RPLIE: RPL for Indoor Environments under Midterm Link Fluctuations

Hyung-Sin Kim, Jeongyeup Paek, and Saewoong Bahk

Abstract: **Link dynamics due to environmental change is a critical challenge for wireless networking, and handling it is essential for wireless protocols, especially for low-power and lossy multihop wireless networks (LLNs). Prior work have only studied methods to detect and cope with short-term and long-term link dynamics in LLN. In this work, we tackle the *midterm dynamics* due to the opening and closing of doors and windows that can happen in a time scale of a few minutes to hours in an indoor LLN. Specifically, we study how midterm link fluctuations impact the performance of standard IPv6 routing protocol for LLN (RPL), and design '*RPLIE*,' a novel and backward-compatible lightweight enhancement to RPL that detects and overcomes the negative impact of such dynamics. We define '*opportunistic links*' and propose a novel routing metric, *expected breakage cost (EBC)*, which represents the expected number of link transmissions required from a link breakage to a routing parent change. We implement *RPLIE* on real embedded devices, and evaluate its performance against the standard RPL via experiments on a 31-node testbed to show that *RPLIE* achieves significantly better packet delivery performance while using less overhead.**

Index Terms: **IEEE 802.15.4, IPv6, interference, Internet of things (IoT) low-power lossy network (LLN), RPL, routing, wireless sensor network (WSN)**

I. INTRODUCTION

LOW-POWER and lossy wireless networks (LLNs) comprised of thousands of embedded wireless networking devices can be used in a variety of emerging Internet of Things (IoT) applications including smart grid automated metering infrastructures (AMIs) [1], [2], industrial monitoring [3], and wireless sensor networks [4]. However, link dynamics due to interference, fading, multipath, and environment change are critical challenges for wireless networking, and handling of it is essential for wireless protocols, especially for multihop low-power wireless networks such as the IEEE 802.15.4 [5]–[7].

Prior work [8]–[12] have studied the impact of, and methods to detect, mitigate, and cope with link dynamics in IEEE 802.15.4 networks using various techniques. Several work have also studied handling of external interference (e.g., WiFi, Bluetooth, Microwave oven) [13]–[19], and many routing and link-layer protocols are designed to handle internal interference (e.g., collisions) as well as link dynamics due to environmental changes [20]–[25] (Details of related work in Section VII). However, none of these work addresses the *midterm fluctuations* that we discuss in this work.

Link dynamics in a low-power wireless network can be categorized temporally: (1) Short-term fluctuations in the order of sub- to several seconds occur often due to internal (e.g., collisions) and external (e.g., WiFi) interference or human movements within the environment, and are often handled through link-layer ARQ mechanisms. (2) Long-term fluctuations are usually due to changes in physical environment (node relocation, building structure, furniture, etc.), and most routing protocols adapt to them through proactive beaconing [20], [21] or reactive search [26]. (3) Another type of link fluctuations that have received relatively less attention so far are the *midterm* fluctuations that are affected by the number of human occupants in the environment and the opening and closing of doors or windows that can happen in a time scale of a few minutes to hours.

These *midterm* fluctuations are not well handled by the short-term techniques such as link layer retransmissions, nor by routing protocols that update the topology slowly to avoid incurring huge control overhead. However, this *midterm* fluctuation problem has been neglected by the research community so far. Instead, researchers often conduct their experiments during night times when there are less variability in link dynamics from humans and their WiFi activities to generate and provide consistent, reproducible, and explainable data with as static physical environment as possible. We argue that, to gain better understanding of LLN performance, we must understand the dynamic characteristics of indoor LLN deployment when there are midterm fluctuations due to human activities, including physical environment change such as opening and closing of doors and windows. In addition, it is imperative that a network protocol built for indoor LLN must be capable of detecting and adapting to these changes.

Thus our goal in this work is to detect and distinguish *midterm* fluctuations from short-term ones in an indoor low-power wireless multihop networks, and adapt to them quickly by changing routes when necessary. To the best of our knowledge, this is the first work that tackles *midterm* fluctuations in the design of a wireless routing protocol.

In this work, we focus on how midterm physical environment

Manuscript received March 9, 2021; approved for publication March 18, 2021. This paper is specially handled by EIC and Division Editor with the help of three anonymous reviewers in a fast manner.

This work was supported by the Institute of Information & communications Technology Planning & Evaluation(IITP) grant funded by the Korea government(MSIT) (No.2018-0-00923, Scalable Spectrum Sharing for Beyond 5G Communication), by the National Research Foundation of Korea (NRF) grant funded by the Korea government Ministry of Science and ICT (MSIT) (No.2020R1F1A1051282), and also by the New Faculty Startup Fund from Seoul National University.

H. Kim is with the Graduate School of Data Science, Seoul National University, Seoul, Republic of Korea, email: hyungkim@snu.ac.kr

J. Paek is with the Department of Computer Science & Engineering, Chung-Ang University, Seoul, Republic of Korea, email: jpaek@cau.ac.kr.

S. Bahk is with the Department of Electrical Engineering, Seoul National University and INMC, Seoul, Republic of Korea, email: sbahk@snu.ac.kr.

J. Paek and S. Bahk are the corresponding authors.

Digital Object Identifier: 10.23919/JCN.2021.000012

changes such as opening and closing of doors impact the performance of a standard IPv6 routing protocol for LLN, and propose a novel backward-compatible design to overcome the negative impact of such dynamics. For this purpose, we first distinguish wireless links as 'good,' 'bad,' and '*opportunistic*' links. 'Good' links are those that can deliver packets reliably most of the time without experiencing large fluctuations. Short-term fluctuations may exist, but can be well coped with using link-layer retransmissions.[1] 'Bad' links are those that cannot deliver packets reliably most of the time, or are good for only small fraction of time due to significant short-term fluctuations in link quality. 'Opportunistic' links are those that are good for significant portion of time, but has non-negligible duration of times during which links are bad. Since these links are good for large fraction of times, they should be used for routing when beneficial. However, since they have non-negligible duration of times during which they are bad, usually due to midterm link fluctuations, they should be avoided during those times.

One may imagine that there is a fourth category of links that are stably in between good and bad without fluctuations (e.g., stably 50%). However, our link measurement study (Section III) shows that all links that are between good and bad are 'opportunistic' with certain level of fluctuations. *Thus our goal is to utilize these 'opportunistic' links as much as possible when beneficial, while avoiding them as much as possible when harmful.*

To achieve this goal, we propose a new routing metric, *expected breakage cost (EBC)*, which represents the expected number of link layer transmissions required from a link breakage to a routing parent change. Conceptually, EBC will be small if we can utilize the opportunistic link for a long time, and it will be large if the opportunistic link breaks often and results in parent changes. EBC has the same unit (in 'transmissions-per-packet') as ETX [27], and can be used additively with ETX in any routing protocol that uses ETX as the primary routing metric.

We will demonstrate how we use this idea in RPL, the IETF standard IPv6 routing protocol for LLNs [20], with objective function OF0 [28] and ETX link metric. Specifically, we propose an enhanced version of RPL, called *RPLIE*, that combines the concept of EBC and opportunistic links for routing metric and parent selection. We evaluate *RPLIE* through experiments on a 31-node multihop IEEE 802.15.4 testbed and show that *RPLIE* provides improved packet delivery reliability compared to the standard RPL while triggering less overhead.

The contributions of this work are threefold:

- We categorize link dynamics into short-term, *midterm*, and long-term, and attempt to capture the characteristics of the midterm dynamics in an indoor LLN using the concept of *'opportunistic'* links. We do so by introducing a new routing metric called *EBC* which presents the cost of link breakage when the link is used in a path by a routing protocol.
- We design '*RPLIE*,' a lightweight enhancement to RPL's routing metric and parent selection mechanism, which mitigates the effect of midterm fluctuation and provides improved routing paths.
- We implement *RPLIE* on embedded devices and experimentally evaluate its performance against the standard RPL on a real 31-node testbed to show that *RPLIE* achieves significantly better packet delivery performance than RPL while using less overhead.

The remainder of this paper is structured as follows. Section II presents our experiences which motivated our work, and Section III investigates link characteristics through a measurement study. Section IV describes the problem of RPL in handling midterm link dynamics, and Section V proposes the design of *RPLIE* to address the problem. Section VI evaluates *RPLIE* using testbed experiments, Section VII discusses the related work, and Section VIII concludes the paper.

II. MOTIVATION

The motivation behind this work is from our own experiences while working with a multihop routing protocol on indoor LLN testbeds. For example, one of our testbeds is composed of 31 telosb motes [29] (30 sensor nodes and 1 root node), and is deployed in an office environment on the 3rd floor of INMC building at Seoul National University. While running experiments, we have often noticed significant changes and differences in the routing topology constructed by the routing protocol, even though we are running the experiments on the same testbed with same transmission power (e.g., -15 dBm).

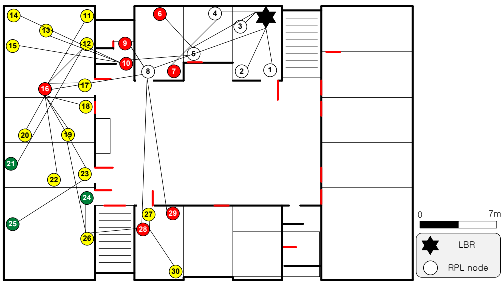
The differences were significant and noticeable, as seen in Figs. 1(a) and 1(b) where the former is a 4-hop topology and the latter is a 7-hop topology. After investigation, we found that the door of the room where nodes 27, 28, 29, and 30 are placed causes the topology difference between Figs. 1(a) and 1(b). The former is the routing topology constructed when the door was open, and the latter is when the door was closed. We found that the links between node 8 and nodes 28, 29 across the center of the building existed at one point in time, but vanished at other times depending on the opening and closing of the doors.

We looked into how frequently the door is opened/closed and why it happens. It turns out that the room is a staff lounge which is occupied several times in a day, and its occupants open the door to control the room temperature because the air conditioner was not controllable from within the room (centralized control). Therefore, once the door is opened, it remains open for minutes to hours until it is closed, and this happens several times a day resulting in repeated topology changes as shown in Fig. 1. We were able to manually *reproduce* this phenomenon by opening and closing the door during the night-times, verifying that the door status does result in the observed topology changes.

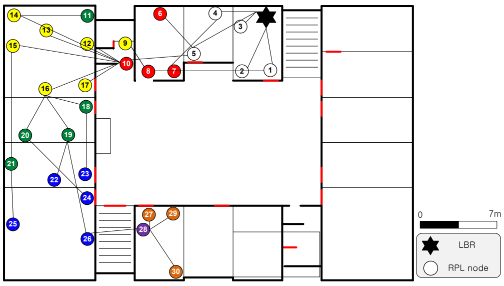
Although this may not sound surprising after knowing the reason, it was something that made us spend countless hours analyzing the cause of inconsistent experimental results. Then it came to our mind; *"can we characterize those links that appear and disappear due to opening and closing of doors in an indoor environment?"* and *"can the routing protocol detect, utilize, and avoid these links appropriately?"* These are the two questions we ask in this work.

---

[1] Recall that, long-term fluctuations will eventually be adapted to by the link probing mechanisms of link/routing layer protocols, if any.



(a) With doors open: 4-hop network.



(b) With doors close: 7-hop network.

Fig. 1. Routing topology on a 31-node indoor testbed.

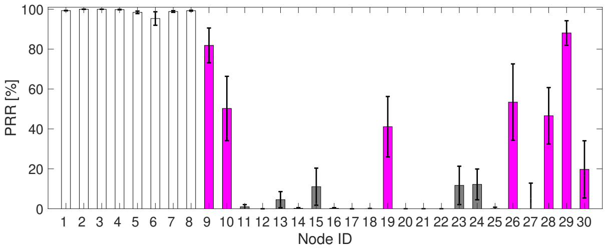


Fig. 2. Average link packet-reception-ratio (PRR) result for 24 hours. white bars are good links, grey bars are bad links, and pink are opportunistic links.

III. PRELIMINARY STUDY

Clearly, opening and closing of doors are at a much longer time scale than what link-layer protocols are designed to cope with, but at the same time can be at a shorter time scale than how often major changes in the physical environment can occur. Based on this intuition, we define three types of link fluctuation as below:

- **Short-term**: Link fluctuations with sub-second or sub-minute time scale. Several prior studies suggest that short-term fluctuations are often due to Wi-Fi interference (characterized by low PRR despite high RSSI) and human movements (characterized by low PRR and low RSSI) [30], [31].
- **Midterm**: Link fluctuations that occur at a time scale of hours (a few times per day). Midterm fluctuations are mainly due to obstacle movements such as door/window opening and closing, and this can be characterized by sudden change of PRR / RSSI / ETX that is not long-term.
- **Long-term**: Link fluctuations that rarely occur (e.g., once per days or months), possibly due to changes in the environment. Wireless links with long-term fluctuations are usually considered to be static.

A. *LINK MEASUREMENT*

To investigate the link characteristics, we conducted link measurements using one transmitter (marked as the star in Fig. 1) and 30 receivers to capture the fluctuation of packet reception ratio (PRR) during a 24-hour period. The transmitter broadcasts a packet with transmission power of 0 dBm every 50 ms, and the receivers measure the *per-minute* PRR for 24 hours. The goal of this experiment was to show the existence of midterm fluctuation; long-term fluctuation is not expected to be captured due to the experiment's time period.

Fig. 2 depicts the PRR for each node averaged over the 24-hour period. For each node, we also plot standard deviation of per-minute PRR for 24 hours as a black error bar. It shows that there are 8 good nodes (white bars) with PRR over 95% and 14 bad nodes (grey bars) with PRR below 15% while the remaining nodes (pink bars) were in between. We call the links between these pink nodes and the transmitter 'opportunistic links.' Then, two questions can arise naturally:

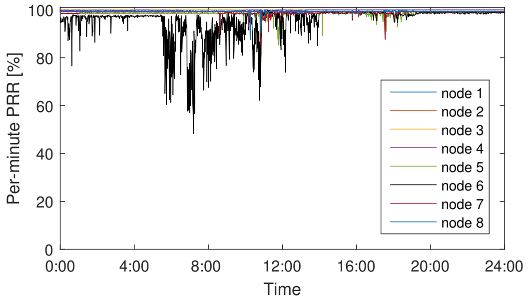
- **Existence of fluctuations**: If an opportunistic link has overall average PRR of 50%, can we say that this link has 50% chance of packet delivery most of the time? or is it time-dependent? (fluctuate over time?)
- **Existence of 'midterm' fluctuations**: If an opportunistic link's PRR fluctuates, how frequently does it fluctuate and at what time scale?

Non-negligible error bars do exist in Fig. 2, which indicates that the answer to the first question is straightforward: opportunistic links' PRR fluctuates over time. However, to answer the second question, we need to look at the PRR in the time-domain.

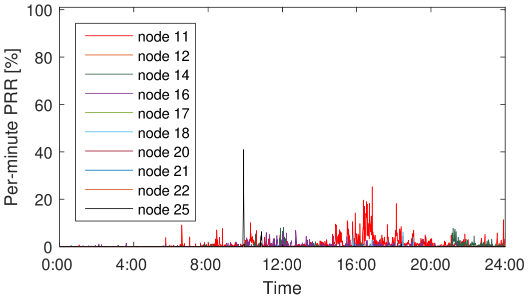
We first look into good and bad links for comparison. Fig. 3 plot the per-minute PRR over 24-hour period for a subset of selected good and bad links. First of all, the eight good links in Fig. 3(a) (represented as the white bars in Fig. 2) were good almost always (although one node had some variation in the morning time). The ten bad links in Fig. 3(b), represented as the ten grey bars in Fig. 2 that are too short to be seen, were always bad. The results show that almost all of good and bad links do not fluctuate over time, at least within a day. Although one good link represented as the black line in Fig. 3(a) has marginally fluctuating per-minute PRR in the morning time, the per-minute PRR is still high enough. For these links, the long-term average PRR can be used as their link qualities.

Furthermore, as shown in Fig. 3(c), there were five bad links which were bad most of the time but had occasional spikes of good PRR that is significantly different from the 24-hour average PRR performance (<15%). Since a high *per-minute* PRR means that a link was good *for a minute*, these five links may be classified as good links by a routing protocol temporarily depending on how the protocol is designed. However, it is not a wise choice for a routing protocol to utilize them due to lack of stability. Thus, although the (long-term) average PRR cannot represent their link characteristics of every single time period, the average PRR is still good enough to be used for link classification.

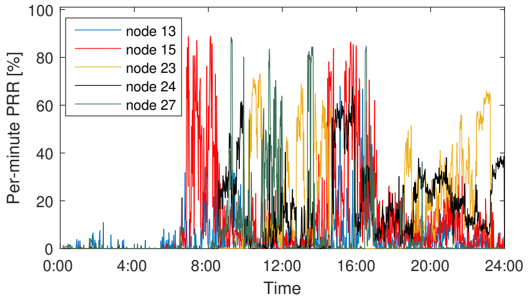
On the other hand, Fig. 4 shows the per-minute PRR of seven opportunistic links that had more interesting characteristics: The *"midterm fluctuations"*. We have written the 24-hour average PRR of each link in the legends for clear explanation. First of



(a) Good links



(b) Bad links, always bad



(c) Bad links, good only for a short period

Fig. 3. Link measurement results for 24 hours: Per-minute PRR of 8 good and 15 bad links.

all, links 9 and 29 in Fig. 4(a) have fairly high average PRR, higher than 80%. Both of them, however, have significantly different characteristics in the time domain. Link 9 is stably good for the first 9 hours but starts to fluctuate for the rest of the day, where its PRR sometimes becomes even below 20%. Link 29 is good almost all the time but suddenly becomes very bad for about 90 minutes around noon, worse than link 9 and close to 0%. The results show that although the two links have fairly high average PRR, they should not be used for non-negligible amount of time (i.e., hour scale); their link qualities cannot be represented by long-term average PRR only.

The four links in Figs. 4(b)–4(d) have average PRR of around 50%, but again fluctuate differently in the time domain. Links 10 and 28 in Fig. 4(b) are stably good for the first 7 hours and significantly fluctuate for the rest of the day. Link 26 in Fig. 4(c) is stably good for the first 16 hours and become totally disabled for the rest of the day. Link 19 in Fig. 4(d) is bad for the first 8.5 hours, fluctuates for the next 9.5 hours, and become stable during the night-time. Although their average PRRs are not good, it turns out that the four links do become available for 5∼16 hours, good enough to be used as valid links. In addition, their fluctuation patterns are significantly different from each other in terms of when and how long they are available.

Lastly, link 30 in Fig. 4(e) has the lowest average PRR among the opportunistic links, below 20%. The link does not exist (∼0% PRR) most of the time, but becomes available for a considerable amount of time in the afternoon where its PRR is sometimes even higher than 80%.

The results in Fig. 4 confirm that all the 7 opportunistic links have midterm fluctuations, making them good or bad for several hours continuously. These midterm fluctuations are clearly distinguished from short-term fluctuations that some good (link 6) and bad links (links 13, 15, 23, 24, and 27) have. Even with short-term fluctuations, the good links are still good and the bad links are still bad since they have marginal and temporary impact. In contrast, with midterm fluctuations, a link that used to be good can suddenly become bad for hours.

B. *Perspective of Routing Protocol Design*

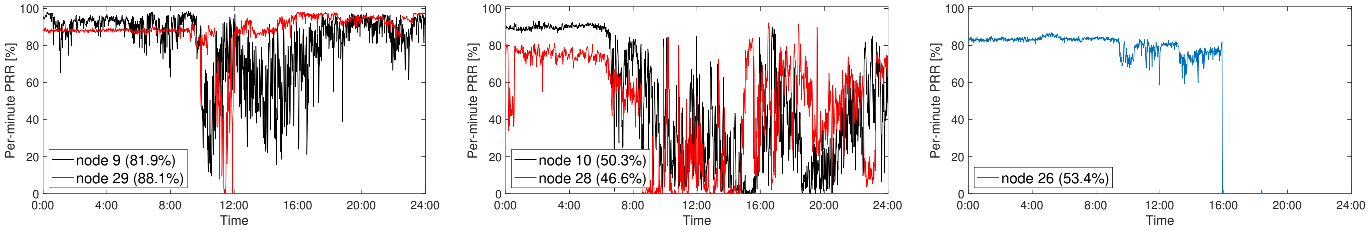
Based on the link measurement study in Section III.A, our goal in this paper is to improve the packet delivery performance of the network by distinguishing links based on their fluctuation characteristics.

Good links are stable with a small amount of short-term fluctuation. For these links, current ETX-based link quality metric is suitable, and link layer retransmission is enough to overcome short-term fluctuation. Bad links should not be used for routing, even if a node luckily received a reachability beacon from a neighbor due to short-term fluctuations (e.g., the case of the five links in Fig. 3(c)). Although the ETX-based metric eventually characterizes these links since ETX increases after continuous transmission failures, it can temporarily be good and cause confusion to a routing protocol. Therefore additional metrics, such as RSSI, needs to be used for a routing protocol to effectively characterize these links as ‘bad’ even when receiving their beacons.

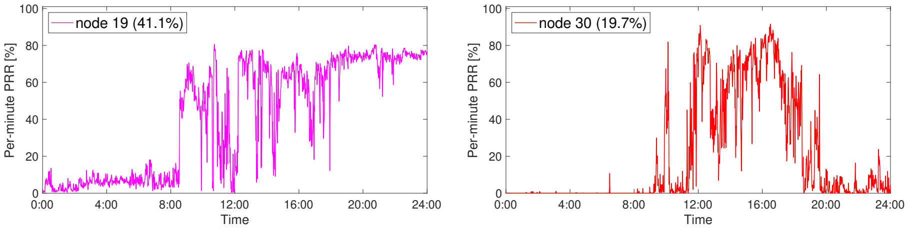
On the other hand, opportunistic links are not stable, but good (or bad) opportunistically, and long-term PRR cannot represent the opportunistic nature of the link. For example, a link with long-term PRR of 50% can provide >90% PRR for several hours and <20% PRR for other several hours. In this case, ETX is not a suitable (or sufficient) link metric since it cannot represent how frequently a link is broken. Furthermore, for opportunistic links, link layer retransmission is not enough for reliable packet delivery because these links can be broken for a considerable amount of time. Thus, we need a new metric to detect and evaluate these opportunistic links and utilize them appropriately, which leads to our design of the new *expected breakage cost (EBC)* metric that we propose in Section V.

IV. PROBLEM

In Section II, we have shown that two clearly distinct routing topologies, a 4-hop and a 7-hop topology, can be formed depending on the opening and closing of doors (Figs. 1(a) and 1(b)). These were the outcome of RPL, the IETF Internet standard IPv6 routing protocol for low-power and lossy networks [20], with the default OF0 [28] objective function. In



(a) Example 1 (b) Example 2 (c) Example 3



(d) Example 4 (e) Example 5

Fig. 4. Link measurement results for 24 hours: Per-minute PRR of 7 opportunistic links with mideterm fluctuations.

this section, we elaborate what problems may occur when using LLN routing protocols such as RPL faced with opportunistic links due to midterm dynamics.

A. RPL: IPv6 Routing Protocol for LLNs

RPL is designed for resource constrained embedded devices to support upcoming smart grids and many other LLN applications [20], [32]. It is a distance vector type routing protocol that builds directed acyclic graphs (DAGs) based on routing metrics and constraints. In most deployment scenarios, RPL constructs quasi-forest routing topology called destination-oriented directed acyclic graph (DODAG) rooted at an LLN border router (LBR), and supports bi-directional IPv6 communication between network devices. Each node in RPL advertises routing metrics and constraints through DODAG information object (DIO) messages, and builds a DAG according to its objective function (OF, rules governing how to build a DAG) and the information in DIO messages. Upon receiving DIO messages from its neighbors, a node chooses a routing parent (default path) according to its OF and local policy, and then constructs a routing topology (i.e., DODAG).

Although there have been several prior works that propose enhancements to RPL, the basic default version of RPL with OF0 uses hop-count and ETX for selecting parent nodes and constructing routing paths towards the root node. Specifically, OF0 calculates node k's rank from hop count, as

$$RANK(k) = RANK(p) + 1, \tag{1}$$

where node p is node k's parent node. The routing cost of node k toward node p is defined as

$$Cost(k,p) = RANK(p) + ETX(k,p). \tag{2}$$

Thus in RPL, a node includes a neighbor in its parent candidate set if it receives a DIO message from the neighbor (i.e., packet reception-based link validity check), and all types of neighbors (with good, opportunistic, and bad links) are evaluated using the same routing metric (RANK + ETX) without any classification according to their link fluctuation characteristics. When an opportunistic or bad neighbor has much lower rank (i.e. hop count) than good neighbors, RPL will select such a node as the parent and suffer from unreliability, unstable topology, and high routing overhead.

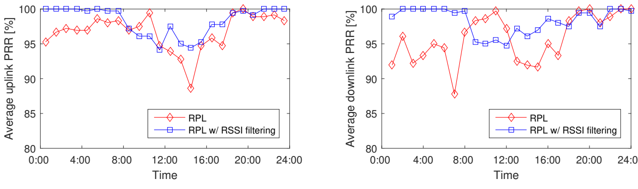
Since low RSSI is one of the main reasons for bad links [33], 'RSSI filtering' is a simple and well-known technique in LLN to remove most of the bad links: insert a node into the parent candidate set only when receiving a DIO message from that node with an RSSI higher than a threshold. However, this may not be sufficient to properly deal with opportunistic neighbors, which we will investigate further in the next subsection.

B. RPL in the Presence of Opportunistic Links

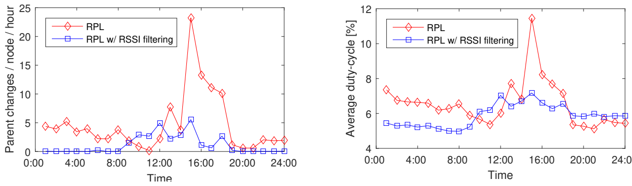
We experimentally investigate how RPL handles various types of links to build its topology. To this end, Fig. 5 plots various performance metrics of RPL[2] with and without RSSI filtering when using the same configuration as Fig. 1 (i.e., 1 root and 30 nodes, all with transmission power of -15 dBm). Each node generates an upward packet every 1 minute and the LBR generates a downward packet for each node every 1 minute (i.e., 30 downward packets per minute). We set the RSSI filtering threshold value to -90 dBm.

Fig. 5 shows that without RSSI filtering, RPL suffers from low PRR, topology churns, and high duty-cycle throughout the day. This indicates that packet reception-based link validity check can be vulnerable even when wireless links are stable (i.e., night-time). This is mainly because RPL misunderstands a bad link as good when it receives control packets from the bad link fortunately due to short-term fluctuations. RPL with hop count and ETX only is vulnerable to fluctuating bad links, such as the ones in Fig. 3(c).

[2] For our experiments, we used TinyRPL (RPL implementation on TinyOS) [34] and BoX-MAC-2 (the default MAC of TinyOS) [35] for the underlying link layer protocol.



(a) Uplink PRR (b) Downlink PRR



(c) Parent change frequency (d) Duty-cycle

Fig. 5. Performance of default RPL, with and without RSSI filtering.

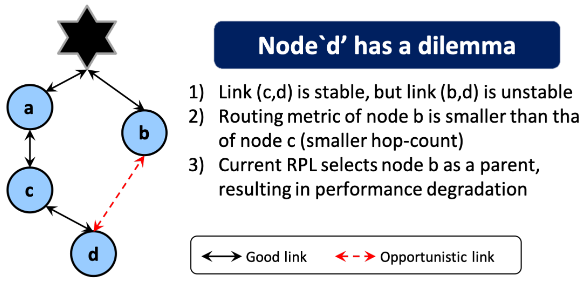


Fig. 6. Problem scenario that *RPLIE* tries to tackle.

With RSSI filtering, PRR becomes almost perfect and routing topology becomes stable during the night-time, which show that signal strength information does help in removing bad links with short-term fluctuations from routes [36]. However during the day-time, it is still insufficient; the network still suffers bad performance due to lack of a mechanism to deal with opportunistic links having midterm fluctuation.

Overall, evaluating link and path quality using RSSI, ETX and hop-count is a simple but insufficient approach in indoor environments. The problem of ETX and RSSI is that they represent only current link quality; these metrics do not provide any information on how frequently a link becomes bad (or good).

Fig. 6 illustrates the problem scenario in the standard RPL when an opportunistic link exists. If link (b, d) fluctuates but node b has lower RANK than node c, node d suffers routing instability even though it has a stable neighbor c. Specifically, when node d receives a DIO from node b while having node c as the parent, it changes the parent to node b due to its lower RANK. When node b is disconnected, node d experiences significant amount of packet losses and finally changes its parent to node c. When link (b, d) becomes available again, however, node d selects node b as the parent again. This is the problem that *RPLIE* tries to tackle.

V. *RPLIE* WITH EBC-BASED ROUTING METRIC

Identifying an *opportunistic link* and estimating how long it will be available is a challenging task. A good news in the case of indoor LLN, however, is that nodes are usually deployed at fixed locations and human activities during a day are usually similar in an indoor environment. To this end, we design *RPLIE*, a simple but effective enhancement to RPL that improves both the reliability and efficiency by learning the long-term and indoor-specific link characteristics and use this information for classifying link types. *RPLIE* is differentiated from RPL as follows:

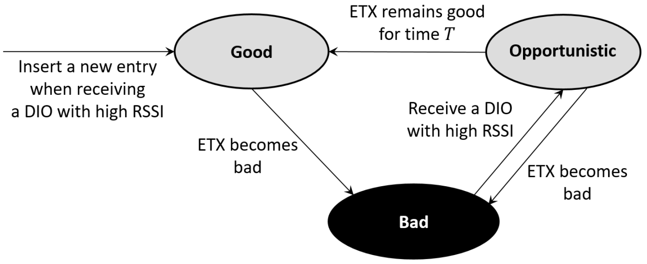


Fig. 7. Link state management finite state machine (FSM) of *RPLIE*.

- *RPLIE* classifies neighbor's link quality into three types; good, opportunistic and bad.
- *RPLIE* uses not only ETX and RSSI, but also the stability of those (represented in a new metric EBC (expected breakage cost)) to evaluate link quality of neighbor nodes.
- *RPLIE* has two types of parent nodes; good and opportunistic parents; that can be used for routing.
- *RPLIE* provides opportunistic upward routing and conservative (good) downward routing.

A. *Link State / Parent Table Management*

Fig. 7 illustrates *RPLIE*'s link state management finite state machine (FSM) for parent candidates in its parent table. *RPLIE* manages three link states as follows:

- **New entry (Good):** When a *RPLIE* node receives a DIO from a new neighbor after RSSI filtering, it adds the neighbor to the parent table and initializes its link state as 'good' and ETX as 1 (the best value) since it has no prior knowledge of the neighbor's link quality.
- **Good/opportunistic to bad:** When a 'good' or 'opportunistic' neighbor's ETX becomes bad (i.e., higher than a threshold), *RPLIE* changes the neighbor's link state to 'bad.'
- **Bad to opportunistic:** When a *RPLIE* node receives a DIO from a 'bad' neighbor with high RSSI, it changes the neighbor's link state to 'opportunistic.'
- **Opportunistic to good:** Given that an 'opportunistic' neighbor's ETX can fluctuate, *RPLIE* changes the neighbor's link state to 'good' only when its ETX remains good for a long time T (1 day in our experiments).

With this link state management policy, 'good' neighbors' link quality should be *stably* good. 'Opportunistic' neighbors' link quality might be good now but it was bad at least once before, which should be considered carefully for packet forwarding. 'Bad' neighbors' link quality is bad now, which should not be used for packet forwarding. Therefore, *RPLIE* considers only 'good' and 'opportunistic' neighbors as valid parent candidates.

B. Expected Breakage Cost (EBC)

The next question is how to evaluate midterm fluctuation of an 'opportunistic' parent. Even if two 'opportunistic' parents currently have same hop-count and ETX, one should have higher link cost than the other if it is disconnected (i.e., bad) more frequently. To incorporate this into the design, *RPLIE* adds a new metric called *expected breakage cost (EBC)* to RPL's routing cost as:

$$Cost(k,p) = RANK(p) + ETX(k,p) + EBC(k,p). \tag{3}$$

Here $Cost(k,p)$, $ETX(k,p)$, and $EBC(k,p)$ are the routing cost, ETX, and EBC from node k to its parent candidate p, respectively.

Given that $Cost(k,p)$'s unit is in number of transmissions, $EBC(k,p)$ should have the same unit for its smooth integration. Specifically, we define $EBC(k,p)$ as

$$EBC(k,p) = \frac{BC}{MT(k,p) \times TL(k)}, \tag{4}$$

where BC, $MT(k,p)$ and $TL(k)$ are defined as below:

- BC **(Breakage cost)** is the number of link layer transmissions from a link breakage to a parent change. When a node is disconnected from its parent, it transmits upward packets to the disconnected parent until the parent link state becomes 'bad' (EWMA filter increases ETX above the ETX threshold). This wasted number of transmissions is BC, which is a constant value determined by maximum number of retransmissions and the ETX threshold. $EBC(k,p)$ is proportional to BC.
- $MT(k,p)$ **(Maintenance time)** is the time duration (minutes) from when node k selects node p as its preferred parent to when node k changes its preferred parent to another node because link (k,p) is broken; although node k selects another parent better than node p, $MT(k,p)$ is not updated if the link (k,p) is still valid. $MT(k,p)$ is initialized to maximum value (1 day in our experiments) and updated using an EWMA filter. A larger $MT(k,p)$ value means that link (k,p) is more robust, resulting in a smaller $EBC(k,p)$.
- $TL(k)$ **(Traffic load)** is the number of packet transmissions at node k's network (i.e. IP) layer per minute, including its packet relaying. We use an EWMA filter to update $TL(k)$. A larger $TL(k)$ value means that node k can send more packets to node p while its link is valid, which leads to lower $EBC(k,p)$.

Overall, $EBC(k,p)$ represents the ratio of wasted transmissions because of a link breakage to valid transmissions before the link breakage.

C. Differentiated Routing for Uplink and Downlink

Unlike RPL which sets the downward routes as the reverse of upward routes, *RPLIE* differentiates upward and downward routing. This is done without changing the basic operations of RPL (e.g., DIO/DAO exchange) for backward compatibility. The key reason behind this design choice is because in RPL, a bad link in a downward route is detoured (if possible) much slower than that of an upward route [37]. Therefore we aim to construct downward routes in a conservative manner while taking chances for shorter upward routes in an opportunistic manner. That is, *RPLIE* uses not only 'good' but also 'opportunistic' links (under certain conditions explained below) for upward routing, and a downward route consists only of 'good' links.

Specifically, a *RPLIE* node selects two parents, a 'good' parent from 'good' parent candidates and an 'opportunistic' parent from 'opportunistic' candidates. Then it sends DAO only to the 'good' parent for (conservative) downward routing, but sends upward data packets to either of the two parents which has a smaller routing cost including EBC. Given that a good parent's EBC is almost zero, an opportunistic parent having a smaller routing cost than the good parent even with its EBC penalty is assumed to be valuable enough to replace the good parent.

After establishing its bidirectional route, a *RPLIE* node calculates its RANK based on the 'good' parent's RANK even if/when the node uses the 'opportunistic' parent for upward route:

$$RANK(k) = RANK(p_{good}) + 1 \quad \text{in case of OF0}. \tag{5}$$

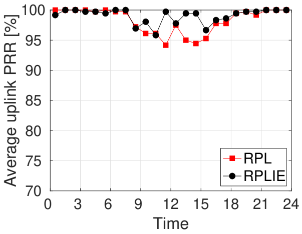
This is because the RANK may change frequently if it is based on the 'opportunistic' parent, which destabilizes the *RPLIE* node's subtree.

VI. EVALUATION

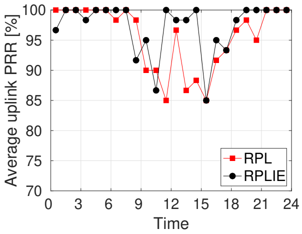
We evaluate the performance of *RPLIE* and compare it against that of RPL. We perform our experiments in the same environment as before, the same indoor testbed as shown in Fig. 1 comprising 31 Telosb motes with transmission power of -15 dBm. Each node generates an upward packet every 1 minute, and the LBR generates a downward packet for each node every 1 minute (i.e., 30 downward packets per minute), totalling 60 packets per minutes (1 pkt/sec). Since our link measurement study in Section III shows that good and bad links are stable during a day, we set T to 1440 minutes for *RPLIE*. Then we plot results for various performance metrics over time for 24 hours to observe the impact of human activities. As shown in Fig. 3, there are mainly good and bad links during the night-time, but all the three link types including opportunistic link appear during the day-time.

Fig. 8 plots various performance metrics of RPL and *RPLIE* from our experiment. Figs. 8(a)–8(d) plot the average bidirectional PRR of RPL and *RPLIE* for all the 30 nodes, as well as the average for only nodes 26 to 30. Given that the selected parents of nodes 26-30 are significantly affected by opportunistic links as shown in Fig. 1, we plot these nodes' performance separately to clearly show the effectiveness of *RPLIE*. Figs. 8(a) and 8(c) show that both RPL and *RPLIE* provide reliable packet delivery during the night-time; handling only good and bad links are relatively easy for both protocols. During the day-time, however, PRR performance of both protocols fluctuate, where *RPLIE* outperforms RPL almost all the time.

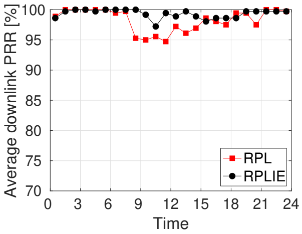
Figs. 8(b) and 8(d) show that nodes 26-30 are affected by channel fluctuation during the day-time much more severely than the other twenty five nodes. Downlink PRR is degraded more than uplink PRR, verifying that RPL's downward route is more fragile to link fluctuation than its upward route. *RPLIE* almost always outperforms RPL, which verifies the effectiveness



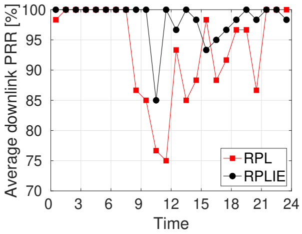
(a) Uplink PRR for all 30 nodes



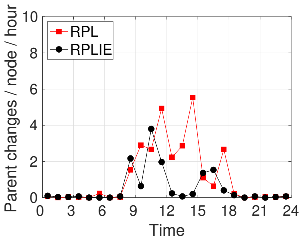
(b) Uplink PRR for nodes 26-30



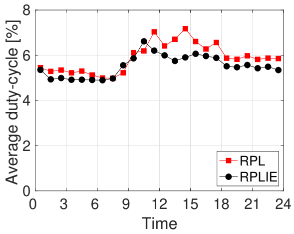
(c) Downlink PRR for all 30 nodes



(d) Downlink PRR for nodes 26-30

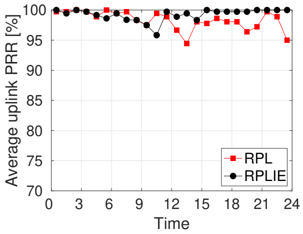


(e) Parent change for all 30 nodes

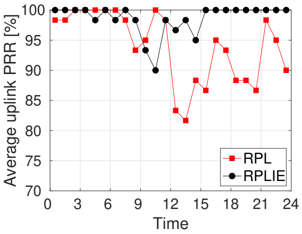


(f) Duty cycle for all 30 nodes

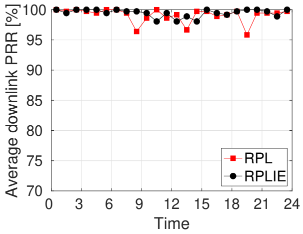
Fig. 8. Performance of RPL and *RPLIE* for a day (first run).



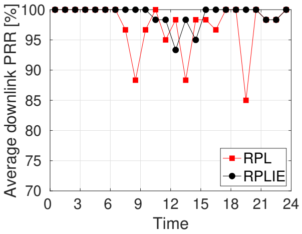
(a) Uplink PRR for all 30 nodes



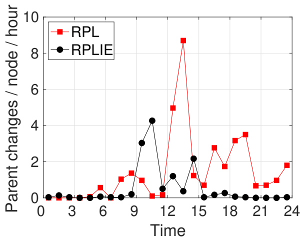
(b) Uplink PRR for nodes 26-30



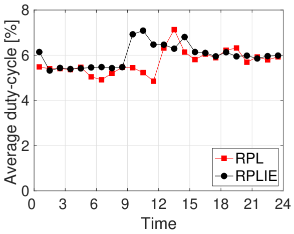
(c) Downlink PRR for all 30 nodes



(d) Downlink PRR for nodes 26-30



(e) Parent change for all 30 nodes



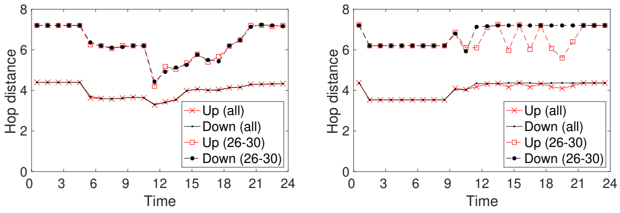
(f) Duty cycle for all 30 nodes

Fig. 9. Performance of RPL and *RPLIE* for a day (second run).

of distinguishing opportunistic link from good and bad links via EBC and handling them differently and appropriately.

To make sure that these results are not an outcome of just one lucky run, we ran the experiments several times and observed consistent results. For example, Fig. 9 is the performance results from another run of the experiments, similar to Fig. 8. An interesting observation is that in this second set of results, RPL does not provide reliable packet delivery after noon, even in the evening. This phenomenon can happen since some opportunistic links fluctuate in the evening, as already shown in Fig. 4(b). This shows that *RPLIE*'s performance gain is not strictly for the day-time but also for the night-time.

To investigate more deeply, Figs. 10(a) and 10(b) plot hop distance of RPL and *RPLIE*, respectively. The two figures show that RPL decreases hop distance of nodes 26-30 at noon and maintain the low hop distance even in the evening. This is because RPL tries to use opportunistic links whenever it becomes available for a short time. This aggressive utilization of the unreliable opportunistic links degrade performance. As shown in Fig. 9(e), RPL frequently triggers parent changes in the evening, showing its effort to find stable parents. Without a proper method that characterizes opportunistic links, however, the effort causes nothing but topology churn. In contrast, *RPLIE* recognizes and de-prioritizes these unreliable opportunistic links via high EBC values. Lowering priority of the unreliable opportunistic links increases hop distance of nodes 26-30 by two, which is a reasonable compromise for reliable packet delivery.



(a) RPL

(b) *RPLIE*

Fig. 10. Average hop distance of RPL and *RPLIE* for a day (second run).

Figs. 10(a) and 10(b) show another important characteristic of *RPLIE*, opportunistic upward routing and conservative downward routing. In contrast to RPL, *RPLIE* provides different hop distance for upward and downward routes. The figures show that the former becomes smaller than the latter occasionally, confirming *RPLIE*'s behavior that tries to utilize opportunistic links for upward routing as much as possible. *RPLIE*'s opportunistic upward routing is not too aggressive like RPL, but utilizes opportunistic links only if they are stable enough, which does not cause performance degradation.

Fig. 8(e) plots parent change frequency of all 30 nodes. It shows that neither RPL nor *RPLIE* incur any parent change during the night-time since every parent-child link is stably good. During the day-time, however, both trigger parent changes where RPL changes parents more frequently than *RPLIE*. This is because a RPL node can select a neighbor link that temporarily has good quality as its parent, which causes it to change its

Table 1. Performance summary.

Metric	First run		Second run	
	RPL	**RPLIE**	**RPL**	**RPLIE**
Uplink PRR [%]	98.29	99.07	98.34	99.24
Downlink PRR [%]	98.30	99.33	99.12	99.47
Parent change frequency	1.16	0.54	1.47	0.53
Duty cycle [%]	5.88	5.52	5.69	5.99

parent later again. In contrast, *RPLIE* considers not only link quality but also link stability, providing more reliable parent selection during the day-time. Fig. 8(f) plots radio duty-cycle of each node, which shows that *RPLIE* consumes lower energy than RPL, specifically during the day-time. This is thanks to its reliable routing strategy which reduces routing control overhead.

Figs. 9(e) and 9(f) tell a similar story. In this case, opportunistic links start to fluctuate at different times, 9 AM for *RPLIE* but 12 PM for RPL. Thus, *RPLIE* starts to change parents (Fig. 9(e)), increase hop distance (all-node cases in Figs. 10(a) and 10(b)), and also increase duty cycle (Fig. 9(f)) from 9 AM. This explains why *RPLIE* has higher duty cycle than RPL in the morning time for this experiment, not because it is inefficient but because it starts to handle opportunistic links earlier than RPL.

Table 1 summarizes average performance of the two sets of experiments. Compared to RPL, *RPLIE* provides better packet delivery, routing stability, and energy efficiency. Slightly higher duty cycle in the second run is because opportunistic links start to fluctuate earlier in the *RPLIE* experiment than in the RPL experiment, which is reasonable operation as explained previously. Overall, the results prove that it is worthwhile to classify opportunistic links, measure their stability using the EBC metric, and utilize them differently from good or bad links.

Discussion: *RPLIE* transitions an opportunistic link state to 'good' only when its ETX remains good for a configurable time T. The value of parameter T determines the stability and agility of *RPLIE*, and T should be configured sufficiently long enough to include most midterm fluctuation periods, but at the same time short enough to effectively take advantage of the opportunistic links. For example, if T is 10 hours, *RPLIE* will change every opportunistic link to good links during the night-time, which will cause more link breakages next day-time. If T is too long (e.g., days or weeks), it will exclude too many options for downward paths since an unfortunate link breakage will exclude the links for days or weeks. Since what we define as 'midterm dynamics' is mostly due to daily human behavior, 1 day used in our experiments seems to be a reasonable choice. However, it can be configured based on the target scenario and application requirements.

VII. RELATED WORK

There have been a few pieces of work that evaluated RPL under link dynamics. Han *et al.* experimentally evaluated RPL's performance under WiFi interference [13], and showed that RPL experiences not only severe packet losses but also a large number of redundant parent changes in the presence of wireless interference. Mohammad *et al.* conducted a measurement study at various areas such as shopping malls, parking lots, residential complex, and cafeteria to confirm the existence of wireless interference at those area [38]. To mitigate interference, the authors designed Oppcast that combines opportunistic routing with a simple frequency hopping mechanism and a receiver-initiated MAC protocol, and showed that Oppcast outperforms both RPL and ORPL in various real-world fields. Lee *et al.* proposed to use BLE (instead of IEEE 802.15.4, suggested by 6LoWPAN [39]) under RPL to avoid interference because BLE has an adaptive frequency hopping mechanism [40]. Through testbed experiments, they showed that RPL over BLE provides more reliable packet delivery than RPL over IEEE 802.15.4 under wireless interference.

In general, mitigation of multipath fading and wireless interference at the 2.4 GHz ISM band is a popular research area in LLNs since many practical application environments have significant external interference due to overlapping frequency band. However, most previous work focused on improving PHY and MAC layers for interference classification [11], [16], frequency hopping [41], [42], multi-antenna diversity [12], [43], adaptive duty-cycling [17], [44], and error recovery [14], [15], [45], [46] without considering routing layer's behavior nor suggesting any improvement to the routing protocol. These link quality enhancement techniques are either insufficient or orthogonal to our work since, for example, they cannot undo a metal door closing. We argue that the best way is not to try to recover an opportunistic link when it is broken (which may not be possible), but characterize the link properly for the routing protocol to make the right decision.

There are also numerous prior work on improving the performance of RPL under various scenarios (e.g., load balancing [24], TX power control [25], P2P [47], dissemination and collection [48]), but none of them have handled the *midterm* link fluctuations. To the best of our knowledge, there has been no work that discusses the impact of *midterm* link quality fluctuations in indoor environments, nor has anyone proposed solutions to the problem.

VIII. CONCLUSION

We have tackled an indoor-specific *midterm* wireless link fluctuation problem in LLNs. We were motivated by the fact that human activities such as opening and closing of doors generate non-negligible amount of *opportunistic links* with dynamics in the order of few minutes to hours, and without an appropriate mechanism to handle these links, RPL misidentifies each as either good or bad, resulting in routing topology churn. To resolve the problem, we proposed a new link cost metric EBC which measures the stability of an opportunistic link, and designed *RPLIE* that incorporates EBC into the routing cost metric and selects parent nodes for opportunistic upward and conservative downward routing. Through indoor testbed experiments on 31 nodes, we evaluated *RPLIE* and compared its performance against the standard RPL to show that *RPLIE* improves performance under human activities by carefully handling opportunistic link.

REFERENCES

[1] Cisco, "Connected Grid Networks for Smart Grid - Field Area Network / CG-Mesh," Available online: http://www.cisco.com/web/strategy/energy/field_area_network.html, accessed on 2021-03-09.

[2] E. Ancillotti, R. Bruno, and M. Conti, "The role of the RPL routing protocol for smart grid communications," *IEEE Communications Magazine*, vol. 51, no. 1, pp. 75–83, Jan. 2013.

[3] V. Gungor and G. Hancke, "Industrial wireless sensor networks: Challenges, design principles, and technical approaches," *IEEE Trans. Ind. Electron.*, vol. 56, no. 10, Oct. 2009.

[4] J. Paek *et al.*, "The tenet architecture for tiered sensor Networks," *ACM Trans. Sensor Netw.*, vol. 6, no. 4, July 2010.

[5] "IEEE Standard for Information technology – Telecommunications and information exchange between systems – Local and metropolitan area networks. Specific requirements – Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs)," Available at http://www.ieee802.org/15/pub/TG4.html, May 2003, accessed on 2021-03-09.

[6] J. Zhao and R. Govindan, "Understanding packet delivery performance in dense wireless sensor networks," in *Proc. ACM SenSys*, 2003.

[7] H. Kermajani and C. Gomez, "On the network convergence process in RPL over IEEE 802.15.4 multihop networks: Improvement and trade-offs," *Sensors*, vol. 14, no. 7, pp. 11 993–12 022, 2014.

[8] K. Srinivasan, M. Kazandijeva, S. Agarwal, and P. Levis, "The β-factor: measuring wireless link burstiness," in *Proc. ACM SenSys*, 2008.

[9] E. Ancillotti, R. Bruno, M. Conti, E. Mingozzi, and C. Vallati, "Trickle-L2: Lightweight link quality estimation through Trickle in RPL networks," in *Proc. IEEE WoWMoM*, 2014.

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

[11] F. Hermans *et al.*, "SoNIC: Classifying interference in 802.15.4 sensor networks," in *Proc. ACM/IEEE IPSN*, 2013.

[12] J. Ock, H. Kim, H.-S. Kim, J. Paek, and S. Bahk, "Low-power wireless with denseness: The case of an electronic shelf labeling system—design and experience," *IEEE Access*, vol. 7, pp. 163 887–163 897, 2019.

[13] D. Han and O. Gnawali, "Performance of RPL under wireless interference," *IEEE Commun. Mag.*, vol. 51, no. 12, pp. 137–143, Dec. 2013.

[14] C.-J. M. Liang, N. B. Priyantha, J. Liu, and A. Terzis, "Surviving Wi-Fi interference in low power zigBee networks," in *Proc. ACM SenSys*, 2010.

[15] A. Hithnawi, S. Li, H. Shafagh, J. Gross, and S. Duquennoy, "CrossZig: Combating cross-technology interference in low-power wireless networks," in *Proc. ACM/IEEE IPSN*, 2016.

[16] A. Hithnawi, H. Shafagh, and S. Duquennoy, "TIIM: Technology-independent interference mitigation for low-power wireless networks," in *Proc. ACM/IEEE IPSN*, 2015.

[17] X. Zheng, Z. Cao, J. Wang, Y. He, and Y. Liu, "ZiSense: Towards interference resilient duty cycling in wireless sensor networks," in *Proc. ACM SenSys*, 2014.

[18] J. Ock, J. Paek, and S. Bahk, "QBT: Queue-size based busy tones for protecting multihop low-power networks," in *Proc. IEEE MASS*, 2019.

[19] S. Kim and J. Paek, "Vulnerability of WiFi's noise floor calibration," *IEEE Internet Things J.*, 2020.

[20] R. Alexander *et al.*, "RPL: IPv6 routing protocol for low-power and lossy networks," RFC 6550, Mar. 2012.

[21] O. Gnawali *et al.*, "Collection tree protocol," in *Proc. ACM SenSys*, 2009.

[22] S. Kim *et al.* "Flush: A reliable bulk transport protocol for multihop wireless networks," in *Proc. ACM SenSys*, 2007.

[23] S. Jeong *et al.*, "MAPLE: Mobility support using asymmetric transmit power in low-power and lossy networks," *J. Commun. Netw.*, vol. 20, no. 4, pp. 414–424, 2018.

[24] H.-S. Kim, H. Kim, J. Paek, and S. Bahk, "Load balancing under heavy traffic in RPL routing protocol for low power and lossy networks," *IEEE Trans Mobile Comput.*, vol. 16, no. 4, pp. 964–979, Apr. 2017.

[25] H.-S. Kim, J. Paek, D. E. Culler, and S. Bahk, "PC-RPL: Joint control of routing topology and transmission power in real low-power and lossy networks," *ACM Trans Sensor Netw.*, vol. 16, no. 2, Mar. 2020.

[26] T. Clausen, J. Yi, and A. C. de Verdiere, "Loadng: Towards aodv version 2," in *Proc. IEEE VTC*, Fall 2012.

[27] D. S. J. D. Couto, D. Aguayo, J. Bicket, and R. Morris, "A high-throughput path metric for multi-hop wireless routing," in *Proc ACM MobiCom*, 2003.

[28] P. Thubert, "Objective function zero for the routing protocol for low-power and lossy networks (RPL)," RFC 6552, Mar. 2012.

[29] J. Polastre, R. Szewczyk, and D. Culler, "Telos: Enabling ultra-low power wireless research," in *Proc. ACM/IEEE IPSN/SPOTS*, 2005.

[30] H.-S. Kim, H. Cho, M.-S. Lee, J. Paek, J. Ko, and S. Bahk, "MarketNet: An asymmetric transmission power-based wireless system for managing e-Price tags in markets," in *Proc. ACM SenSys*, 2015.

[31] T. Gao *et al.* "Wireless medical sensor networks in emergency response: Implementation and pilot results," in *Proc. IEEE HST*, 2008.

[32] H.-S. Kim, J. Ko, D. E. Culler, and J. Paek, "Challenging the IPv6 routing protocol for low-power and lossy networks (RPL): A survey," *IEEE Commun. Surveys & Tut.*, vol. 19, no. 4, pp. 2502–2525, 2017.

[33] K. Srinivasan and P. Levis, "RSSI is Under Appreciated," in *Proc. EmNets*, 2006.

[34] H.-S. Kim, J. Paek, and S. Bahk, "QU-RPL: Queue Utilization based RPL for Load Balancing in Large Scale Industrial Applications," in *Proc. IEEE SECON*, 2015.

[35] D. Moss and P. Levis, "BoX-MACs: Exploiting Physical and Link Layer Bounrdaries in Low-Power Networking," Stanford Information Networks Group, Tech. Rep. SING-08-00, 2008.

[36] J. W. Hui, "An extended internet architecture for low-power wireless networks - design and implementation," Ph.D. dissertation, EECS Department, University of California, Berkeley, Sept. 2008.

[37] H.-S. Kim, H. Cho, H. Kim, and S. Bahk, "DT-RPL: Diverse bidirectional traffic delivery through RPL routing protocol in low power and lossy networks," *Comput. Netw.*, vol. 126, pp. 150–161, 2017.

[38] M. Mohammad, X. Guo, and M. C. Chan, "Oppcast: Exploiting spatial and channel diversity for robust data collection in urban environments," in *Proc. ACM/IEEE IPSN*, 2016.

[39] J. Hui and P. Thubert, "Compression format for IPv6 datagrams over IEEE 802.15.4-based networks," *RFC 6282*, Sept. 2011.

[40] T. Lee, M.-S. Lee, H.-S. Kim, and S. Bahk, "A synergistic architecture for RPL over BLE," in *Proc. IEEE SECON*, 2016.

[41] S. Duquennoy, B. Al Nahas, O. Landsiedel, and T. Watteyne, "Orchestra: Robust mesh networks through autonomously scheduled TSCH," in *Proc. SenSys*, 2015. 337–350.

[42] S. Jeong, H.-S. Kim, J. Paek, and S. Bahk, "OST: On-Demand TSCH Scheduling with traffic-awareness," in *Proc. IEEE INFOCOM*, July 2020, pp. 69–78.

[43] B. Kusy, D. Abbott, C. Richter, C. Huynh, M. Afanasyev, W. Hu, M. Brünig, D. Ostry, and R. Jurdak, "Radio diversity for reliable communication in sensor networks," *ACM Trans. Sensor Netw.*, vol. 10, no. 2, pp. 1–29, 2014.

[44] M. Sha, G. Hackmann, and C. Lu, "Energy-efficient low power listening for wireless sensor networks in noisy environments," in *Proc. ACM/IEEE IPSN*, 2013.

[45] T. Mandel and J. Mache, "Practical error correction for resource-constrained wireless networks: Unlocking the full power of the CRC," in *Proc. ACM SenSys*, 2013.

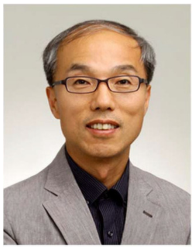
[46] W. Dong, J. Yu, and X. Liu, "CARE: Corruption-aware retransmission with adaptive coding for the low-power wireless," in *Proc. IEEE ICNP*, 2015.

[47] Y. Kim and J. Paek, "NG-RPL for efficient P2P routing in low-power multihop wireless networks," *IEEE Access*, vol. 8, pp. 182 591–182 599, Oct 2020.

[48] M. Park and J. Paek, "On-demand scheduling of command and responses for low-power multihop wireless networks," *Sensors*, vol. 21, no. 3, Jan. 2021.

Hyung-Sin Kim received the B.S. degree in electrical engineering and the M.S. and Ph.D. degrees in electrical engineering and computer science (EECS) from Seoul National University (SNU), Seoul, South Korea, in 2009, 2011, and 2016, respectively, all with outstanding thesis awards. He was a Postdoctoral Scholar at Network Laboratory (NETLAB), SNU, until August 2016 and Real-time, Intelligent, Secure, Explainable systems (RISELab), University of California, Berkeley, until August 2019, and a Software Engineer at Google Nest until February 2020. He received the Qualcomm Fellowship in 2011 and the National Research Foundation (NRF) Global Ph.D. Fellowship and Postdoctoral Fellowship in 2011 and 2016, respectively. He is currently an assistant professor at Graduate School of Data Science, SNU. His research interest includes Internet of Things and ambient artificial intelligence.

Jeongyeup Paek received his B.S. degree from Seoul National University in 2003 and his M.S. degree from University of Southern California in 2005, both in Electrical Engineering. He then received his Ph.D. degree in Computer Science from the University of Southern California (USC) in 2010. He worked at Deutsche Telekom Inc. R&D Labs USA as a Research Intern in 2010, and then joined Cisco Systems Inc. in 2011 where he was a Technical Leader in the Internet of Things Group (IoTG), Connected Energy Networks Business Unit (CENBU, formerly the Smart Grid BU). At Cisco, he was one of the Lead Engineers for Connected Grid Mesh (CG-Mesh) system and WPAN software. In 2014, he was with the Hongik University, Department of Computer Information Communication as an Assistant Professor. He is currently an Associate Professor at the School/Department of Computer Science and Engineering, Chung-Ang University, Seoul, Republic of Korea. He is an Editor for the Journal of Communications and Networks (JCN), an editorial board member of Sensors, and also an IEEE senior member.



Saewoong Bahk is a Professor at Seoul National University (SNU). He served as Director of the Institute of New Media and Communications during 2009-2011. Prior to joining SNU, he was with AT&T Bell Laboratories as a Member of Technical Staff from 1991 to 1994 where he had worked on network management. He received KICS Haedong Scholar Award in 2012. He has been leading many industrial projects on 3G/4G/5G and IoT connectivity supported by Korean industry, and published more than 200 technical papers and holds more than 100 patents. He has been serving as Chief Information Officer (CIO) of SNU, General Chair of IEEE DySPAN 2018 (Dynamic Spectrum Access and Networks), General Chair of IEEE WCNC 2020 (Wireless Communication and Networking Conference), and Director of Asia-Pacific region of IEEE ComSoc. He is President-Elect of the Korean Institute of Communications and Information Sciences (KICS), TPC Chair for IEEE VTC-Spring 2014, General Chair of JCCI 2015. He is co-EIC of the Journal of Communications and Networks (JCN) and an Editor of IEEE Network Magazine. He was on the editorial board of Computer Networks Journal (COMNET) and IEEE Transactions on Wireless Communications (TWireless). He is a Member of the National Academy of Engineering of Korea (NAEK), an IEEE Senior Member, and a Member of Whos Who Professional in Science and Engineering. He received B.S. and M.S. degrees in Electrical Engineering from SNU in 1984 and 1986, respectively, and the Ph.D. degree from the University of Pennsylvania in 1991.