MAPLE: Mobility Support using Asymmetric Transmit Power in Low-Power and Lossy Networks

Seungbeom Jeong, Eunjeong Park, Dongyeon Woo, Hyung-Sin Kim, Jeongyeup Paek, and Saewoong Bahk

Abstract: **With the proliferation of emerging Internet of Things (IoT) devices and applications, mobility is becoming an integral part of low-power and lossy networks (LLNs). However, most LLN protocols have not yet focused on the support for mobility with an excuse of resource constraints. Some work that do provide mobility support fail to consider radio duty-cycling, control overhead, or memory usage, which are critical on resource-limited low-power devices. In this paper, we introduce *MAPLE*, an asymmetric transmit power-based routing architecture that leverages a single resource-rich LLN border router. *MAPLE* supports mobility in duty-cycled LLNs using received signal strength indicator *(RSSI) gradient field*-based routing. High-power transmission of the gateway not only allows LLN endpoints to be synchronized for low duty-cycle operation, but also establishes an RSSI gradient field which can be exploited for opportunistic routing without a need for any neighbor or routing table. This eliminates the scalability problem due to memory limitation, and provides a responsive routing metric without control overhead. *MAPLE* also addresses the RSSI local maximum problem through local adaptation. We implement *MAPLE* on a low-power embedded platform, and evaluate through experimental measurements on a real multihop LLN testbed consisting of 31 low-power ZigBee nodes and 1 high-power gateway. We show that *MAPLE* improves the performance of mobile devices in LLN by 27.2%/55.7% and 17.9% in terms of both uplink/downlink reliability and energy efficiency, respectively.**

Index Terms: **Asymmetric transmit power, IEEE 802.15.4, low-power lossy network (LLN), mobility, routing.**

I. INTRODUCTION

LOW-POWER and lossy network (LLN), multihop wireless network composed of resource-constrained embedded devices, has been used for a variety of applications including smart grid automated metering infrastructure (AMI), environmental monitoring, and wireless sensor network (WSN). Furthermore, with the emergence of Internet of Things (IoT) and cyber-physical systems (CPS), LLN is now going into a new phase for smart and daily life applications which include medical care services [1]–[3], smart market maintenance [4], [5], networked robots [6], [7], and more. A key challenge in many of these emerging applications is that they incorporate not only stationary but also *mobile nodes*. As an example, a hospital network can be connected with sensing and actuating devices on mobile patients and patient beds, which enables remote monitoring of medical signals. In smart market applications, mobile shopping carts are connected to an LLN, which is used for real-time advertisement of hot deals, cart location tracking, and virtual fencing. Market staffs can also carry low-power portable terminals for reporting status of inventory/stock and market condition.

For over a decade, LLN research community has elaborated network layer protocols for energy efficiency and high reliability on resource-constrained devices. As a cornerstone work, 6LoWPAN [8] and IPv6 routing protocol for LLNs (RPL) [9] were designed and standardized, enabling multihop IoT networks. Numerous network protocols based on RPL have been devised [4], [10], [11], such as ORPL [10], surpassing performance of RPL in terms of delay, energy-consumption, and reliability. However, although these protocols have been making progress gradually under the assumption of stationary network, they cannot be apparently adopted in LLNs with mobility, due to lack of providing any specific operation for mobility and identifying mobile nodes [12]–[14]. Several studies [12]–[14] confirm that protocols designed for stationary network, such as RPL, experience significant performance degradation when operating with mobile devices.

There have been several work [12], [15]–[19], such as LLN On-demand Ad-hoc Distance-vector Routing Protocol - Next Generation (LOADng) [16], designed to provide seamless connectivity for mobile nodes in LLN. However, each of these work has at least one of the limitations among the followings. (1) They disregard the protocol operation with radio duty-cycling, which is one of the most critical characteristics of LLN with battery-powered nodes, (2) up-to-date neighbor or routing information is required according to topology dynamics caused by mobility, adding a significant network overhead, and (3) even if they showed performance improvement in idealistic simulator-based evaluation, the same has not been shown in practice via real experiments.

This work investigates how to provide bi-directional connectivity between the LLN border router (LBR, also referred to as

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 in part by a grant to Bio-Mimetic Robot Research Center Funded by Defense Acquisition Program Administration, and Agency for Defense Development (UD160027ID), in part by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIP) (No. 2015R1A2A2A01008240), and also in part by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (NRF-2017R1D1A1B03031348).

S. Jeong, E. Park, D. Woo, and S. Bahk are with the Department of Electrical Engineering and Computer Science, INMC, Seoul National University, Seoul, Republic of Korea, email: {sbjeong, ejpark, dywoo}@netlab.snu.ac.kr and sbahk@snu.ac.kr.

H.-S. Kim is with the Department of Electrical Engineering and Computer Science, University of California, Berkeley, CA, USA, email: hs.kim@berkeley.edu.

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

S. Bahk is the corresponding author.

Digital Object Identifier

'gateway') and each *mobile node* in LLN, both reliably and energy efficiently. In contrast to previous approaches, we exploit *asymmetric transmit power (ATP) architecture* [20] for LLN with mobile endpoints. It has been more common place that LLN routers are plugged in, especially in indoor environments which have abundant outlets [2], [4], and only the endpoints are battery-operated. It is also possible for smart grid applications such as automated metering infrastructure (AMI) where power is a given [21]. In this context, recent LLN protocols, such as Thread [22] and BLEmesh [23], even 'force' LLN routers to be plugged in. Without an energy constraint, these plugged-in routers can utilize much higher transmit power than battery-powered endpoint devices (e.g., 30 dBm vs. 0 dBm), which allows asymmetric transmit power to be a viable design choice in this regime.

As an example of ATP-based applications, a smart market [4], [5] uses a high-power gateway for disseminating price information and advertisements. At the same time, it collects stock status from portable terminals carried by mobile staffs, or locations of shopping carts. In a hospital network, low-power sensing and actuating devices report patient's condition such as vital signals. The information gathers in a single network gateway, and it can maintain and process the information. In addition, when an emergency occurs, it can control remote medical devices (attached to patients) immediately using high-power transmissions.

ATP-LLN provides a *single hop downlink* (from gateway to endpoints) and *multihop uplink* (from endpoints to gateway) architecture. A number of studies have explored its potential [4], [5], [20], showing improvement in downlink reliability and energy consumption when all nodes are static. Building on these previous work, *we argue that with a careful design, the ATP architecture is also useful for supporting mobile nodes*. To this end, we design *MAPLE*, an ATP-based LLN protocol that supports mobile endpoints by providing reliable and energy-efficient bi-directional communication under dynamic topology variation.

In our *MAPLE* design, each low-power node expects to receive a downward packet *directly* from the high-power gateway (i.e., single hop). To improve reliability, each node sends a negative acknowledgement packet (NACK) when detecting a missed downward packet, which triggers *local retransmissions* from its neighbor nodes. Unlike the previous approach in [20], *MAPLE*'s neighbor forwarding works *without topology information*. On the other hand, low-power uplink transmissions require multihop routing. To obtain path diversity in dynamic mobile environments, *MAPLE* uses *opportunistic routing* [10] where a sender simply broadcasts packets and each receiver decides whether to relay the packets by considering its and the sender's uplink routing metric. For the routing metric, *MAPLE* uses *RSSI of the gateway's high power transmissions* given that RSSI (Received Signal Strength Indicator) generally decreases with distance (i.e., providing indirect geographical information). A single high-power transmission can update all nodes' routing metric at once, creating an *RSSI gradient field* in the network, and low-power nodes are completely free from a control packet overhead and routing table size limitation.

In doing so, we address well-known concerns about the RSSI measurements; it is unstable and time varying [24]–[26]. To this end, *MAPLE* obtains a *stable and interference-free RSSI value* from each packet of the gateway by using a high-resolution multi-sampling technique [27], [28], and updates this value with periodic high-power beacon transmissions [4]. Furthermore, we also tackle the *local maximum problem* [29], a representative problem in geographical routing, through dynamic and distributed adaptation of the RSSI metric. We implement *MAPLE* on ContikiOS and extensively evaluate the performance of *MAPLE* on a real LLN testbed in both stationary and mobile scenarios. Our results show that *MAPLE* significantly outperforms representative LLN protocols, i.e., RPL [9], ORPL [10], and LOADng [16], in terms of reliability and energy consumption.

The contributions of this work are threefold:

- We propose ATP-based *MAPLE* system for reliable and low-power bidirectional communication in mobile LLNs. It provides single-hop downlink based on the high-power gateway and multihop uplink based on an RSSI gradient field based opportunistic routing.
- We design several mechanisms to support this system architecture: (1) NACK-based local downlink retransmission improves downlink reliability without topology information. (2) High-resolution multi-sampling makes RSSI measurement stable enough to be used as a routing metric. (3) RSSI adaptation addresses the local maximum problem of the RSSI-based gradient field.
- We implement *MAPLE* on real embedded devices and experimentally evaluate its performance against the standard RPL, ORPL, and LOADng on a real 32-node testbed. Our evaluation shows that *MAPLE* achieves significantly better packet delivery performance and route adaptation according to topology change than RPL.

The remainder of this paper is organized as follows: We first discuss the related work and a brief background in Section II. In Section III, we present the design of our proposed scheme, *MAPLE*, and elaborate on its main functional blocks. We discuss the implementation details and present the evaluation results in Section IV. We conclude the paper in Section V.

II. BACKGROUND AND RELATED WORK

In this section, we review standard LLN protocols and their derivatives, and then introduce several work we were motivated by. They are summarized in Table 1.

A. Representative Standard LLN Protocols

This section presents two representative LLN routing protocols as our benchmark, RPL (IETF standard) [9] and LOADng (IETF draft) [16], and their related work.

RPL and Mobility Support: RPL [9] is a routing protocol for low-power IPv6 networks which enables IPv6 Internet connectivity to embedded devices by providing reliable routes over lossy wireless links. Several studies [12]–[14], however, found that RPL suffers significant performance degradation when operating with mobile endpoints since it was not designed with mobility in mind. To alleviate this problem, ME-RPL [19] gives lower priority to mobile parent candidates than static candidates when choosing a preferred parent. In addition, when par-

Table 1. Feature comparison of *MAPLE* with related work.

	RPL [9]	MoMoRo [12]	mRPL [15]	LOADng [16]	MarketNet [4]	ORPL [10]	***MAPLE***
Mobility support	×	◯	◯	◯	×	×	◯
Duty-cycling	◯	×	×	×	◯	◯	◯
Asymmetric transmit power	×	×	×	×	◯	×	◯
Non neighbor-information	×	×	×	×	×	×	◯
Spatial diversity	×	×	×	×	×	◯	◯

ent changes occur frequently, DIS transmission interval is reduced for prompt neighbor discovery. Ko *et al.* introduced MoMoRo [12], which detects route disconnections based on uplink packet losses and quickly gathers neighbor information by requesting a unicast reply. It can find out neighbors with a good link based on a fuzzy estimator. In mRPL [15], a mobile node broadcasts a batch of DIS messages when the RSSI from its parent drops, triggering replies from its neighbors. The mobile node measures RSSI from the neighbors and selects the neighbor with a good RSSI as a preferred parent. Gaddour *et al.* used position information for mobile routing in [18] where corona ID is defined as the minimum of reachable hop distances from the DAG root.

However, most of these protocols are designed assuming *no radio duty-cycling* for more responsiveness, sacrificing low-power operation. If a duty-cycled MAC is adopted under these protocols, it necessarily delays the update of routing costs, resulting in more packet losses and control overhead [15]. Moreover, proposals in [12], [15], [18], [19] require maintaining *up-to-date topology information*. This incurs a significant control overhead to keep track of topology changes caused by mobile nodes. Lastly, evaluations of [18], [19], [30] are performed only on simulators, which cannot show their feasibility in the unpredictable real world.

LOADng: LOADng [16] is a simplified version of well known AODV (Ad hoc On-demand Distance Vector Routing) [31] to support mobile LLNs. Like AODV, LOADng discovers a route between a source and a destination based on flooding of routing packets from the source. However, Clausen *et al.* showed that LOADng suffers from flooding overhead, particularly in applications with collection traffic [17]. To resolve this problem, the authors designed LOADng-CTP, where only the sink (not a source but a destination) floods routing messages to enable each data source to obtain a path towards the sink. It outperforms LOADng with respect to delivery ratio, overhead, and delay. However, both LOADng and LOADng-CTP are evaluated without a duty-cycled MAC and only with simulations.

B. Background of MAPLE

This section describes three main bases that our *MAPLE* design builds on to support mobility in LLNs: (1) ATP architecture, (2) opportunistic routing, and (3) RSSI-based routing metric.

Asymmetric Transmit Power Architecture: As opposed to traditional symmetric (or homogeneous) transmit power networks, ATP architecture is another option for LLNs, given that plugging in LLN routers has been more commonplace in many applications [21] and network protocols [22], [23]. These wall-powered LLN routers can use higher transmit power than battery-powered nodes, creating an ATP-based network.

Benefiting from the ATP architecture, SHDP [20] provides a multihop uplink but single-hop downlink network where a wall-powered gateway uses higher transmission power than battery-powered nodes. It achieves reliable downward packet delivery by using local acknowledgement and neighbor forwarding. MarketNet [4], [5] separates low-power uplink transmissions from high-power downlink transmissions in the time domain, making uplink communications not interfered by high-power gateway. However, both SHDP and MarketNet require up-to-date neighbor information and neither considered mobility. While our *MAPLE* builds on the ATP architecture, it uses *opportunistic routing* for uplink and provides reliable downlink *without topology information*.

Opportunistic Routing: Traditional routing in multihop networks is composed of a series of unicast transmissions along a path. Instead, opportunistic routing has been introduced in which multiple potential forwarders are used by anycast transmission for spatial diversity [10], [32], [33]. ORW [33] brought opportunistic routing in duty-cycled networks for many-to-one (uplink) communication by using a unique routing metric called expected duty-cycle (EDC). ORPL [10] augments the ORW design by providing support for one-to-many (downlink) and any-to-any communication with a simplified routing set by using a bloom filter. However, these opportunistic routing protocols have not been evaluated in mobile scenarios. Given that the EDC metric cannot be updated fast, *MAPLE*'s opportunistic uplink routing uses an *RSSI gradient field*, which can be fast updated in mobile LLNs owing to unique characteristics of the ATP architecture.

RSSI as a Routing Metric: As an uplink routing metric, *MAPLE* exploits RSSI measured from the high power transmissions of the gateway since it is simple to measure and generally decreases with distance from the gateway (i.e., providing geographical information). However to this end, the local maximum problem [29], a representative issue in geographical routing, should be handled. Moreover, it has been shown that a single RSSI value does not ideally represent distance due to random attenuation phenomena such as multipath fading, shadowing, and external interference [24], [25]. OR-RSSI [34] is similar to *MAPLE* in that it uses opportunistic uplink routing based on an RSSI gradient, which is generated by periodic high-power beacons. However, it assumes that the RSSI gradient is ideal without addressing any of the above issues. In contrast, *MAPLE* handles the local maximum problem and fast stabilizes RSSI values by using a multi-RSSI sampling technique [27].

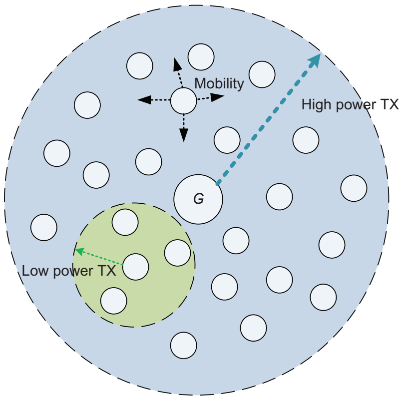


Fig. 1. Network model of *MAPLE*.

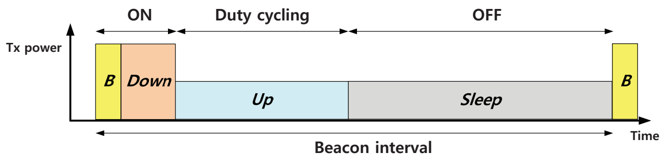


Fig. 2. Superframe structure of *MAPLE*.

III. MAPLE DESIGN

In this section, we describe *MAPLE*, Asymmetric Transmit Power-based Mobile LLN architecture that provides mobility support with low radio duty-cycle. Fig. 1 depicts the system model we consider. There are a large number of low-power nodes and a single high-power gateway node (*G*). We assume that each node could be mobile, but is always located within high-power transmission coverage of the gateway. Thus transmission of gateway reaches all the nodes in a single hop. However, uplink communication works in a multihop manner due to low transmit power of low-power nodes.

As shown in Fig. 2, *MAPLE* repeats a superframe every beacon interval, which is divided into four periods: (1) Beacon period (*B*), (2) downlink period (*Down*), (3) uplink period (*Up*), and (4) sleep period (*Sleep*). First, the gateway transmits a beacon in the *B* period and downward packets in the *Down* period, both with high transmit power. During these two periods, the other low-power nodes are not allowed to send any packet to ensure that the gateway's transmissions are free from contention and collision [4]. Instead, they continuously listen to the medium to receive packets from the gateway. In the subsequent *Up* period, the low-power nodes send and receive upward packets on top of a duty-cycling MAC protocol for low-power operation[1]. At the end of *Up* period, a low-power node stops duty-cycling and turns off its radio for energy saving. During this *Sleep* period, there is no packet communication. After the *Sleep* period, every low-power node turns its radio on again to receive next beacon at the right time.

[1]Any of synchronous [35], [36] or asynchronous duty-cycle MAC protocols [37]–[39] could be used in *MAPLE*. Without loss of generality, we use the ContikiMAC [37], a representative asynchronous MAC protocol for LLN.

Except for the *B* period, lengths of the other periods could be controlled by the gateway according to traffic generation rate and network size. For instance, *Sleep* duration might be set to zero in order to minimize channel contention in the *Up* period when intensity of uplink traffic is high. On the contrary, *Sleep* period can be made longer for ultra low-power operation when uplink traffic load is light. The remainder of this section provides detailed descriptions of *B*, *Down*, and *Up* periods.

A. Beacon and Beacon Period

In the *B* period, the gateway transmits beacons using high transmit power so that all nodes can receive them. The beacon has three major roles. Firstly, it is used for *network-wide time synchronization* of all the nodes. Every beacon includes durations of *B*, *Down*, *Up*, and *Sleep* periods, and a node willing to join the *MAPLE* network must wait and listen for the first beacon reception. Once a node receives a beacon correctly, it can be *synchronized* and share the superframe structure illustrated in Fig. 2. Secondly, the beacon includes the destinations of the downward packets which will be sent in the subsequent *Down* period. The destination information is used for NACK-based local retransmission of downward packets (explained in Section III.B). In our experiments, we have used 5 seconds as the beacon interval.

Lastly, the beacon is used to generate an RSSI-based gradient field throughout the network. Whenever each low-power node receives a high-power packet from the gateway, it records the RSSI ($RSSI_G$). Ideally, the closer a node is to the gateway, the larger $RSSI_G$ it obtains. *MAPLE* exploits this $RSSI_G$-gradient field for multihop opportunistic routing in the *Up* period, as described in Section III.C.

B. Downlink Transmission: Local NACK and Retransmission

In the *Down* period, the gateway transmits downward packets with a high transmit power. Low-power nodes are not allowed to transmit any packet during this period to avoid packet collision [4]. While the single-hop high-power downlink transmission based on the ATP architecture removes the need for downlink routing, a subtle issue still remains: how to acknowledge a downward packet for reliable packet delivery. This is because a low-power destination node is not likely to deliver an acknowledgement (ACK) to the gateway in a single hop due to its limited transmit power.

An ACK may be forwarded towards the gateway through a *multihop* route [40], [41], which creates a significant communication overhead. Another approach is for the destination node to send ACK packets *locally* to its neighbors [4], [20]. When the destination's neighbors overhear a downward packet but do not receive a local ACK from the destination, they locally retransmit the downward packet on behalf of the gateway. However, this requires up-to-date neighbor information which is hard and expensive to get in mobile LLNs since topology changes continuously. Furthermore, this approach creates large number of (potentially redundant) local ACKs, which may be an overkill under high-power gateway transmissions where downlink loss rates are typically low for most endpoints.

For these reasons, our solution is to send a *local negative-ACK (NACK)* for a downward packet loss. As described in Sec-

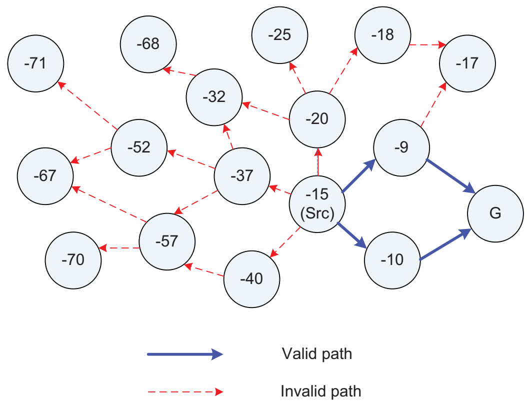


Fig. 3. Opportunistic uplink routing using RSSI-gradient field.

tion III.A, a low-power node knows the destinations of all downward packets that are sent in a *Down* period by receiving a beacon in the previous *B* period. When a low-power node expects to receive a downward packet for itself in a *Down* period but misses it, the node sends a local NACK. Given that the *Down* period is only for the gateway's transmission, local NACK is transmitted in the following *Up* period with a low transmit power.

At the same time, each low-power node overhears all downward packets in a *Down* period and holds them until the end of the subsequent *Up* period. The gateway also holds these downward packets. Upon receiving a NACK in the *Up* period, a low-power node (or the gateway) searches the stored downward packets to check if any of them is destined to the NACK sender. If it has one, it forwards the downward packet to the NACK sender with a low transmit power. The node can try the local retransmission several times until receiving an ACK from the destination. This approach improves downlink reliability without topology information nor redundant ACK transmissions.

When a node fails to receive a beacon in a *B* period, it can still use the superframe structure given that the time synchronization is valid for a while. But it does not find out the destination information of downlink transmissions in the following *Down* period. In this case, the node assumes that a downward packet towards itself is lost and transmits a NACK in the following *Up* period. This triggers local retransmissions from the neighbors if the downward packet loss really happens, providing reliable downward packet delivery regardless of a beacon loss.

MAPLE also supports network-wide broadcast service from the gateway. To this end, the gateway can inform all low-power nodes of the existence of a broadcast packet by including IPv6 link-local broadcast address in the beacon, instead of a downward unicast destination. The rest of operation with local NACK and retransmission is the same as the unicast case.

Lastly, like high-power beacon transmissions, each high-power downlink transmission is also used to update $RSSI_G$ for uplink routing, regardless of its destination. This enables a low-power node to update its RSSI-based routing metric frequently even if it fails to receive a recent beacon.

C. Uplink Transmission: RSSI Gradient-based Routing

In the *Up* period, the gateway mainly listens to the medium, but can transmit ACKs for uplink traffic or perform local retransmissions for NACKs, both with a low level of transmit power. Meanwhile, low-power nodes send/receive packets with duty-cycle for energy saving.

For multihop uplink transmissions, *MAPLE* borrows opportunistic routing concept in ORW [10] but uses a gradient-field of $RSSI_G$ rather than the EDC metric. Specifically, each packet sender piggybacks its $RSSI_G$ value in an upward packet and simply broadcasts it. Given that large $RSSI_G$ indicates high proximity to the gateway, when a node receives an upward packet and it has higher $RSSI_G$ than the packet sender, it sends an ACK and forwards the packet. Fig. 3 shows an example of the $RSSI_G$-gradient based opportunistic forwarding. The number in each node indicates $RSSI_G$ value obtained through a previous beacon or downward packet reception (The unit is dBm). Although there are five neighbors of data source (expressed as Src), only two nodes among them have higher $RSSI_G$ values (-9 dBm and -10 dBm) than $RSSI_G$ of the source (-15 dBm), and thus are valid candidates for data forwarding.

Compared to the state-of-the-art LLN routing protocols such as RPL [9] and LOADng [16], $RSSI_G$-based opportunistic routing of *MAPLE* has three primary strengths. Above all, *MAPLE*'s opportunistic routing requires each node to maintain only its $RSSI_G$ value without any neighbor information, letting a resource-constrained device keep low and constant memory footprint regardless of network size or density. In addition, $RSSI_G$ is updated solely based on the gateway's high power transmissions, which enables a low-power node to maintain a valid routing metric without any control packet overhead. Lastly, given that one high-power transmission can update $RSSI_G$ of all nodes, the gateway can freely adjust periodicity of $RSSI_G$ update depending on mobility scenarios.

However, an $RSSI_G$-gradient has the local maximum problem which is common in geographical routing. It is even more so when applying the $RSSI_G$-gradient in a real wireless environment since RSSI is highly variable even if all nodes are stationary due to multipath fading, shadowing, and various interference. When a route with a non-monotonic $RSSI_G$-gradient is encountered, uplink packets may not be relayed anymore in the middle of the path, leading to packet losses. *MAPLE* addresses this phenomenon by adjusting $RSSI_G$ intentionally, but carefully. This is explained in Section III.D.

D. Local Maximum Problem and RSSI Adaptation

In a free space where every node has a line-of-sight (LOS) link with the gateway, $RSSI_G$ can be used to approximately indicate the straight-line distance to the gateway using free-space RF propagation models. Meanwhile, *MAPLE* interprets this information as the distance along the routing path. If the network density is high enough to provide sufficient number of forwarding nodes such that linear shaped routing path can be obtained for any node, as depicted in Fig. 4(a), this interpretation is valid.

However, this interpretation might not hold when the path is curved like Fig. 4(b), which is highly probable when the node density is low. Furthermore, it is also invalid if there are ob-

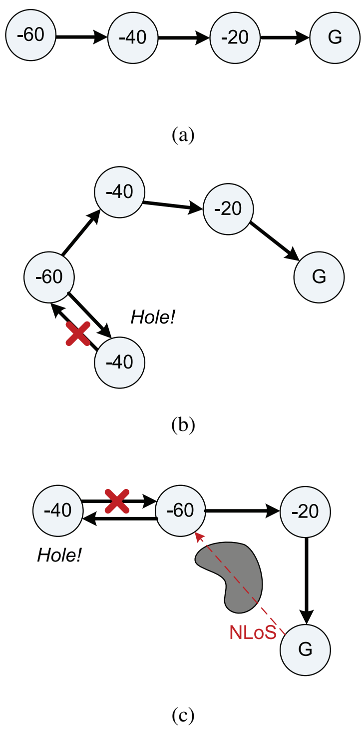


Fig. 4. Ideal case and local maximum problems in RSSI gradient based routing: (a) Ideal RSSI gradient, (b) local maximum problem 1, and (c) local maximum problem 2.

stacles which block off LOS with the gateway as shown in Fig. 4(c). In an ATP network where LOS and Non-LOS (NLOS) nodes with the gateway coexist, $RSSI_G$ field is not generated with a monotonic gradient along the desired path. For example, in Figs. 4(b) and 4(c), a packet which is generated by or sent to a node with *local maximum RSSI* (denoted as a '*Hole!*' in the figures) cannot be forwarded any further towards the gateway as if it gets stuck at a dead-end, or in what we call a *hole*. This problem makes the hole node suffer from consecutive packet losses either at the link since there is no forwarder with better $RSSI_G$, or at the queue due to memory overflow as resource-constrained nodes have very small size queues.

To resolve this *hole problem*, we design a light-weight but effective algorithm for $RSSI_G$ adaptation at the hole node. Each low-power node maintains the most recent transmission history list ($list_{tx}$). As a first-in first-out (FIFO) list, $list_{tx}$ is updated whenever an anycast transmission is completed. If the transmission is acknowledged, which means there is at least one neighbor who offers routing progress as the next hop, transmission success (S) is recorded in $list_{tx}$. On the other hand, when there is no incoming ACK packet before a timeout (which is, for example, a whole sleep interval in asynchronous sender-based duty-cycle MAC protocols), transmission failure (F) is added to the list. If the number of F (N_F) exceeds a pre-specified threshold, N_F^{th}, the node recognizes itself as a *hole*. Then it tries to escape from the hole by lowering its $RSSI_G$ deliberately. With N_F above N_F^{th}, $RSSI_G$ is reduced by Δ_{hole} every packet loss in the link. The updated RSSI in the hole, $RSSI_{hole}$, is expressed as,

$$RSSI_{hole} = \max\left(RSSI_G - (N_F - N_F^{th})\Delta_{hole},\ RSSI_{\min}\right),$$

where $RSSI_{\min}$ is the minimum RSSI value available in a radio. In our implementation, N_F^{th} and Δ_{hole} are 2 and 20 dBm, respectively.

E. Implementing Reliable $RSSI$ Capture

Even though *MAPLE* handles the hole problem, instantaneous RSSI is well known to be unpredictable in wireless links due to multipath fading, external/internal interference and various environmental factors. On the other hand, the primary principle for uplink routing of *MAPLE* is to adapt an $RSSI_G$-gradient field to physical topology changes like node's mobility, while minimizing the effect of wireless unpredictability. As one of possible approaches, more $RSSI_G$ samples could be collected by increasing the number of high-power transmissions. Then, the average $RSSI_G$ can be used for an uplink routing metric. However, this inevitably brings about more energy consumption.

We consider an IEEE 802.15.4 compliant radio, i.e., CC2420 [42], as an implementation example. During a packet reception, a 2-byte frame check sequence (FCS) follows the last MAC payload byte. FCS is automatically generated and verified by the hardware when the `MODECTRL0.AUTOCRC` control bit is set[2]. Then the first FCS byte is replaced with the 8-bit RSSI value, which can be read by the upper layer. In CC2420, this RSSI value is measured over the first 8 symbols following the start of frame delimiter (SFD), and can be obtained from the `RSSI.RSSI_VAL` register.

In our system, instead of reading only the first byte of FCS for RSSI, we obtain *multiple* RSSI values from a *single* packet in a similar way to [27]. The RSSI value in `RSSI.RSSI_VAL` is always averaged over 8 symbol periods (128 microseconds) and continuously updated for each symbol after RSSI has become valid. We let a low-power node detect an SFD interrupt for the *B* and *Down* periods and then immediately read and store `RSSI.RSSI_VAL` register value every 8 symbols. Given that, following SFD, a frame length byte and IEEE 802.15.4 MPDU (maximum size of 127 bytes) come, 32 RSSI samples can be acquired at most from a single packet. In our implementation, we use the beacon size of 51 bytes and let 10 RSSI samples obtained. We average these RSSI samples and use the averaged value as $RSSI_G$.

IV. PERFORMANCE EVALUATION

In this section, we evaluate *MAPLE* experimentally through a prototype implementation, and compare it with RPL, ORPL, and LOADng in terms of reliability and energy efficiency. We evaluate on a network with and without mobility using three scenarios; 1) static network (no mobility) 2) a single mobile device, and 3) three mobile devices. In addition, we also run Cooja simulations to evaluate and compare the performance under high level of network mobility.

[2]It is recommended to always have this control bit enabled, except possibly for debug purposes [42].

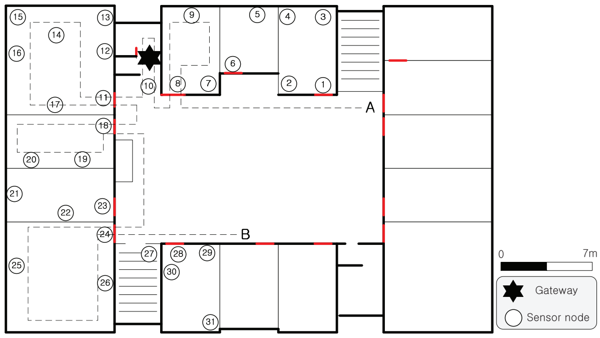


Fig. 5. Topology map of indoor 32-node LLN testbed with a moving path for node mobility.

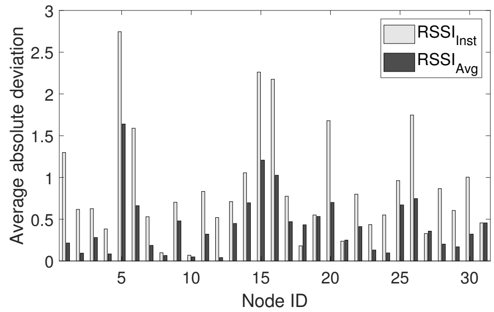


Fig. 6. Average absolute deviation for instantaneous RSSI and average RSSI.

A. Methodology and Experiment Setup

Fig. 5 presents the topology of our testbed where a total of 32 TelosB clone devices are deployed with one node acting as the gateway (or root) of the network. For *MAPLE*, we use the MTM-CM3300MSP device as the high-power gateway, which is similar to a TelosB with a 10 dB power amplifier. The other low-power nodes use a transmit power of -20 dBm while the high-power root uses 10 dBm. For other compared schemes, the gateway uses the same transmit power as other nodes. This leads to a maximum diameter of 5 hops in case of RPL.

We consider a bidirectional traffic scenario. Each node generates an uplink packet every 75 seconds, while the gateway generates equal rate of downlink packet per node, resulting in average inter-packet interval of 2.5 seconds in both directions. In our experiments with *MAPLE*, within the beacon interval of 5 seconds, the duration of *Down* period is 20 ms to accommodate two downlink packets. Remaining time is used for *Up* period (with no *Sleep* period). In all experiments, the application payload is 24 bytes, which is carried in UDP datagrams over 6LoWPAN. All our experiments were done on Zigbee/IEEE 802.15.4 channel 26 (i.e., no WiFi interference) and in a stable channel environment with minimal external interfering factors, such as uncontrolled human movement and environment changes. Unless specified, all our results are an average of three runs of 1-hour experiments from different times of the day.

For comparison with state-of-the-arts, we use RPL [9], ORPL [10], and LOADng [16]. All these protocols including *MAPLE* are implemented on top of ContikiMAC [37] in ContikiOS [43]. We use Contiki's default values for the number of transmission attempts and duty cycle rate, 5 and 8 Hz, respectively. Note that the gateway does not duty-cycle in order to handle all network traffic. ContikiMAC has a phase-lock mechanism, where a sender records wake-up phase of its neighbors, and uses it for timely transmission in an energy-efficient manner. Phase-lock can be used for unicast of RPL or LOADng, neither broadcast nor anycast of ORPL and *MAPLE*.

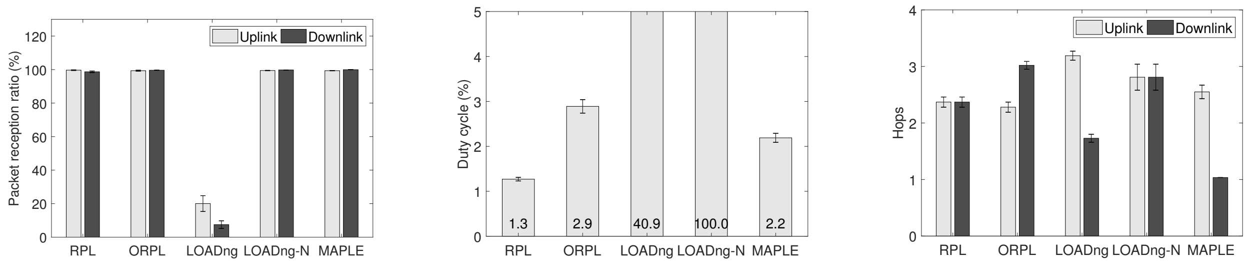
With this configuration, we first check the total size of volatile memory for each protocol. RPL and ORPL consume 8.6 and 8.4 kBytes, respectively, and LOADng uses 9.5 kBytes of RAM. On the other hand, *MAPLE* spends only 7.3 kBytes of memory as it does not need to maintain neighbor or routing information. This result verifies *MAPLE* outperforms state-of-the-arts with regard to memory footprint.

B. Static Network

We first evaluate the effect of multiple RSSI sampling in a single packet, which is used in *MAPLE* for reliable RSSI capture. To this end, we run *MAPLE* with beacon interval of 5 seconds in the testbed shown in Fig. 5. Whenever a low-power node receives a beacon, it records two kinds of RSSI values, $RSSI_{Inst}$ and $RSSI_{Avg}$. $RSSI_{Inst}$ indicates the instantaneous RSSI value read from the first FCS byte of the received beacon. On the other hand, $RSSI_{Avg}$ is the average of 10 RSSI samples stored after an SFD interrupt based on our approach described in Section III.E. The experiment ran for 4-hours, leading to about 2,900 beacon receptions. We observed that the difference between the mean values of $RSSI_{Inst}$ and $RSSI_{Avg}$ is marginal less than 1%. However, their deviations over time are quite different. Fig. 6 presents the average absolute deviations over time for each node. For the $RSSI_{Inst}$ measurements, in the worst case (i.e., node 5), the deviation is close to 3. We stabilized this unpredictable RSSI by obtaining and averaging multiple RSSI samples, resulting in 40% improvement.

Using our stabilized $RSSI_{Avg}$, now we evaluate the performance of *MAPLE* against RPL, ORPL, and LOADng in a static network. Fig. 7(a) shows the end-to-end packet reception ratio (PRR) for uplink and downlink traffic. We observe that LOADng is not suited for a duty-cycled LLN, showing severe PRR degradation. This result comes from the way of its reactive route search based on network flooding. A broadcast message for route request occupies the medium within a whole sleep interval (i.e., 125 ms in 8 Hz duty-cycle rate). What is worse, it is propagated throughout the network hop-by-hop. This significant overhead incurs network congestion extremely, leading to excessive energy consumption, as illustrated in Fig. 7(b) as a metric of duty-cycle, the portion of radio on-time.

Apart from duty-cycling MAC, in order to see routing performance of LOADng solely, we also examine LOADng without duty-cycling. Instead of ContikiMAC, we build LOADng on NullRDC (a simple MAC implementation without duty-cycling, provided by Contiki), denoted as *LOADng-N* in Fig. 7. Figs. 7(a) and 7(b) show that LOADng-N has comparable PRR with RPL and ORPL with the cost of 100% duty-cycle. RPL shows the lowest duty-cycle with nearly perfect PRR, owing to the phase-lock mechanism used in ContikiMAC for unicast transmissions. ORPL also achieves about 100% reliability, but spends more energy than RPL due to anycast-based transmissions without the phase-lock operation. Fig. 7(c) presents the number of hops for



(a) (b) (c)

Fig. 7. PRR, duty-cycle, and number of hops results from a static network: (a) PRR, (b) duty-cycle, and (c) hops.

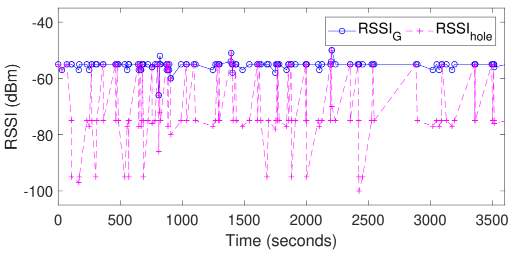
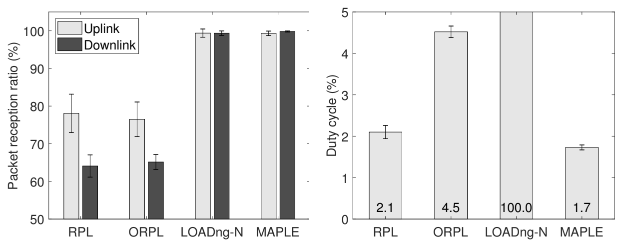


Fig. 8. RSSI adaptation of a hole node (node 27).



(a) (b)

Fig. 9. PRR and duty-cycle results for the mobile node: (a) PRR and (b) duty cycle.

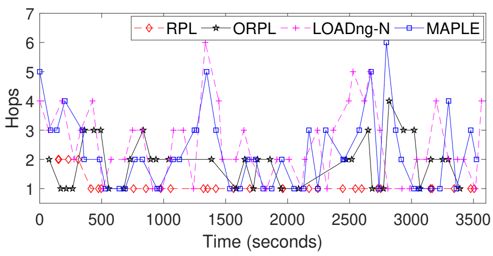


Fig. 10. Hops for uplink traffic from the moving node.

uplink and downlink traffic. In ORPL, uplink hop is shorter than that of downlink. As the root always listens to the medium without duty-cycling, it is likely to receive and acknowledge an uplink packet from a neighbor earlier than any other duty-cycling nodes. In case of downward traffic, on the other hand, it is common that the packets are relayed by other early wake-up nodes.

MAPLE has over 99% uplink PRR with reasonable energy consumption in the static network even though *MAPLE* was devised for mobile network. During the experiment, we observed local maximum RSSI problem in node 27 (see Fig. 5). It was deployed in a relatively open space. As a result, it receives beacons or downlink packets from the gateway with higher RSSI than its neighbors, which are located inside the rooms. Fig. 8 depicts how the node stuck in a hole adapts its RSSI. Whenever it detects itself in a local-maximum point, experiencing transmission failures, it escapes from the hole by adjusting its routing metric from $RSSI_G$ into $RSSI_{hole}$. With this approach, *MAPLE* tackles the local-maximum problem, guaranteeing high reliability. However, the hole node shows the highest energy consumption (i.e., 4.15% of duty cycle) due to transmission failures it experiences during the RSSI adaptation. Nevertheless, *MAPLE* shows lower energy consumption than ORPL since it benefits from its ATP architecture which enables single-hop downlink transmission.

Thanks to NACK-based local retransmission, *MAPLE*'s downlink PRR is also nearly perfect. We discovered that more than 3% of high-power downlink packets were lost during the experiments. In particular, the nodes which have low SNR from the gateway, such as nodes 26 and 27, went through about 20% of downlink packet loss. Nevertheless, by broadcasting NACK locally, they could receive the downlink packets successfully from neighbors, reaching 99.94% downlink PRR with average hop count of 1.035, as shown in the Fig. 7.

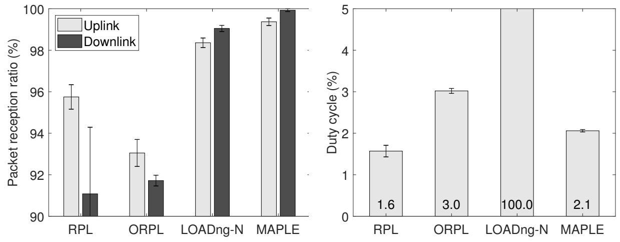
C. Network with Mobility

Having the experimental results from a static LLN as a basis, we now move on to our main evaluation with mobility.

C.1 One Mobile Node

In this experiment, we first introduce a single mobile node into the network to examine its performance under mobility. To keep the number of nodes consistent, we use node 15, which is in a corner of testbed topology, as a mobile node. While carrying the node, we walked back and forth along the path between points *A* and *B* shown in Fig. 5. In each room we enter, we stay 90 seconds while still walking, and then exit. With this movement, one-way trip time for the whole path is about 5 minutes.

Fig. 9(a) presents the PRR achieved for the uplink and downlink traffic from/to the mobile node. The performance of RPL and ORPL degrades severely. RPL did not react properly to link disconnections. Fig. 10 shows the number of hops that uplink traffic from the mobile node goes through. In RPL, the mobile node has the root as its preferred parent most of time. Even if it sometimes detects link failure with the root, the link becomes valid again by node's mobility. It makes the link quality remain



(a) (b)

Fig. 11. PRR and duty-cycle results from the network with three mobile nodes: (a) PRR and (b) duty cycle.

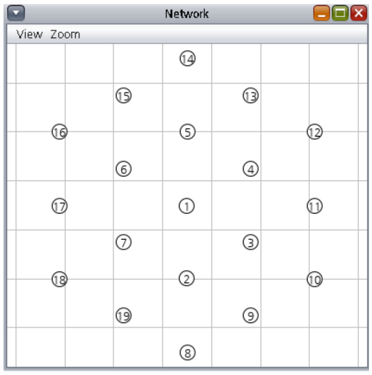
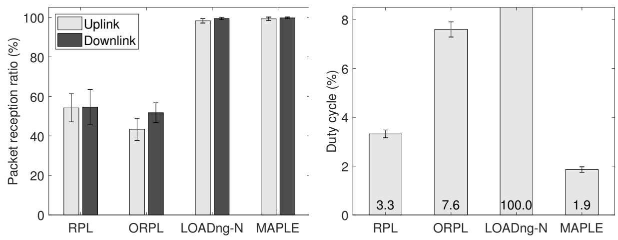


Fig. 12. Network topology for static nodes in Cooja simulation.



(a) (b)

Fig. 13. PRR and duty-cycle results from 50-node simulation: (a) PRR and (b) duty cycle.

good enough for the mobile node to keep the current preferred parent. In addition, *the effort after detecting packet losses* could not be a solution to provide seamless connectivity. The main problem in ORPL is that, to update link quality with neighbors, it relies on Trickle algorithm [44], thus not reacting topology dynamics promptly. In both RPL and ORPL, uplink PRR is better than downlink since the root is always-on. From the view of the always-on root, a train of transmission strobes, which were intended for duty-cycling receivers, have an effect of multiple retransmissions.

Fig. 10 also shows the mobile node with *MAPLE* or LOADng-N[3] adapts its uplink hops according to change of its location, achieving PRR above 99% in all cases as shown Fig. 9(a). Fig. 9(b) illustrates duty-cycle of the mobile node. *MAPLE* shows the highest energy efficiency whereas RPL and ORPL suffer from frequent packet losses and retransmission, incurring more energy consumption.

C.2 Three Mobile Nodes

Now we consider three mobile nodes with 29 static nodes. Among three mobile nodes, two nodes continuously moved back and forth at typical walking speed along the path shown in Fig. 5, but in opposite direction to each other. Their one-way trip time is 3 minutes. The other mobile node moved in the same manner as the previous experiment with a single mobile device. Overall, the performance of each mobile node was consistent with our previous results. Thus, in this subsection, we focus on how much the mobile nodes affect the performance of the whole network.

Fig. 11(a) plots the average PRR for all low-power nodes. For RPL and ORPL, not only the mobile nodes but also their descendants sequentially are affected by wrong routing decisions with outdated information. The PRR of LOADng-N remains still good under mobility. However, it sometimes fails to transmit a packet with more than 1% of loss rate for both downlink and uplink. As an intrinsic drawback of unicast transmission, it cannot benefit from multi-path diversity.

Meanwhile, there are three reasons why *MAPLE* shows the highest PRR for uplink traffic. Firstly, *MAPLE* is more robust to link failure with spatial diversity using opportunistic transmissions. Next, routing information throughout the network (i.e., an $RSSI_G$ gradient) is newly updated every beacon interval of 5 seconds. Given the network mobility of human walking speed, with this interval, it is enough for the $RSSI_G$ gradient to be tuned to topology dynamics. Lastly, even though the routing information is not perfect, (i.e., with a non-monotonic $RSSI_G$ gradient along a path) incurring some transmission failures, the node tries more and more paths by lowering its $RSSI_G$ gradually before the packet is dropped. Downlink PRR of *MAPLE* is more reliable, by using high-power transmission and introducing local NACK-based retransmission mechanism.

As presented in Fig. 11(b), *MAPLE*'s energy-efficiency is also good, which is mainly attributed to the effect of eliminating multihop downlink relay between low-power nodes. Additionally, with regard to routing overhead, while the compared protocols use broadcast packets which occupy the medium during a whole sleep interval and need to be forwarded hop-by-hop, the cost in *MAPLE* is negligible, a single timely transmission of the beacon.

D. Simulation Study under More Mobility

In our testbed experiments, we were unable to increase the number of mobile nodes to more than 3 due to limitations in human resources. For this reason, we instead used Contiki-based Cooja simulator to add more mobile nodes and expand our evaluation to showcase the performance of *MAPLE* under higher level of network mobility. As shown in Fig. 12, we firstly deployed 19 static nodes to guarantee connectivity between mobile nodes and the gateway (i.e., node 1). The distance between two adjacent nodes are 15 m. Then, for network mobility, we added another 31 nodes (total of 50 nodes) which independently move within the range of 45 m from the gateway. Each node follows Random way-point model [45] with the minimum and maxi-

[3]We exclude LOADng with duty-cycling from mobile experiments since we already identified its chaotic performance through the previous static experiment.

mum speeds of 0.5 m/s and 2.0 m/s, respectively. For *MAPLE*, the gateway uses a transmit power of 0 dBm. The other low-power nodes use -10 dBm, having transmission range of about 17 m. All the other experimental settings are identical to those of previous testbed experiments.

Figs. 13(a) and 13(b) plots the average PRR and duty-cycle of all low-power nodes, respectively. Compared to the previous experiment results, we found that RPL and ORPL are impacted severely by increased network mobility, showing less than 60% PRR with larger duty-cycle. Meanwhile, LOADng-N achieves PRR near 100% with the expense of 100% of duty-cycle. Most importantly, *MAPLE* outperforms the others in terms of both PRR and duty-cycle. Surprisingly, it shows lower energy consumption than RPL despite RPL works over ContikiMAC which includes the phase-lock operation to minimize energy consumption for unicasts. Overall, simulation results are in-line with the experiment results, and shows that *MAPLE* achieves significantly better reliability as well as energy efficiency under high mobility.

V. CONCLUSION

We presented *MAPLE*, an asymmetric transmit power-based routing architecture that supports mobility of resource-constrained devices in LLNs. Using high transmit power of the gateway, LLN nodes are synchronized for low duty-cycle operation, and RSSI gradient field based opportunistic routing is designed which eliminates the need for any neighbor or routing table. This enables scalability, low and constant memory footprint, and provides responsive routing metric without control overhead. We obtain reliable RSSI measurements via multi-sampling approach, and resolve the local maximum problem through adaptive and local adjustment of the routing metric. We implemented *MAPLE* on a low-power embedded platform, and evaluated through experiments on a real multihop LLN testbed consisting of 31 low-power ZigBee nodes and 1 high-power gateway. We showed that *MAPLE* improves the performance of mobile devices in a multi-hop LLN testbed by 27.2%/55.7% and 17.9% in terms of both uplink/downlink reliability and energy efficiency, respectively. As future work, we plan to improve *MAPLE* in terms of latency and energy-consumption, and evaluate *MAPLE* on large-scale testbed such as Indriya and IoT-LAB. We envision that our approach can be used in many practical indoor IoT applications where mobility is becoming an integral part of LLNs.

REFERENCES

[1] J. Park *et al.*, "Glasses for the third eye: Improving clinical data analysis with motion sensor-based filtering," in *Proc. ACM SenSys*, pp. 99–112, Nov. 2017.

[2] J. Ko *et al.*, "MEDiSN: Medical emergency detection in sensor networks," *ACM Trans. Embedded Comput. Syst.*, vol. 10, no. 1, p. 11, 2010.

[3] M. S. Shahamabadi, B. B. M. Ali, P. Varahram, and A. J. Jara, "A network mobility solution based on 6LoWPAN hospital wireless sensor network (NEMO-HWSN)," in *Proc. IMIS*, 2013.

[4] 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.

[5] H.-S. Kim, J. Ko, and S. Bahk, "Smarter markets for smarter life: Applications, challenges, and deployment experiences," *IEEE Commun. Mag.*, vol. 55, no. 5, pp. 34–41, 2017.

[6] S. Berman, V. Kumar, and R. Nagpal, "Design of control policies for spatially inhomogeneous robot swarms with application to commercial pollination," in *Proc. IEEE ICRA*, 2011.

[7] P. Tokekar, D. Bhadauria, A. Studenski, and V. Isler, "A robotic system for monitoring carp in Minnesota lakes," *J. Field Robotics*, vol. 27, no. 6, pp. 779–789, 2010.

[8] G. Montenegro, N. Kushalnagar, J. Hui, and D. Culler, "Transmission of IPv6 packets over IEEE 802.15.4 networks," *RFC 4944*, Sept. 2007.

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

[10] S. Duquennoy, O. Landsiedel, and T. Voigt, "Let the tree bloom: Scalable opportunistic routing with ORPL," in *Proc. ACM SenSys*, 2013.

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

[12] J. Ko and M. Chang, "Momoro: Providing mobility support for low-power wireless applications," *IEEE Syst. J.*, vol. 9, no. 2, pp. 585–594, 2015.

[13] 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 Tuts.*, vol. 19, pp. 2502–2525, Sept. 2017.

[14] A. Oliveira and T. Vazão, "Low-power and lossy networks under mobility: A survey," *Computer Networks*, vol. 107, pp. 339–352, 2016.

[15] H. Fotouhi, D. Moreira, and M. Alves, "mRPL: Boosting mobility in the Internet of Things," *Ad Hoc Networks*, vol. 26, pp. 17–35, 2015.

[16] T. Clausen *et al.*, "The lightweight on-demand ad hoc distance-vector routing protocol-next generation (LOADng)," *draft-clausen-lln-loadng-09*, 2013.

[17] T. Clausen, J. Yi, and U. Herberg, "Lightweight on-demand ad hoc distance-vector routing-next generation (LOADng): Protocol, extension, and applicability," *Computer Networks*, vol. 126, 2017.

[18] O. Gaddour, A. Koubâa, R. Rangarajan, O. Cheikhrouhou, E. Tovar, and M. Abid, "Co-RPL: RPL routing for mobile low power wireless sensor networks using Corona mechanism," in *Proc. IEEE SIES*, 2014.

[19] I. El Korbi, M. B. Brahim, C. Adjih, and L. A. Saidane, "Mobility enhanced rpl for wireless sensor networks," in *Proc. IEEE NOF*, 2012.

[20] H.-S. Kim, M.-S. Lee, Y.-J. Choi, J. Ko, and S. Bahk, "Reliable and energy-efficient downward packet delivery in asymmetric transmission power-based networks," *ACM Trans. Sensor Networks*, vol. 12, no. 4, 2016.

[21] Cisco, "Connected Grid Networks for Smart Grid - Field Area Network." [Online] Available: http://www.cisco.com/web/strategy/energy/field_area_network.html

[22] T. Group, "Thread Stack Fundamentals." [Online] Available: http://threadgroup.org/, July 2015

[23] M. W. Group, "Mesh Profile," *Bluetooth Specification*, July 2017.

[24] D. Lymberopoulos, Q. Lindsey, and A. Savvides, "An empirical characterization of radio signal strength variability in 3-D IEEE 802.15. 4 networks using monopole antennas," in *Proc. EWSN*, 2006.

[25] M. Lindhé, K. H. Johansson, and A. Bicchi, "An experimental study of exploiting multipath fading for robot communications," in *Proc. RSS*, June 2007.

[26] M. Malajner, K. Benkic, P. Planinsic, and Z. Cucej, "The accuracy of propagation models for distance measurement between WSN nodes," in *Proc. IEEE IWSSIP*, 2009.

[27] F. Hermans, O. Rensfelt, T. Voigt, E. Ngai, L.-Å. Norden, and P. Gunningberg, "SoNIC: classifying interference in 802.15. 4 sensor networks," in *Proc. ACM/IEEE IPSN*, 2013.

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

[29] F. Cadger, K. Curran, J. Santos, and S. Moffett, "A survey of geographical routing in wireless ad-hoc networks," *IEEE Commun. Surveys Tuts.*, vol. 15, no. 2, pp. 621–653, 2013.

[30] C. Cobarzan, J. Montavont, and T. Noel, "Analysis and performance evaluation of rpl under mobility," in *Proc. IEEE ISCC*, 2014.

[31] C. Perkins, E. Belding-Royer, and S. Das, "Ad hoc on-demand distance vector (aodv) routing," *RFC 3561*, 2003.

[32] S. Biswas and R. Morris, "Exor: Opportunistic multi-hop routing for wireless networks," *ACM SIGCOMM Computer Communication Review*, vol. 35, no. 4, pp. 133–144, 2005.

[33] O. Landsiedel, E. Ghadimi, S. Duquennoy, and M. Johansson, "Low power, low delay: Opportunistic routing meets duty cycling," in *Proc. ACM IPSN*, 2012.

[34] G. Huo and X. Wang, "An opportunistic routing for mobile wireless sensor networks based on RSSI," in *Proc. IEEE WiCOM*, 2008.

[35] F. Ferrari, M. Zimmerling, L. Thiele, and O. Saukh, "Efficient network flooding and time synchronization with glossy," in *Proc. IEEE IPSN*, 2011.

[36] S. Duquennoy, A. Elsts, B. Nahas, and G. Oikonomou, "TSCH and 6TiSCH for Contiki: Challenges, design and evaluation," in *Proc. IEEE DCOSS*, 2017.

[37] A. Dunkels, "The contikimac radio duty cycling protocol," *SICS Technical Report*, 2011.

[38] X. Ji, Y. He, J. Wang, W. Dong, X. Wu, and Y. Liu, "Walking down the stairs: Efficient collision resolution for wireless sensor networks," in *Proc. IEEE INFOCOM*, 2014.

[39] S. Jeong, H.-S. Kim, S.-G. Yoon, and S. Bahk, "Q-BT: Queue-based burst transmission over an asynchronous duty-cycle MAC protocol," *IEEE Commun. Lett.*, vol. 20, no. 4, pp. 812–815, 2016.

[40] V. Shah and S. Krishnamurthy, "Handling asymmetry in power heterogeneous ad hoc networks: A cross layer approach," in *Proc. IEEE ICDCS*, 2005.

[41] X. Du, D. Wu, W. Liu, and Y. Fang, "Multiclass routing and medium access control for heterogeneous mobile ad hoc networks," *IEEE Trans. Veh. Tech.*, vol. 55, no. 1, pp. 270–277, 2006.

[42] T. Instruments, "CC2420: 2.4 GHz IEEE 802.15. 4/ZigBee-ready RF Transceiver," [Online] Available: http://www.ti.com/lit/gpn/cc2420, vol. 53, 2006

[43] A. Dunkels, B. Gronvall, and T. Voigt, "Contiki-a lightweight and flexible operating system for tiny networked sensors," in *Proc. IEEE LCN*, 2004.

[44] P. Levis and T. H. Clausen, "The trickle algorithm," *RFC 6206*, 2011.

[45] D. B. Johnson and D. A. Maltz, "Dynamic source routing in ad hoc wireless networks," *Mobile computing*, pp. 153–181, Springer, 1996.

Seungbeom Jeong is currently a Ph.D. student at the School of Electrical Engineering and Computer Science, Seoul National University (SNU), Seoul, Republic of Korea. He also received his B.S. degree from SNU in 2013. His research interests include network and communication protocols for mobile IoT system and low-power wireless networks.

Eunjeong Park is currently working toward the M.S. degree at the School of Electrical Engineering and Computer Science, Seoul National University (SNU), Seoul, Republic of Korea. She also received her B.S. degree from SNU in 2016. Her research interests include Bluetooth protocol design and wireless IoT system.

Dongyeon Woo received his B.S. degree in Electrical Engineering from Korea Advanced Institute of Science and Technology (KAIST) in 2016, and his M.S. degree at the School of Electrical Engineering and Computer Science, Seoul National University (SNU) in 2018. His research interests include network traffic engineering, wireless communication protocol and IoT systems.

Hyung-Sin Kim received his B.S. degree in Electrical Engineering from Seoul National University (SNU), Korea, in 2009. He also received his M.S. degree in 2011 and his Ph.D. degree in 2016, both in Electrical Engineering and Computer Science, from SNU. From March 2016 to August 2016, he was a postdoctoral researcher at the Network Laboratory (NETLAB) at SNU led by Dr. Saewoong Bahk. He is currently a postdoctoral researcher of the Software-Defined Building (SDB) group at the University of California, Berkeley led by Dr. David E. Culler. His research interests include development of IoT systems for urban marketplaces, smart grid and SDB, and network and communication protocols for low-power wireless systems.

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 where he was a member of the Networked Systems Laboratory (NSL) led by Dr. Ramesh Govindan. 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 CG-Mesh system and WPAN software. In 2014, he was with the Hongik University, Department of Computer Information Communication as an assistant professor. Jeongyeup Paek is currently an assistant professor at Chung-Ang University, School of Computer Science and Engineering, Seoul, Republic of Korea since 2015. He is an IEEE and ACM 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 Samsung Electronics, LG Electronics, SK Telecom, etc and published more than 200 technical papers and holds 72 patents. He is 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 was vice president 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.