a) grid with randomness deployment (ρ is a constant). (b) random planar deployment (ρ is not a constant).



(a) (b)

Fig. 12 Scenario II: The PDR comparison. (a) grid with randomness deployment (ρ is a constant). (b) random planar deployment (ρ is not a constant).

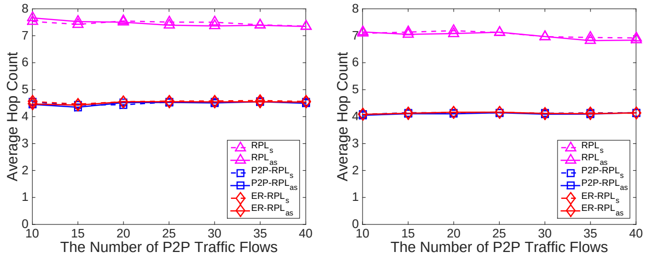
3) *Energy consumption per data successfully delivered* is the ratio of the total energy consumption to the number of data packets that are successfully delivered to the destination nodes. The total energy consumption of the network includes the energy spent for all phases of the network during the simulation.
4) *Average hop count* refers to the average number of hops for the discovered route between the source nodes and the destination nodes.
5) *Average end-to-end delay* includes all possible delay during data transmission due to transmission time, re-transmission caused by collision and queuing time.

C. Performance Evaluation

As shown in Fig. 9, ER-RPL improves the packet delivery ratio (PDR) by 150% compared to RPL. ER-RPL$_s$ achieves a very close performance to P2P-RPL with symmetric links (P2P-RPL$_s$), which is the optimal value with symmetric links. It shows that although only partial nodes are involved in the P2P route discovery process, ER-RPL can still choose the nearly optimal route to provide reliable routing, which is one of our goals in this work. With the region-based route discovery, ER-RPL$_{as}$ can maintain the performance and outperform P2P-RPL with asymmetric links (P2P-RPL$_{as}$) by around 10%. In P2P-RPL, the P2P route is discovered throughout the whole network. Therefore, the optimal route from the destination node to the source node will be selected by P2P-RPL. The PDR degrades in P2P-RPL if the wireless link becomes asymmetric. Because in P2P-RPL, the temporary DODAG is rooted at the source node, so that the route for data delivery may not be the optimal one from the source to the destination under asymmetric links. Different from P2P-RPL, whose temporary DODAG is rooted at the source node, ER-RPL forms the temporary DODAG at the destination node, so the best route from the source to destination is always used.

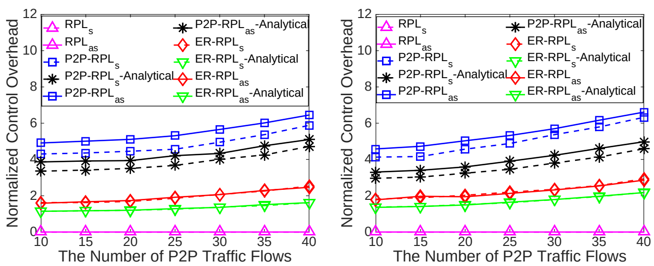
Fig. 10 shows that the routing control overhead for route discovery decreases with the increment of the maximum time interval size (I_{max}). ER-RPL achieves average 60% less control overhead compared to P2P-RPL. The control overhead of P2P-RPL$_{as}$ is larger than P2P-RPL$_s$ due to fewer packets received by the destinations. RPL uses the pre-established route for the data delivery, so the route discovery is not required and we consider that no additional control messages occur in this case. Fig. 10 also depicts the theoretical results with Eq. (19) and (22). With a short maximum time interval, the network reaches the convergence stage quickly. So the effect of high frequency during the network initialization stage is negligible. Meanwhile, due to the valid routing entries, the overhead is slightly lower in the simulation than the theoretical results, which regards the number of one-hop neighbours as the number of valid routes. However, with the increment of I_{max}, it takes more time for the network to converge, thus it occurs higher overhead.

Fig. 11 depicts the energy consumption decreases with the decreasing frequency of control messages. ER-RPL achieves around 66% and 60% energy conservation compared to RPL on average, as shown in Fig. 11(a) and Fig. 11(b), respectively, because the longer routes in RPL consume more energy. The impact of the frequency of control message is more significant on P2P-RPL than ER-RPL. P2P-RPL disseminates the control messages throughout the network, and the large amount of control messages constitute a significant portion of the total energy cost. So the variation of the frequency of control messages leads to a obvious difference on the energy consumption performance in P2P-RPL. With a great reduction



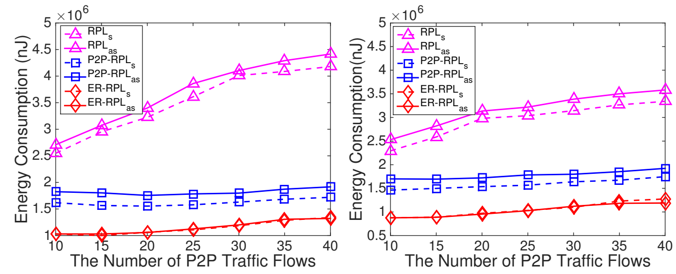
(a) (b)

Fig. 13 Scenario II: Hop count comparison.. (a) grid with randomness deployment (ρ is a constant). (b) random planar deployment (ρ is not a constant).



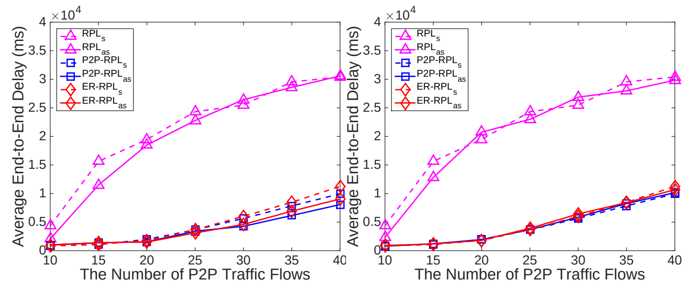
(a) (b)

Fig. 14 Scenario II: Control overhead comparison of simulation vs theoretical results. (a) grid with randomness deployment (ρ is a constant). (b) random planar deployment (ρ is not a constant).



(a) (b)

Fig. 15 Scenario II: Energy consumption per successful data delivery comparison. (a) grid with randomness deployment (ρ is a constant). (b) random planar deployment (ρ is not a constant).



(a) (b)

Fig. 16 Scenario II: End-to-End delay comparison. (a) grid with randomness deployment (ρ is a constant). (b) random planar deployment (ρ is not a constant).

of the control overhead, ER-RPL achieves a significant energy conservation compared to P2P-RPL. The energy consumption of ER-RPL only changes slightly with a variation of the frequency of control messages. Apart from the frequency of the control messages, many other factors also affect the energy performance, for instance, the size of data packets and control messages.

Fig. 12 shows the PDR performance drops as the traffic increases. In RPL, the root becomes the bottleneck of the network when the traffic is heavy, resulting in a serious performance drop. The dead zone problem may take place in the random planner deployment. But ER-RPL still achieves a nearly optimal value, as represented by the PDR performance of P2P-RPL$_s$. It shows that the proposed ARS is an efficient solution for the dead zone problem.

Fig. 13 depicts that the average hop count of P2P routes selected by ER-RPL is very close to P2P-RPL, which is 40% less than that of RPL. In RPL, the data packets are delivered through the pre-established route via the root, thus it suffers from long route. The average hop count is slightly more in Fig. 13(a) compared to Fig. 13(b). Because the node density of the grid with randomness deployment is lower than that of the random planar deployment in this study.

Fig. 14 depicts that the control overhead increases with the increasing of P2P traffic flows. In Fig. 14(a), ER-RPL achieves about 59%, 66% less control overhead than P2P-RPL with symmetric and asymmetric link, respectively. In Fig. 14(b), ER-RPL$_s$ and ER-RPL$_{as}$ achieve about average 55%, 58% less overhead compared with P2P-RPL$_s$ and P2P-RPL$_{as}$, respectively. This is because more regions join the route discovery with the ARS to solve the dead zone problem of the random planar network. Results from the simulation study are close to that from the theoretical analysis. The subtle difference is due to some factors which are considered in the simulation model but are not considered in the analytical model (Eq. 19 and 22), such as the randomness of control frequency in the initialization stage, the routing lifetime, the routing table, etc. R2R routing further reduces the control overhead in ER-RPL, which is reflected by the smaller gap between the theoretical and simulation results than P2P-RPL.

Fig. 15 shows that the energy consumption increases with the increase of P2P traffic flows. ER-RPL achieves a great energy conservation compared to RPL and P2P-RPL. In RPL, all traffic flows are delivered through the root. The area near the root will become congested with high traffic, and a large amount of packets get dropped in that case. So the increase of P2P traffic flows has a more significant impact on RPL than ER-RPL and P2P-RPL. ER-RPL and P2P-RPL shows a more robust energy performance with the increase of network traffic flows. In addition, ER-RPL has similar energy performances with symmetric and asymmetric links, while P2P-RPL$_{as}$ consumes more energy than P2P-RPL$_s$ due to a less amount of successfully delivered data packets.

Fig. 16 depicts that the end-to-end delay increases as the traffic increases. ER-RPL achieves a similar performance as P2P-RPL. RPL suffers from long delay due to the longer route used for the data delivery. Additionally, the increasing traffic load has a significant impact on the delay performance of RPL, because of the bottleneck in the network.

VI. Conclusion

This paper proposes ER-RPL, which is a distributed energy-efficient region-based routing protocol. ER-RPL is a hybrid of proactive and reactive routing protocol, and it exploits the region information of networks with stationary nodes. The proposed routing protocol provides a generic traffic pattern support, which inherits the DODAG network structure optimized for MP2P traffic pattern from RPL, and discovers the best quality routing path within the regions between the source and destination nodes for P2P communication. In general, we provide an efficient routing solution to achieve both the reliability and the energy-efficiency simultaneously. Theoretical analysis of control overhead reduction is provided. Our results also indicate that ER-RPL achieves a great reduction of overhead, and it is robust to the wireless channel conditions. Furthermore, simulation results depict that ER-RPL can select the nearly optimal reliable routes with a great energy conservation. As this work only focuses on the static networks, in our future work, we plan to extend our work into the mobile networks scenario.

REFERENCES

[1] K.-C. Chen and S.-Y. Lien, "Machine-to-machine communications: Technologies and challenges," *Ad Hoc Networks*, vol. 18, pp. 3–23, 2014.

[2] Z. Sheng, S. Yang, Y. Yu, A. Vasilakos, J. Mccann, and K. Leung, "A survey on the ietf protocol suite for the internet of things: Standards, challenges, and opportunities," *IEEE Wireless Communications*, vol. 20, no. 6, pp. 91–98, 2013.

[3] O. Gaddour and A. Koubâa, "Rpl in a nutshell: A survey," *Computer Networks*, vol. 56, no. 14, pp. 3163–3178, 2012.

[4] B. Pavkovic, A. Duda, W.-J. Hwang, and F. Theoleyre, "Efficient topology construction for rpl over ieee 802.15. 4 in wireless sensor networks," *Ad Hoc Networks*, vol. 15, pp. 25–38, 2014.

[5] J. W. Hui and D. E. Culler, "Ipv6 in low-power wireless networks," *Proceedings of the IEEE*, vol. 98, no. 11, pp. 1865–1878, 2010.

[6] J. Tripathi, J. C. de Oliveira, and J.-P. Vasseur, "A performance evaluation study of rpl: Routing protocol for low power and lossy networks," *In the Proceedings of the 44th Annual IEEE Conference on Information Sciences and Systems (CISS)*, pp. 1–6, 2010.

[7] E. Baccelli, M. Philipp, and M. Goyal, "The p2p-rpl routing protocol for ipv6 sensor networks: Testbed experiments," *In the Proceedings of the 16th IEEE International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*, pp. 1–6, 2011.

[8] J. Martocci, P. Mil, N. Riou, and W. Vermeylen, "Building automation routing requirements in low-power and lossy networks," 2010.

[9] A. Brandt and J. Buron, "Home automation routing requirements in low-power and lossy networks," 2010.

[10] K. Pister, P. Thubert, S. Dwars, and T. Phinney, "Industrial routing requirements in low-power and lossy networks," Tech. Rep., 2009.

[11] P. Levis, A. Tavakoli, and S. Dawson-Haggerty, "Overview of existing routing protocols for low power and lossy networks," *Internet Engineering Task Force, Internet-Draft draftietf-roll-protocols-survey-07*, 2009.

[12] P. A. Levis, N. Patel, D. Culler, and S. Shenker, *Trickle: A self regulating algorithm for code propagation and maintenance in wireless sensor networks*, Computer Science Division, University of California, 2003.

[13] A. Brachman, "Rpl objective function impact on llns topology and performance," in *Internet of Things, Smart Spaces, and Next Generation Networking*,Springer, pp. 340–351, 2013.

[14] M. Goyal, E. Baccelli, M. Philipp, A. Brandt, and J. Martocci, "Reactive discovery of point-to-point routes in low-power and lossy networks," Tech. Rep., 2013.

[15] C. Perkins, E. Belding-Royer, and S. Das, "Ad hoc on-demand distance vector (aodv) routing," Tech. Rep., 2003.

[16] J. Yi, T. Clausen, and Y. Igarashi, "Evaluation of routing protocol for low power and lossy networks: Loadng and rpl," *In the Proceedings of IEEE Conference on Wireless Sensor (ICWISE)*, pp. 19–24, 2013.

[17] U. Herberg and T. Clausen, "A comparative performance study of the routing protocols load and rpl with bi-directional traffic in low-power and lossy networks (lln)," *In the Proceedings of the 8th ACM Symposium on Performance evaluation of wireless ad hoc, sensor, and ubiquitous networks*, pp. 73–80, 2011.

[18] M. Vucinic, B. Tourancheau, and A. Duda, "Performance comparison of the rpl and loadng routing protocols in a home automation scenario," *In the Proceedings of IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 1974–1979, 2013.

[19] B. Karp and H.-T. Kung, "Gpsr: Greedy perimeter stateless routing for wireless networks," *In the Proceedings of the 6th ACM annual international conference on Mobile computing and networking*,pp. 243–254, 2000.

[20] A. Rao, S. Ratnasamy, C. Papadimitriou, S. Shenker, and I. Stoica, "Geographic routing without location information," *In the Proceedings of the 9th ACM annual international conference on Mobile computing and networking*, pp. 96–108, 2003.

[21] M. Dohler, D. Barthel, T. Watteyne, and T. Winter, "Routing requirements for urban low-power and lossy networks," 2009.

[22] F. Ren, T. He, S. Das, and C. Lin, "Traffic-aware dynamic routing to alleviate congestion in wireless sensor networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 22, no. 9, pp. 1585–1599, 2011.

[23] P. Misra, N. Ahmed, D. Ostry, and S. Jha, "Characterization of asymmetry in low-power wireless links: an empirical study," *Distributed Computing and Networking*, Springer, pp. 340–351, 2011

[24] D. S. De Couto, D. Aguayo, J. Bicket, and R. Morris, "A high-throughput path metric for multi-hop wireless routing," *Wireless Networks*, vol. 11, no. 4, pp. 419–434, 2005.

[25] Q. Wang, M. Hempstead, and W. Yang, "A realistic power consumption model for wireless sensor network devices," *In the Proceedings of 3rd Annual IEEE Communications Society on Sensor and Ad Hoc Communications and Networks (SECON'06)*, vol. 1, pp. 286–295, 2006.

[26] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," *In the Proceedings of the 33rd annual Hawaii international conference on system sciences*, pp. 10–20, 2000.

[27] O. Younis and S. Fahmy, "Heed: a hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks," *IEEE Transactions on Mobile Computing*, vol. 3, no. 4, pp. 366–379, 2004.

[28] C. Song, M. Liu, J. Cao, Y. Zheng, H. Gong, and G. Chen, "Maximizing network lifetime based on transmission range adjustment in wireless sensor networks," *Computer Communications*, vol. 32, no. 11, pp. 1316–1325, 2009.

[29] D. Niculescu and B. Nath, "Dv based positioning in ad hoc networks," *Telecommunication Systems*, vol. 22, no. 1-4, pp. 267–280, 2003.

[30] P. Thubert, "Objective function zero for the routing protocol for low-power and lossy networks (rpl)," 2012.

[31] M. Goyal, A. Brandt, and E. Baccelli, "A mechanism to measure the routing metrics along a point-to-point route in a low-power and lossy network," 2013.

[32] D. M. Blough and P. Santi, "Investigating upper bounds on network lifetime extension for cell-based energy conservation techniques in stationary ad hoc networks," *In the Proceedings of the 8th annual international conference on Mobile computing and networking*, ACM, pp. 183–192, 2002.

[33] G. F. Riley and T. R. Henderson, "The ns-3 network simulator," in *Modeling and Tools for Network Simulation*, Springer, pp. 15–34, 2010.

[34] N. Baccour, A. Koubaa, L. Mottola, M. A. Zuniga, H. Youssef, C. A. Boano, and M. Alves, "Radio link quality estimation in wireless sensor networks: a survey," *ACM Transactions on Sensor Networks (TOSN)*, vol. 8, no. 4, p. 34, 2012.

[35] R. Fonseca, O. Gnawali, K. Jamieson, and P. Levis, "Four-bit wireless link estimation." in *HotNets*, 2007.

[36] J. Zheng and M. J. Lee, "A comprehensive performance study of ieee 802.15. 4," 2004.