

Energy-Efficient Overlay Protocol for BLE Beacon-based Mesh Network

Pai Chet Ng, *Student Member, IEEE*, James She, *Member, IEEE*
and Petros Spachos, *Senior Member, IEEE*,

Abstract—Bluetooth Low Energy (BLE) beacons are designed to operate for years on a coin-cell battery. However, the formation of a mesh network, overlaying on the existing Bluetooth Low Energy (BLE) beacons infrastructure, can severely degrade the lifetime of underlying beacons owing to the excessive current drawn by the scanning event. Even though we can sustain the lifetime of the underlying beacon with duty-cycle scanning, such duty-cycle scanning imposes another challenge to the overlay mesh in disseminating the packet. To this end, this paper proposes a novel overlay protocol that: 1) employs duty-cycle scanning to guarantee the lifetime of the underlying beacon, while 2) defining a set of scanning policies to increase the packet dissemination rate through the overlay mesh network. The duty-cycle scanning defines the scanning time slot based on the lowest feasible duty cycle unveiled through a comprehensive analysis of energy consumed by advertising and scanning events. The scanning policies, on the other hand, allow each node to explore all possible time slots before locking their scanning event to a particular time slot that is most likely to hear the incoming packet. Extensive experiments with practical implementation demonstrate the feasibility of our proposed overlay mesh for real-world use cases.

Index Terms—BLE, Beacon Networks, Mesh Networks, underlying Beacon, Overlay Mesh, Duty Cycle

1 INTRODUCTION

IN 2018, twelve boys were reported missing inside a cave around the Chiang Rai province, Thailand [1]. The rescue team took approximately 10 days to locate the missing boys. The rescue mission would be much easier if these boys could use their smartphones to connect to the Internet and report their approximate location. Despite the wide accessibility of the Internet nowadays, it is relatively hard, if not impossible, to access the Internet inside a cave. In this paper, we define this type of location, where it is hard to access the Internet, as a harsh environment. It is of crucial importance if we can remotely monitor the proximity of humans inside these harsh environments. Remote monitoring in this context refers to the ability to locate a person anywhere on the earth when the person under monitored is located in a harsh environment that does not have the Internet. Nowadays, with the Internet and GPS technology, it is very easy to report one's location and we can even remotely monitor the location of our children from the comfort of our home. However, if our children went to the cave and lost the Internet connection, it would be a challenge for us to remotely monitor their location. While it might be quite difficult to identify the exact location, obtaining proximity information of a human trapped inside a harsh environment is still possible. This proximity information can provide useful first-hand information to the rescue team when someone is trapped inside, and thus shorten the rescuing time.

Even though Bluetooth Low Energy (BLE) beacons can be deployed in these harsh environments for proximity

sensing, the sensing information is only available locally on the smartphone located near the beacon. By measuring the received signal strength (RSS) from all the deployed beacons, the smartphone can estimate to which beacon the smartphone is in proximity with [2]. The capability of beacons for proximity estimation has led to massive beacon deployment in many public areas, such as museums [3] and art galleries [4]. Many smartphone-based applications leverage this proximity information to provide proximity-based services, such as notifying the user about a particular artifact or enabling on-the-spot interaction with the artifact. These notification and interaction services are delivered through the Internet connectivity available on the smartphone. If accessing the Internet is an issue in harsh environments, it is almost impossible to monitor the user's proximity solely based on the current beacon technology.

Motivated by the Internet accessibility issue in harsh environments, we propose a novel mesh network overlay on BLE beacons. Our novel BLE-beacon-based network enables each beacon to forward the packet containing the sensing information to a location that can access the Internet. This is achieved by performing a firmware update on the existing beacon without modifying the underlying infrastructure, as described in our previous work [5]. Originally, a beacon is designed to broadcast an advertising packet periodically targeting proximity-based applications like the museum and the gallery applications we discussed previously. With our firmware update, each beacon can now disseminate the packet to all the neighboring beacons, enabling the applications like emergency alert, remote control, remote monitoring, etc. Rather than pushing the notification to the smartphone, every beacon can help to forward the notification packet to all the neighboring beacons when there is an emergency; or, relaying the sensing information

Pai Chet Ng and James She are with the Department of Electronics and Computer Engineering, Hong Kong University of Science and Technology, Hong Kong. E-mail: {pcng, eejames}@ust.hk
Petros Spachos is with the School of Engineering, University of Guelph, Canada. E-mail: petros@uoguelph.ca

from an underground workstation to a remote workstation. However, the lifetime of the underlying beacon is severely degraded due to the formation of the overlay mesh. Such a lifetime degradation has greatly discounted the potential of a low-power beacon for IoT development.

This paper takes a step forward to model the lifetime of the underlying beacon by analyzing the energy consumed by: 1) the advertising event, and 2) the scanning event. In light of the above analysis, a duty-cycle scanning approach is employed to define the scanning event. Even though there are works that use duty-cycle to achieve low power wireless sensor network [6] [7] [8], these works assume that the device sleeps most of the time and only wakes up when there is a packet to transmit or to receive. However, the beacon by itself is already a duty-cycle device that wakes up periodically to trigger the advertising event while spending most of its time sleeping [9]. Our duty-cycle scanning is more than just defining the active and sleep duration, rather it needs to explore the available sleep duration in between advertising events to perform the scanning. That being said, duty-cycle scanning trade-offs the packet forwarding activity of the overlay mesh to sustain the lifetime of the underlying beacon.

To this end, we propose an overlay protocol 1) to guarantee the lifetime of the underlying beacon, and 2) to improve the packet forwarding activity by the overlay mesh network. In particular, we use the lowest feasible duty cycle \bar{D} unveiled from the comprehensive energy analysis to define the scanning time slot T_w , while defining a set of scanning policies allowing each beacon to explore all possible time slots before locking their scanning event to a particular time slot. To the best of our knowledge, this is the very first work that achieves an energy-efficient overlay protocol considering both the underlying beacon infrastructure and overlay mesh network. The main contributions of this paper are summarized as follows:

- a comprehensive energy analysis is conducted to study energy consumption by advertising and scanning. Such an analysis unveils the lowest feasible duty cycle to receive the incoming packet.
- the lifetime of the underlying beacon is modeled based on the above analysis. The upper bound and lower bound of the lifetime are derived disclosing the effect of advertising, scanning, and meshing on the beacon lifetime.
- our scanning policies impose each node to explore all possible time slots before locking their scanning event to a particular time slot, increasing the likelihood of hearing the packets transmitted by neighboring beacons.
- the performance of packet forwarding activity and the feasibility of overlay mesh network for real-world problems is demonstrated with a real network testbed.

Following the introduction, Section 2 reviews the current work that exploits either BLE beacon or mesh networks. Section 3 succinctly describes the BLE technology and then defines the 3 events related our overlay mesh. Section 4 presents the comprehensive analysis of energy consumption by the 3 events. Section 5 models the lifetime of the underly-

ing beacon. Section 6 discusses the packet forwarding problem with our overlay protocol. Section 7 demonstrates the feasibility of our overlay mesh with real network testbed. Section 8 concludes the paper.

2 RELATED WORK

This section reviews the BLE network technologies, focusing on beacon, ad-hoc, and mesh networks.

2.1 BLE Beacon Network

Each node in the beacon networks broadcasts their advertising packet periodically according to the pre-defined advertising interval [10]. By measuring the RSS from the beacon [11], we can infer the proximity between the beacon and the receiver. For example, [12] leverages the RSS to detect the proximity between the beacon and the smartphone. Besides detecting the proximity between devices, we can also manipulate the RSS measurements from the beacon to detect the proximity between humans in the workplace [13]. All the above works are intended for short-range applications based on proximity detection.

BLE beacon has attracted a lot of interest since its introduction in 2013, leading to massive deployment in public locations. Accordingly, many research works have been conducted to study the problems related to large-scale beacon networks. Specifically, [14] studies the beacons network with different topologies and claimed that the minimum spacing between each beacon should be at least $0.5m$ to maximize the packet delivery; whereas [2] presents an adaptive scanning method to detect the beacons even they are closely spaced, i.e., the distance between each beacon is less than $0.5m$. Some applications have been developed utilize these dense beacon network for item identification [15], occupancy detection [16] and smart interaction [17]. Many works have exploited this massively deployed beacon network for user positioning. For example, [18] constructs the location fingerprints by measuring the RSS contributed by a network of 19 beacons; whereas [19] provides the theoretical bound regarding the number of beacons required to achieve an unambiguous user positioning. However, none of these works leverage the massive number of beacons to forward the data. Rather, each beacon just keeps broadcasting its own packet periodically regardless if there was any receiver in their vicinity.

2.2 BLE Ad-hoc Network

In the past decade, Bluetooth has been widely studied for its feasibility in supporting ad-hoc networks besides being used as a replacement to wired infrastructure [20]. The characteristics of Bluetooth allowing device discovery and connection establishment at any time are the key elements in realizing the ad-hoc network [21]. However, the connection establishment is only possible when there is at least one master device and slave device located within the transmission range of each other. The device discovery could be a painful process before handshaking can be initiated to establish the connection. Hence, the potential of Bluetooth-based ad-hoc network is greatly discounted owing to the tedious processes involving device discovery and connection establishment.

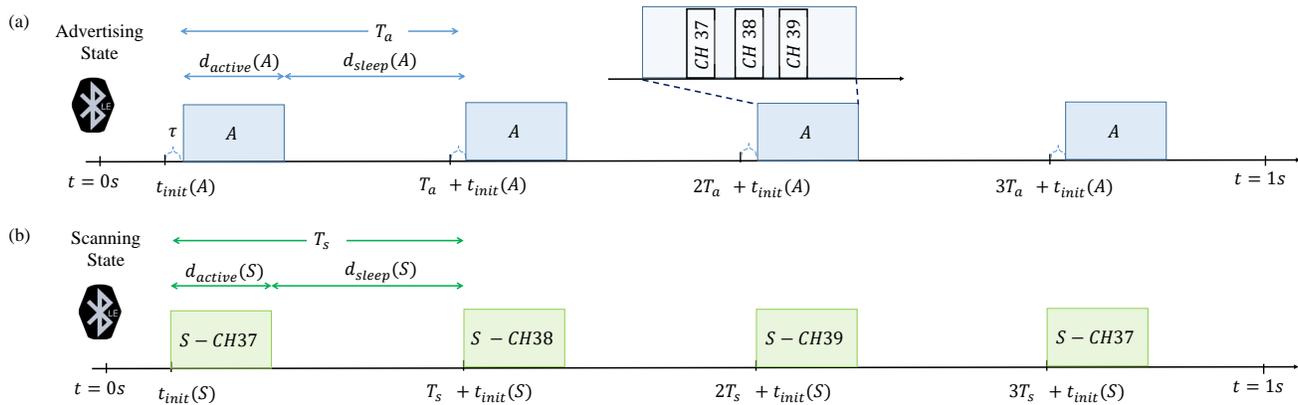


Fig. 1: (a) A BLE device in the advertising state triggers a sequence of advertising events. The advertising event transmits the advertising packet on the three advertising channels (i.e., CH 37, 38, and 39) sequentially during each advertising interval T_a . (b) A BLE device in the scanning state triggers the scanning event to listen to the incoming packet. The scanning event only scans on one of the advertising channels during each scanning interval T_s .

Recently, BLE has been introduced as a low-power alternative to classic Bluetooth. Besides being a low-power radio technology, BLE also reduces the latency in device discovery and thus allows more spontaneous ad-hoc networks, especially the network involving mobile devices [22]. Besides using the master/slave connection protocol, BLE allows encountered devices to exchange data in a connectionless approach. In other words, no connection establishment is required to transmit the data from one device to another device. The common example is to use a beacon to work as a periodic broadcaster, and any receiver can receive the data broadcast by the beacon when it comes into the broadcasting range [23]. The problem with the above approach is that the data transmission can only be done in a single hop. While the connectionless approach allows quick establishment for ad-hoc networks, such a single-hop communication is limited to applications like the smart museum that relies on unidirectional device-to-device communication. However, applications like emergency notification and remote monitoring always require either bi-directional or multi-hop communication to disseminate the data to all the neighboring devices, as well as devices located in a remote location.

2.3 BLE Mesh Network

Bluetooth Special Interest Group (SIG) announced their first official mesh model in July 2017 [24]. This mesh model is based on a newly defined mesh stack, which might induce a steep learning curve for those who wish to switch from their current beacon network to the mesh network. Detailed descriptions about the mesh stack and its new terminologies can be found in [25]. In fact, a number of BLE mesh networks have been proposed prior to the launching of the official mesh model. These mesh networks can be broadly classified into 2 categories: connectionless and connection-oriented.

A connectionless mesh network, such as BLEmesh [26], exploits the flooding mechanism to disseminate its packet to all the nodes. The work in [27] also exploits connectionless BLE network to broadcast and forward the packet regarding the RF fingerprint information so that to improve the

application that requires these dynamic RF fingerprints. On the other hand, a connection-oriented mesh network leverages the connection information to route the packet. For example, BLE Mesh Network (BMN) [28] employs the Directed Acyclic Graph (DAG) structure in constructing the routing table. The DAG structure is similar to a tree-based network in which each node is required to store a routing table containing the information regarding their parents and the list of children. Meanwhile, [29] eliminates the need to store a routing table with an on-demand routing approach. Bluetooth Now [30] is another mesh network that can switch between the connectionless and connection-oriented approaches subject to the priority of the packet. So far, it is hard to integrate these BLE mesh networks directly with the existing BLE beacon network. The easiest way is to deploy new mesh nodes in between beacon nodes to help to forward the packet broadcast by the neighboring beacons. However, this is a tedious and ineffective approach considering the massive number of already deployed beacons. To this end, this paper proposes a novel overlay mesh, enabling existing beacons with mesh functionality while retaining their existing advertising events.

3 OVERVIEW OF BLUETOOTH LOW ENERGY AND BLE-BASED OVERLAY MESH (OUR OVERLAY MESH)

This section provides a succinct overview of advertising and scanning events with BLE technology before recapping the overlay mesh proposed by our previous work [5].

3.1 Advertising and Scanning

BLE divides the 2.4 GHz ISM band into 40 channels, with three advertising channels (i.e., CH 37, 38, and 39) and 37 data channels. According to BLE specifications [31] [32], a BLE device can be categorized into five different states, i.e., standby, advertising, scanning, initiating, and connected states. When the BLE device is in the advertising or scanning state, it only uses the 3 advertising channels to broadcast its packet or listen to the incoming packet. The BLE device will only enter into initiating and connected states when there

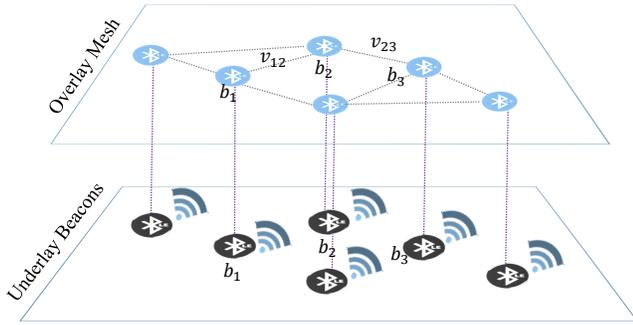


Fig. 2: Our proposed overlay mesh network enables existing beacons to participate in the packet forwarding activity without modifying the underlying infrastructure.

is a connection establishment request. Once a connection is established, it will use the 37 data channels for secure data exchange.

In this paper, we focus on the BLE device that mainly operates in the advertising state. This type of device is commonly known as a BLE beacon. When in the advertising state, the beacon executes a sequence of advertising events periodically according to the advertising interval T_a . During each T_a , the advertising packet is transmitted on all the 3 advertising channels sequentially. The advertising packet can be picked up by any BLE receiver, such as Bluetooth-compatible smartphones. A BLE receiver is a device in the scanning state, executing sequential scanning events for a duration defined by the scanning window T_w . The receiver will scan on the first advertising channel and then repeat the same scanning on the next advertising channel at the next scanning interval T_s . Fig. 1 illustrates the BLE device in (a) the advertising and (b) the scanning states. As discussed, the advertising and scanning events are subject to the following parameters: advertising interval T_a , and scanning interval T_s , and scanning window T_w .

3.1.1 Advertising Interval T_a

Advertising interval defines how frequently the advertising event is triggered to transmit the advertising packet. In practice, the link layer of BLE imposes a pseudo-random delay τ in between advertising events to resolve any possible collision. The advertising interval should be refined to $T'_a = T_a + \tau$, where $\tau = [0\text{ ms}, 10\text{ ms}]$. For the simplicity of exposition, we did not take τ into consideration for the rest of this paper. Moreover, τ is generally negligible when T_a is large. The BLE device will go into the sleep mode in between T_a to minimize the energy consumption. As illustrated in Fig. 1(a), the device only wakes up to broadcast the packet, while spending most of its time in the sleep mode. Let d_{active} be the duration of the active mode and d_{sleep} the duration of the sleep mode, we can describe the advertising interval as $T_a = d_{active}(A) + d_{sleep}(A)$, where A denotes the advertising event.

3.1.2 Scanning Interval T_s and Scanning Window T_w

The scanning interval defines how frequently the scanning event is triggered, whereas the scanning window defines the duration of the scanning event. The scanning event is a sequential event as illustrated in Fig. 1(b): if the device

scanned CH37 at $t = T_s$, it will then scan CH38 at $t = 2T_s$, CH39 at $t = 3T_s$, and so on. In general, the device will spend most of its time sleeping if $T_w \ll T_s$; whereas it will be active at all times if $T_s = T_w$. Let S be the scanning event, we have $T_s = d_{active}(S) + d_{sleep}(S)$ and $T_w = d_{active}(S)$, where T_s and T_w range from 0s to 10.24s. The effect of T_a , T_s , and T_w on energy consumption is discussed in Section 4.

3.2 BLE-based Overlay Mesh

Recently, Bluetooth 5 introduces the mesh functionality into the Bluetooth protocol stack to extend the reachability of BLE devices [32]. While it is possible to deploy the newly defined Bluetooth mesh node in between a cluster of beacon nodes to gather and forward the packet, such an approach is cost-ineffective considering the high deployment cost and laboring force. Motivated by the above limitation, our previous work [5] presents a novel overlay mesh that creates an overlay mesh network on top of the underlying beacon infrastructure to facilitate the packet forwarding request, as shown in Fig. 2.

Let $G_u = (B, R, L)$ represent the underlying beacon network where set $B = \{b_1, b_2, \dots, b_u, \dots, b_n\}$ defines a list of beacons and set $R = \{r_1, r_2, \dots, r_v, \dots, r_n\}$ a list of receivers (e.g., smartphone), and set $L = \{l_{b_u r_v} | \delta(b_u, r_v) \leq \gamma_{b_u}, \forall b_u \in B, \forall r_v \in R\}$ describes the relationship between the beacon and the receiver, where γ_{b_u} denotes the maximum transmission radius of node b_u . The underlying beacon network allows the packet to be transmitted from any beacon to the receiver, but there is no way to exchange the packet between the beacon. Our proposed overlay mesh, on the other hand, allows each beacon to receive the packet from other beacons through the virtual link.

Definition 1. (BLE-based Overlay Mesh) An existing beacon network can be transformed to an overlay mesh when there exists a set of virtual links between any pair of nodes located within the same transmission range to each other. Hence, our overlay mesh refines the existing beacon network G_u to $G_o = (B, R, L, V)$, where V is a set of virtual links,

$$V = \{v_{b_u b_v} | \delta(b_u, b_v) \leq \gamma_{b_u}, \forall b_u \neq b_v \in B\}. \quad (1)$$

With our overlay mesh, there always exists a virtual link between a node b_u to any other nodes located inside the transmission range of node b_u . Accordingly, a set of neighboring nodes to node b_u can be described as follows:

$$B_n(b_u) = \{b_v | \delta(b_u, b_v) \leq \gamma_{b_u}, \forall b_v \in B\}, \quad (2)$$

where $|B_n(b_u)| \leq |B|$ denotes the total number of neighboring nodes.

Intuitively, any node $b_v \in B_n(b_u)$ can help to forward the packet upon receiving the packet forwarding request from node b_u . The packet forwarding is achieved by encapsulating the packet to be forwarded into the existing advertising event defined by the original beacon network. More specifically, the beacon encapsulates additional packet information (e.g., sensor data) into its own advertising packet payload and sets the packet forwarding request flag to high. The packet forwarding request flag is located in the packet header, thus each receiving node can distinguish the type of the packet by inspecting the header. While any node

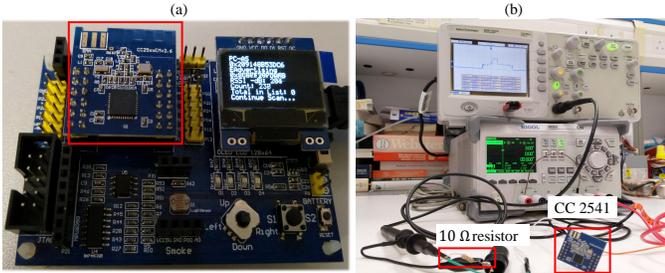


Fig. 3: (a) A Smart RF evaluation board is used to verify and debug the program before proceeding with energy measurement. (b) The measurement setup.

can participate in the packet forwarding activity until every node in the network receives the packet, it might possibly jeopardize the network traffic with redundant transmissions. Many different approaches can be employed to facilitate this intermittent packet forwarding request such that to minimize the packet delivery latency [33] [7] [34]; whereas our previous work proposes RSS-based bounded flooding (RBF) to minimize the redundancy. Further discussion of packet forwarding activity through the overlay mesh is described in Section 6.

4 ANALYSIS OF ENERGY CONSUMPTION BY ADVERTISING AND SCANNING

Even though several works have studied the energy performance of BLE devices, most of them either focus on the advertising event [35] or the neighbor discovery problem between two devices [36]. No works, so far, provide a comprehensive analysis of energy consumption considering both advertising and scanning events. This paper comprehensively analyzes energy consumed by these three events, which paves the way for lifetime modeling, presented in Section 5.

4.1 Measurement Setup

We set up the measurement following the procedures suggested by [37]. We used a smart RF evaluation board provided by Texas Instrument to debug and verify the program flashed into the CC2541 chipset, as shown in Fig. 3(a), before proceeding with the energy measurement. For the advertising event, we examined different types of PDU; whereas for the scanning event, we examined the impact of continuous scanning in comparison to duty-cycle scanning. Energy consumption by the packet forwarding activity can be estimated by averaging the energy consumed by both advertising and scanning events within a fixed interval. Table 1 describes each parameter configuration. As shown in Fig. 3(b), we used an oscilloscope (i.e., DSO1022A from Agilent Technologies) to capture the voltage waveform triggered by advertising and scanning events. All the captured waveforms were saved into comma-separated values (.csv) file format and exported to Matlab for further analysis.

4.2 Energy Consumption of Advertising Events

Advertising is a periodic event that repeats itself at every multiple of T_a . The device spends most of the time in sleep-

TABLE 1: Measurement Settings for Advertising event

Parameters	Settings
T_a	100 ms
P_T	-23 dBm
V_s	3V
R	10Ω
PDU	Scannable

ing mode and only wakes up to transmit the advertising packet for a very short period. The average duration for the device to stay in active mode is about $d_{active}(A) = 4.280$ ms. Accordingly, the duration of sleep mode with $T_a = 100$ ms can be calculated, i.e., $d_{sleep}(A) = T_a - d_{active}(A) = 100 - 4.280 = 95.72$ ms. Such a long sleeping duration is the key to the low power feature of the BLE beacon.

As described in [37], the active duration of an advertising event is defined by nine different phases, as illustrated in Fig. 4. Fig. 4(a) illustrates these nine major phases (i.e., MCU wake up, MCU pre-processing, radio preparation, advertising, transition, scanning, switching, MCU post-processing, and sleep preparation) with the scannable PDU; whereas Fig. 4(b) illustrates the non-connectable PDU. The energy consumption can be computed by considering the current drawn by these nine major phases. Suppose that the supply voltage is V_s , the energy consumed by the advertising event is

$$E_{active}(A) = V_s \sum_{p \in P} (I_p d_p) \quad (3)$$

where I_p and d_p denote the current drawn and the duration by each phase $p \in P$, respectively, and P is a set of phases described above.

On the other hand, energy consumption during the sleep mode can be computed by subtracting T_a with $d_{active} = \sum_{p \in P} d_p$,

$$E_{sleep}(A) = V_s I_{sleep} d_{sleep}(A) = V_s I_{sleep} \left(T_a - \sum_{p \in P} d_p \right) \quad (4)$$

Accordingly, the total energy consumption within a fixed advertising interval T_a can be obtained by summing up $E_{active}(A)$ and $E_{sleep}(A)$:

$$E_{adv} = V_s \sum_{p \in P} (I_p d_p) + V_s I_{sleep} \left(T_a - \sum_{p \in P} d_p \right) \quad (5)$$

Table 2 summarizes the energy consumption of the advertising event configured based on scannable and non-connectable PDUs. Note that scannable PDU has a longer active duration compared to non-connectable PDU. Scannable PDU is beneficial when there is a request for a secure connection involving handshaking. The non-connectable PDU eliminates the transition and scanning phases (Note that this scanning phase is mainly responsible for the handshaking process, which is different from the normal scanning event). The measurement shows that the non-connectable PDU reduces the energy consumption from $154.367 \mu J$ to $123.049 \mu J$. Since our overlay mesh does not require the handshaking process, non-connectable PDU is thus preferable since it can reduce unnecessary energy consumption.

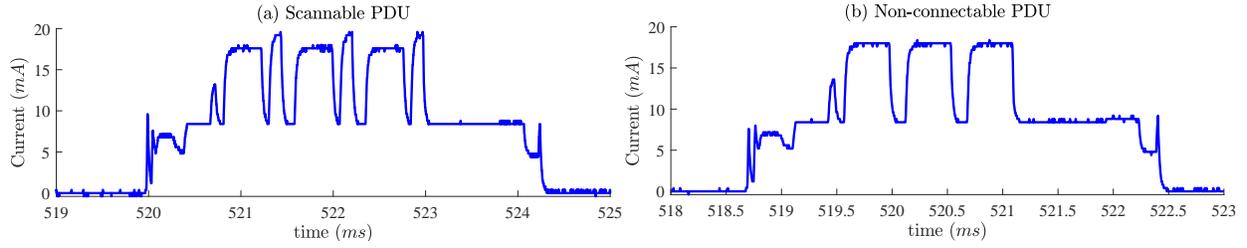


Fig. 4: Comparison between the advertising event with (a) the scannable PDU, and (b) the non-connectable PDU.

TABLE 2: The energy consumption of both PDUs.

PDU	$d_{active} (ms)$	$d_{sleep} (ms)$	$E_{adv} (\mu J)$
Scannable	4.28	95.72	154.367

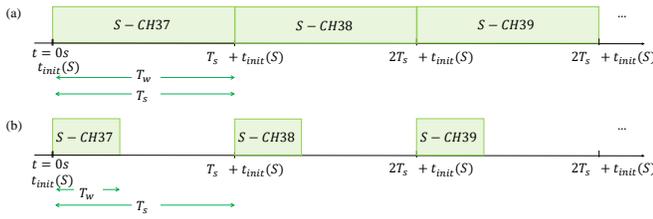


Fig. 5: A device is said to conduct (a) a continuous scanning when $T_w = T_s$ or (b) a duty-cycle scanning when $T_w < T_s$.

4.3 Energy Consumption of Scanning Events

According to Fig. 5(a), the device is expected to scan continuously if $T_w = T_s$; whereas Fig. 5(b) shows that we can configure a duty-cycle scanning by setting $T_w < T_s$. The current drawn by the scanning event is approximately $20mA$, which complies with the measurement listed in the CC2541 datasheet.

Energy consumption by the scanning event can be computed directly by multiplying the scan current with the scan window, i.e., $E_{scan} = V_s I_{scan} T_w$. Let $\mathcal{D} = \frac{T_w}{T_s}$ denote the scanning duty cycle, the energy consumed by a scanning event can be refined to the following:

$$E_{scan} = E_{active}(S) + E_{sleep}(S) = V_s T_s (I_{scan} \mathcal{D} + I_{sleep} (1 - \mathcal{D})) \quad (9)$$

The above equation unveils that energy consumption is dependent on the duty cycle \mathcal{D} . In particular, when $\mathcal{D} = 1$, we have $E_{dscan} = V_s T_s I_{scan}$, which means the scanning is continuous over the entire T_s ; when $\mathcal{D} = 0$, the device will only draw the sleep current I_{sleep} .

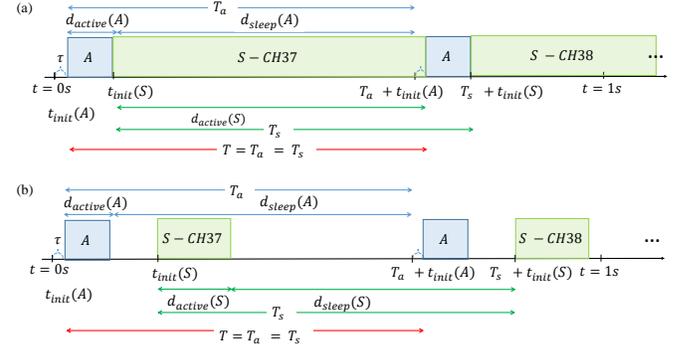


Fig. 6: The BLE device can trigger the scanning event in between the advertising events with either: (a) continuous scanning, or (b) duty-cycle scanning.

4.4 Energy Consumption of Overlay Mesh Node

Our proposed overlay mesh allows the node to retain its advertising event while handling the packet forwarding request. This is achieved by encapsulating the additional information into the advertising packet. To simplify the discussion, we assume a fixed time interval for both scanning and advertising intervals, i.e., $T = T_a = T_s$, as illustrated in Fig. 6. The above simplification is valid because the likelihood for the node to receive the advertising packet is relatively low when $T_s > T_a$.

In contrast to the pure scanning event, we need to take the active duration of the advertising event into account when defining the scanning duty cycle. Given the refined scanning duty cycle, i.e., $\mathcal{D} = \frac{T_w}{T - \sum_{p \in P} d_p}$, energy consumption by the overlay mesh node M can be computed as follows as in Eq. (11).

When $\mathcal{D} = 0$, the above equation reverts back to Eq. (5). Intuitively, the energy consumption is dominated by the advertising event when there is no scanning. When $\mathcal{D} = 1$, we have $E_{mesh} = V_s \sum_{p \in P} (I_p d_p) + V_s I_{scan} (T - \sum_{p \in P} d_p)$, which implies that the energy consumption is subject to the

$$E_{mesh} = E_{active}(M) + E_{sleep}(M) = E_{active}(A) + E_{active}(S) + (E_{sleep}(M)) = V_s \sum_{p \in P} (I_p d_p) + V_s I_{scan} T_w + V_s I_{sleep} \left(T - T_w - \sum_{p \in P} d_p \right) = V_s \sum_{p \in P} (I_p d_p) + V_s \left(T - \sum_{p \in P} d_p \right) (I_{scan} \mathcal{D} - I_{sleep} (1 - \mathcal{D})) \quad (11)$$

duration of advertising and scanning events. Both $\mathcal{D} = 0$ and $\mathcal{D} = 1$ are undesirable for our overlay mesh because it is impossible for the node to proceed with meshing if there is no scanning; whereas the lifetime of the underlying beacon will be greatly affected if we make the device to scan continuously. Clearly, the duty cycle is the dominant factor affecting the lifetime of the underlying beacon.

5 LIFETIME MODEL OF THE UNDERLYING BEACON

As discussed, the formation of the overlay mesh increases the energy consumption, affecting the lifetime of the underlying beacon. Suppose that the underlying beacon is powered by a battery with capacity equals to C , we can model the lifetime, in the unit of hours, with the following equation:

$$\tau = C \frac{V_s T}{E}, \quad (10)$$

where T and E are the fixed time interval and energy consumption, respectively. Take the advertising event, for example, T would be T_a and E would be E_{adv} (cf. Eq. (5)).

Continue with our previous example where energy consumed by the advertising event is $E_{adv} = 154.367 \mu J$. Suppose that a CR2450 coin-cell battery ($C = 620mAH$) is used, then the lifetime of the beacon can be computed as follows:

$$\tau = 620mAH \frac{3 \times 100ms}{154.367 \mu J} = 1204.92H \approx 50days.$$

If $T_a = 1s$, which is the common setting for most beacons on the market, then the beacon can last for 493.42 days, approximately one year and four months.

5.1 Continuous Scanning

The previous example only discussed the lifetime of the beacon subject to advertising events, without disclosing the effect of the overlay mesh on the underlying beacon. Since the overlay mesh requires the node to perform scanning in between advertising events to listen for the incoming packet, the lifetime of the underlying beacon is also subject to the scanning duration. For the worst-case scenario, where the beacon is configured to scan continuously, the current drawn by the scanning event would be the dominant factor determining the lifetime of the underlying beacon.

Theorem 1. (Lower Bound of the Lifetime) *The lifetime of an underlying beacon is lower bounded by*

$$\tau \geq \frac{C}{I_{adv} + I_{scan}} \approx \frac{C}{I_{adv} + 20mA}, \quad (12)$$

where $I_{adv} = \frac{\sum_{p \in P} (I_p d_p)}{T}$ denotes the average current drawn by the 9 major phases during an advertising event.

Proof. By substituting energy consumption by the overlay mesh, ie., Eq. (11), into Eq. (10), we have

$$\begin{aligned} \tau &= C \frac{V_s T}{E_{mesh}} \\ &= \frac{CT}{\sum_{p \in P} (I_p d_p) + I_{scan} (T - \sum_{p \in P} d_{p_i})}, \end{aligned}$$

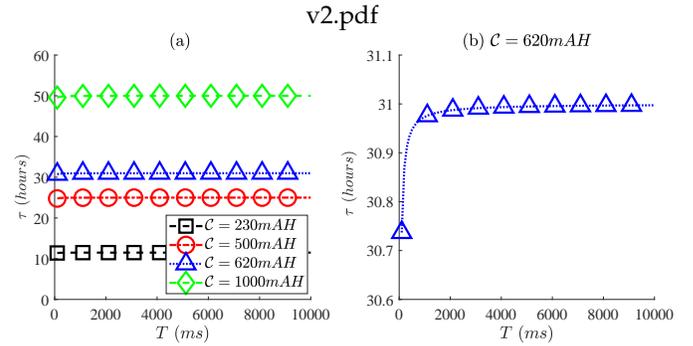


Fig. 7: (a) The lifetime of the underlying beacon is less dependent on T , regardless of the battery capacity. (b) By zooming in the blue curve, we can observe that there is a slight drop in the lifetime when T is short. This is due to the high frequency of advertising events, which became the dominant factor in determining the lifetime of the underlying beacon.

where $T = T_s = T_w$. Since T is much larger than the active duration defined by all the phases, i.e., $T \gg \sum_{p \in P} d_{p_i}$, the duration of those active phases contributes very minimal effect to the lifetime computation. Hence, we can approximate the summation terms to zero,

$$\begin{aligned} &\frac{CT}{\sum_{p \in P} (I_p d_p) + I_{scan} (T - 0)} \\ &\Rightarrow \frac{CT}{\sum_{p \in P} (I_p d_p) + I_{scan} T} \\ &\Rightarrow \frac{C}{\frac{\sum_{p \in P} (I_p d_p)}{T} + \frac{I_{scan} T}{T}} \\ &\Rightarrow \frac{C}{I_{adv} + I_{scan}} \end{aligned}$$

which completes the proof. \square

Clearly, the underlying beacon is guaranteed to operate for at least $\frac{C}{I_{adv} + 20mA}$, as according to Eq. (11). Fig. 7 shows the lifetime of the underlying beacon given different battery capacities. From Fig. 7(a), we can see that τ is less dependent on T , which verifies Eq. (11). When T is small, the frequency of advertising events increases, I_{adv} became the dominant factor determining the lifetime rather than I_{scan} . Since I_{adv} is generally less than 20 mA, the underlying beacon has a slightly longer lifetime when its current consumption is mainly dominated by the advertising event, as shown in Fig. 7(b).

5.2 Duty-cycle Scanning

For duty-cycle scanning, the question is how long shall be the scanning windows? According to Fig. 8, we can see that the beacon can last for 8568.9 hours (i.e., at least 11 months) with $\mathcal{D} = 0.001$ and $T = 1s$. However, the beacon can only last for 1159.8 hours (i.e., at least 1.6 months) when T is reduced to 100ms (with the same \mathcal{D}). Such a lifetime degradation is mainly due to the high frequency of advertising events. Fig. 8 illustrates the implications of I_{adv} and I_{scan} on the underlying beacon's lifetime for different T . It is clear that the lifetime of the underlying

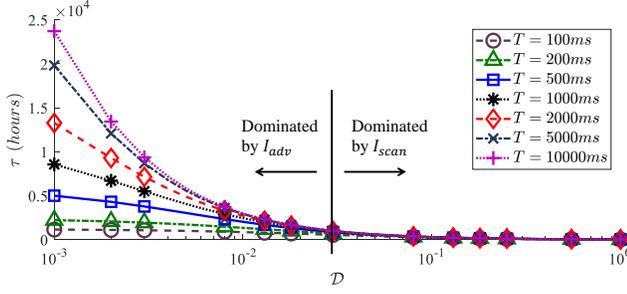


Fig. 8: The relationship between duty-cycle scanning \mathcal{D} and lifetime τ unveils that the lifetime of the underlying beacon is subject to the current drawn by the advertising event when \mathcal{D} is small; whereas when \mathcal{D} is large, the lifetime degrades severely in consequence to the high current drawn by the scanning event.

beacon is highly dependent on I_{scan} when $\mathcal{D} \geq 0.05$. When $\mathcal{D} < 0.05$, then the lifetime of the underlying beacon is mainly subject to the configuration of the advertising event. Given such observations, the next question is what would be the minimum \mathcal{D} such that to have a feasible scanning?

A scanning event is considered infeasible if it is relatively hard, if not impossible, for the device to capture the incoming packet. Specifically, it is infeasible to have a very short scanning window in which the node will forever miss the incoming packet. According to the energy analysis in Section 4.2, we argue that the scanning is feasible as long as the scanning window T_w is greater or equal to the active duration of the advertising event $d_{active}(A)$. Hence, the lowest feasible $\tilde{\mathcal{D}}$ is dependent on $d_{active}(A)$ rather than $d_{active}(S)$.

Lemma 1. (The Lowest Feasible $\tilde{\mathcal{D}}$) To achieve a feasible scanning, the scanning duty cycle \mathcal{D} should not be lower than

$$\frac{d_{active}(A)}{d_{sleep}(A)}. \quad (13)$$

Proof. The scanning duty cycle for the underlying beacon can be described as follows:

$$\mathcal{D} = \frac{T_w}{T_s - d_{active}(A)} = \frac{T_w}{d_{active}(S) + d_{sleep}(S) - d_{active}(A)}.$$

In contrast to a pure scanning event, we need to subtract the scanning interval of the underlying beacon with the active period taken by the advertising event.

Since a feasible scanning requires $T_w \geq d_{active}(A)$, then,

$$\frac{d_{active}(A)}{d_{active}(S) + d_{sleep}(S) - d_{active}(A)}.$$

Since $T_a = d_{active}(A) + d_{sleep}(A)$, we have

$$\begin{aligned} d_{active}(S) + d_{sleep}(S) - d_{active}(A) \\ = T - d_{active}(A) = d_{sleep}(A), \\ \Rightarrow \frac{d_{active}(A)}{d_{sleep}(A)}, \end{aligned}$$

which proves that the lowest feasible $\tilde{\mathcal{D}}$ is solely dependent on the active duration of the advertising event. \square

Since $d_{active}(A)$ for the advertising event with non-connectable PDU is at least $3.70ms$ (cf. Table 2), then the lowest feasible $\tilde{\mathcal{D}}$ is $\tilde{\mathcal{D}} = \frac{3.7}{996.3} = 0.0037$ given $T = 1s$. In this case, the underlying beacon is guaranteed to have a lifetime of 5348.9 hours (i.e., at least 7.4 months) with $C = 620mAH$. Given the above lemma, the upper bound of the lifetime can be derived.

Theorem 2. (Upper Bound of the Lifetime) The lifetime of an underlying beacon is upper bounded by

$$\tau < \frac{Cd_{sleep}(A)}{d_{sleep}(A)(I_{adv} + I_{sleep}) + d_{active}(A)(I_{scan} - I_{sleep})}. \quad (14)$$

Proof. By using the same proofing procedure as in Theorem 1, we have

$$\begin{aligned} \tau &= C \frac{V_s T}{E_{mesh}} \\ &= \frac{CT}{\sum_{p \in P} (I_p d_p) + T(I_{scan} \mathcal{D} + I_{sleep}(1 - \mathcal{D}))}, \\ &= \frac{C}{I_{adv} + (I_{scan} \mathcal{D} + I_{sleep}(1 - \mathcal{D}))}, \\ &= \frac{C}{I_{adv} + I_{sleep} + \mathcal{D}(I_{scan} - I_{sleep})} \end{aligned}$$

where $I_{adv} = \frac{\sum_{i=1}^n (I_{p_i} N_{p_i} d_{p_i})}{T}$ denotes the average current drawn by the advertising event during the active period.

To achieve a feasible scanning, the scanning duty cycle should be at least satisfy the lowest feasible $\tilde{\mathcal{D}}$. Hence, by Lemma 1, we have

$$\begin{aligned} &\frac{C}{I_{adv} + I_{sleep} + \frac{d_{active}(A)}{d_{sleep}(A)}(I_{scan} - I_{sleep})} \\ \Rightarrow &\frac{Cd_{sleep}(A)}{d_{sleep}(A)(I_{adv} + I_{sleep}) + d_{active}(A)(I_{scan} - I_{sleep})}, \end{aligned}$$

which completes the proof. \square

Clearly, the maximum lifetime we can achieve is upper bounded by Eq. (14), which indicates the dependency of the lifetime on I_{adv} , I_{scan} and I_{sleep} .

6 PACKET FORWARDING ACTIVITY THROUGH THE OVERLAY MESH

While it is possible to sustain the lifetime of the underlying beacon with duty-cycle scanning, it is relatively challenging to disseminate the packet to all the nodes with duty-cycle scanning. In other words, using duty-cycle scanning compromises the ultimate goal of overlay mesh when most of the nodes fail to receive the incoming packet. In this section, we discuss packet dissemination through the overlay mesh. After that, we present our proposed overlay protocol that improves the PDR while using the lowest feasible $\tilde{\mathcal{D}}$ to sustain the lifetime of the underlying beacon.

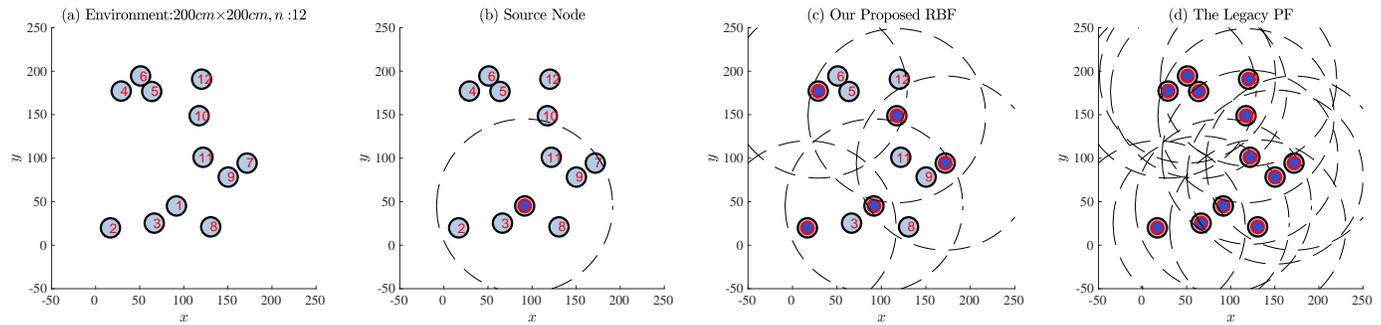


Fig. 9: Illustration of the packet forwarding activity over overlay mesh network: (a) a simulation environment with a total of 12 nodes; (b) node 1 initiates that it has a packet needed to be forwarded; (c) the RBF proposed in our previous work [5] is capable of disseminating the packet to every node with less number of transmissions, compared to the redundant transmission by (d) the legacy flooding strategy.

6.1 Packet Dissemination Rate (PDR)

We define PDR as a ratio to quantify the number of nodes that have successfully received the incoming packet to the total number of nodes in the overlay mesh network $G_o = (B, R, L, V)$. Let k denotes the number of nodes that have received the packet, PDR can be expressed as follows:

$$\text{PDR} = \frac{k}{|R| + |B| - 1}, \quad (15)$$

where $|B|$ and $|R|$ indicate the total number of underlying beacons and receivers, respectively.

Let node b_m be the mesh node that initiates a packet forwarding request by encapsulating the additional information and the request flag into the advertising payload and header, respectively. The resultant packet is known as the meshing packet, which is scheduled to be broadcast at the next $T = T_a$. Ideally, all neighboring nodes to node b_m should be able to receive the packet if all of them employed continuous scanning. For those nodes located outside the transmission range of b_m , they can still receive the packet through their neighboring beacons at the next advertising interval. However, if all the receiving nodes attempt to forward the packet through their next advertising interval, it might jeopardize the network traffic with redundant packet transmission. In our previous work [5], we employed RBF to decide if the node should participate in the packet forwarding activity. Fig. 9 illustrates our simulation environment with a total of 12 nodes and node 1 being the mesh node. Fig. 9(b) shows the neighboring nodes that have a virtual link to node 1. Fig. 9(c) shows that our RBF approach is capable of disseminating the packet to every node with less redundancy compared to the legacy probabilistic flooding (PF) approach.

For both approaches, we achieve $\text{PDR} = 1$ with continuous scanning. However, as discussed in Section 5, continuous scanning is a threat to the lifetime of the underlying beacon. On the other hand, duty-cycle scanning, even though it can guarantee a longer lifetime, imposes another challenge in disseminating the packet.

6.2 Time Slots Synchronization

In fact, it is not obvious to synchronize the time slots for all nodes because each node has its own chipset that

defines the local timer for advertising and scanning events. In this paper, we used a more powerful device (e.g., the smartphone) that has the Internet connection as the central device to synchronize the time slot on each beacon. To achieve this, we make all the beacon nodes enter into the continuous scanning mode directly once they are powered up. The advertising event is disabled at this stage. The continuous scanning mode allows the node to listen to the command from the smartphone. Upon receiving the command from the smartphone, each node will disable the continuous scanning and start to initiate time for advertising event following the time command t_c .

The smartphone is used to command all the nodes to follow the global timing, determined by the smartphone. After that, a simple computation can be imposed such that each node only has to trigger their events when $\text{mod}(t, t_\gamma) = 0$, where t_γ defines the desired duration of each slot. For a fixed interval T , there will be a total of $N = \text{abs}(\frac{T}{t_\gamma})$ number of slots. The synchronization process is illustrated in Algorithm 1.

6.3 Scanning Policies

Let t_γ be the desired duration of each slot, t_γ can be defined according to the active duration of the advertising event. Table 2 shows that the active duration for an advertising event with non-connectable PDU is at least 3.7ms . Hence, $t_\gamma = 4\text{ms}$ should be sufficient while allowing extra 0.3ms tolerance for processing time. Given a fixed interval T , there will be a total of $N = \text{abs}(\frac{T}{t_\gamma})$ number of slots. Normally, the advertising event will take one of the time slots, as shown in Fig. 10, leaving only $N - 1$ times slots for the node to trigger the scanning event. Fig. 10 illustrates the packet dissemination initiated by the first mesh node b_1 to every node. Node b_2 and b_3 are the neighboring nodes to b_1 , whereas node b_4 is not. b_3 triggers the scanning event in the same time slot as the advertising event triggered by b_1 . Hence, b_3 receives the packet and then encapsulates the received packet in its advertising packet which is scheduled to be broadcast in the next advertising interval. Both b_2 and b_4 can then receive the packet from the next mesh node b_3 .

6.4 Performance Analysis of Packet Dissemination

Let $X_v = \{0, 1\}$ be the Bernoulli random variable indicating the success of node b_v in receiving the meshing packet,

Algorithm 1: Time Slots Synchronization

```

Input : Time command from smartphone:  $t_c$ 
Output: The time to initiate the advertising event:
 $t_{a,b_i}^{(init)}, \forall b_i \in B$ 

// beacon node's subroutine
1 Each node enter a continuous scanning mode.
2 while no command received do
3   if command received then
4     stop scanning
5     trigger the advertising event according to  $t_c$ 
6     divide the time slot according to the desirable
       duration  $t_\gamma$ 
7     allocate the scanning event into one of the
       time slot.
8   end
9 end

// smartphone's subroutine
10 while number of synchronized nodes  $< |B|$  do
11   count the number of nodes that has successfully
       trigger the advertising event.
12   if number of synchronized nodes equals  $|B|$  then
13     listed the nodes that are ready for
       deployment
14     break
15   end
16 end

```

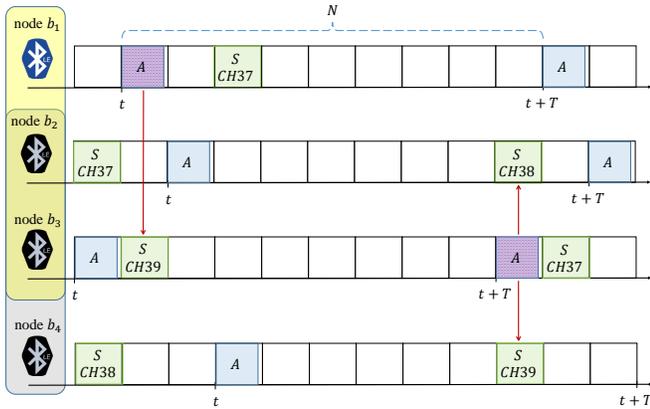


Fig. 10: An illustration of packet dissemination with time-slotted duty-cycle scanning. The purple shaded block denotes the advertising packet containing additional information and with forwarding request flag set, the green block the scanning event, and the blue block the advertising event. Node b_4 is located outside the transmission range of b_1 , but inside the range of b_2 and b_3 .

$\forall b_v \in B/\{b_m\}$. Suppose that the number of virtual links connecting node b_m with the other nodes is $V_m \subset V$, we can describe the probability of $X_v = 1$ as follows:

$$\mathbb{P}(X_v = 1) = \left(\frac{1}{N-1} \right) \left(\frac{|V_m|}{|B|-1} \right) \quad (16)$$

Intuitively, node b_v should satisfy the following two conditions before it can receive the packet from b_m :

- 1) the scanning event of node b_v is triggered at the same time slot as the advertising event of b_m ;
- 2) there exists a virtual link between node b_v and b_m .

Here, we assume a perfect channel condition with no synchronization errors to transmit a packet from one node to another node. Given such an ideal condition, our goal is to verify the PDR when the duty-cycle scanning is applied. That being said, if the PDR is very low under such an ideal condition, then we can anticipate that the PDR would be worst in practical scenarios.

Now, let $k = \sum_{b_v \in B/b_u} X_v$ be the total number of nodes that has successfully receive the packet and $p_r = \mathbb{P}(X_v = 1)$, we can establish the following lemma.

Lemma 2. *The total number of nodes that can receive the packet at z th interval has the following probability distribution:*

$$\begin{aligned} \mathbb{P}(k, z = 1) &= \begin{cases} \binom{|B|-1}{k} p_r^k (1-p_r)^{|B|-1-k}, & 0 \leq k \leq |B|-1 \\ 0, & k > |B|-1 \end{cases} \\ \mathbb{P}(k, z > 1) &= \begin{cases} \sum_{i=0}^{k-z-1} p'_{k-i} p_z \binom{|B|-1}{i} p_r^i (1-p_r)^{|B|-1-i}, & z-1 \leq k \leq |B|-1 \\ 0, & k < z-1, k > |B|-1 \end{cases} \end{aligned} \quad (17)$$

where $p'_{k-i} = P(k-i|\#(M) = m-1)$ denotes the probability of having $k-i$ nodes successfully receive the packet in the previous interval, and p_z denotes the probability of the receiving node to trigger the next interval.

Proof. The proof is straightforward by applying the direct induction proofing method and thus the proof is omitted. \square

The probability of having k nodes successfully receive the packet at the first interval is dependent on the two conditions specified by Eq. (16). Fig. 10 clearly describes these two conditions. Upon receiving the packet, the node can help to further forward the packet at the next interval. Note that the number of nodes that have received the packet at $z > 1$ should not be less than $z-1$ because it is impossible to further forward the packet if none of the nodes has received the packet in the previous interval, hence, k must be at least equal to $z-1$. The probability of having k nodes successfully receive the packet after z intervals are dependent on 1) the total number of receiving nodes resulting from previous meshing p'_{k-i} and 2) the probability of current receiving nodes to attempt to forward the packet p_z . Our previous RBF approach leverages the RSS information to define the meshing probability p_z .

Monte Carlo Simulation. The simulation environment consists of 100 nodes randomly distributed within an area of $100m \times 100m$. We divide the time slot by setting $t_\gamma = 4ms$ and $T = 1s$. Each node can initiate their first advertising event randomly at any time slot and then repeat the same event at every interval T . During the simulation, a node will be selected randomly to trigger the packet forwarding activity, and the number of nodes that receive the meshing

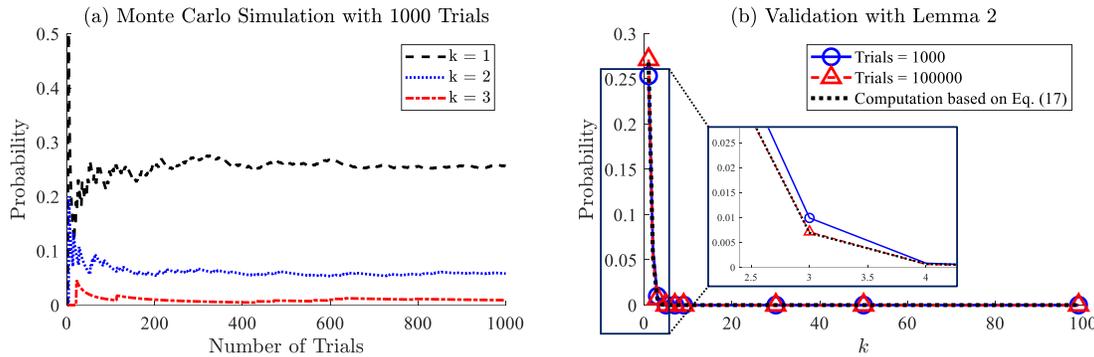


Fig. 11: Validation with Monte Carlo simulation.

packet is recorded. We ran the simulation for 1000 trials, and the result is shown in Fig. 11(a). The probability to have more than one node (i.e., $k > 1$) successfully receive the packet is very low (i.e., less than 0.1) when we define the time slot according to the lowest feasible \bar{D} . We validate the simulation results with Lemma 2, as shown in Fig. 11(b). The simulation results converge to the probability distribution defined in Eq. (17), as the number of trials increased.

The above results imply that it is very difficult to achieve good PDR when using duty-cycle scanning. However, this paper aims to increase the likelihood of the duty-cycle scanning event to capture the incoming packet at the intermittent interval, while still using the lowest feasible \bar{D} . To achieve this, we define an overlay protocol that enables the node to explore the best time slot before locking the scanning event to that particular time slot.

6.5 Proposed Overlay Protocol

Upon synchronizing the time slots between all the nodes, the problem is reduced to selecting the best time slot that can improve PDR. Hence, we propose an overlay protocol that conforms to the following policies:

- 1) Each node listens to all the $N - 1$ time slots until it found a particular time slot in which it can hear the most packets.
- 2) the node reschedules its scanning event to that particular time slot and includes this information in the header of its advertising packet.
- 3) this additional information in the header allows the neighboring nodes to decide if they should lock their scanning event to the same time slot or explore other possible slots.

Intuitively, if many nodes already locked a particular time slot that can hear the most packets, the rest of the nodes should give up this slot and continue to explore other possible slots. This ensures that each node will be at least heard by one of the neighboring nodes, rather than having all the nodes listen to the same time slot. Let κ be the threshold denoting the maximum allowable number of nodes that can trigger the scanning event in the same time slot, κ can be set to the network size. Then, at every iteration, we update the κ by dividing κ by two. The idea is to reduce the number of nodes that take on the same scanning slots but leave other slots that can also hear the

incoming packets unattended. If any node detected their current selected slot already reach the capacity defined by κ , it will have to give up their current selected slot and explore other possible slots. In this way, we can still keep a large number of beacons to scan on the slot that can hear the most packets, while covering the rest of the slot that can hear at least one packet. The detailed algorithm is described in Algorithm 2.

6.6 Simulation and Results

We first verified the effect of κ on selecting the scanning slot, then we examine the PDR performance in terms of network size and BLE's transmission power. We set up the simulation by randomly deploying the beacons within an area of 100 m \times 100 m. Any beacon located within the transmission range of another beacon is considered as a neighbor to that beacon. In the simulation, we consider an idea channel condition in which the beacon can receive the packet as long as it is within the transmission range. However, if two beacons in the same neighborhood transmit the packet at the same time, both packets will experience packet collision. The beacon is configured to have a 1 s advertising interval. Hence, the total number of slots we can have by using the lowest feasible duty cycle is $\frac{1000}{4} = 250$ slots. Each beacon is initiated to advertise their packet at one of the slots, and repeat the advertising every 1 s. Fig. 12 illustrates one of the simulation setup with 100 beacons. Fig. 12(a) shows the advertising slots taken by the 100 beacons, and we can see that there are about 4 beacons taken the same advertising time slots. Since these 4 beacons are not in the same neighborhood, no packet collision happened even they transmit in the same time slot. Fig. 12(b) shows the scanning time slots taken by listening to the time slots that can receive the most packets. Fig. 12(c), on the other hand, shows the scanning time slots after applying our overlay protocol.

We started the simulation with 50 beacons, and then increased the number of beacons to 1000, with 50 increments each time. Each simulation was repeated 50 times, and the results were averaged and logged for further analysis.

6.6.1 Threshold κ

We examined the number of packets that can be heard by at least one node when κ is decreased by half at every iteration. At every iteration, we computed the ratio of packet heard over network size. If the packets transmitted by all the

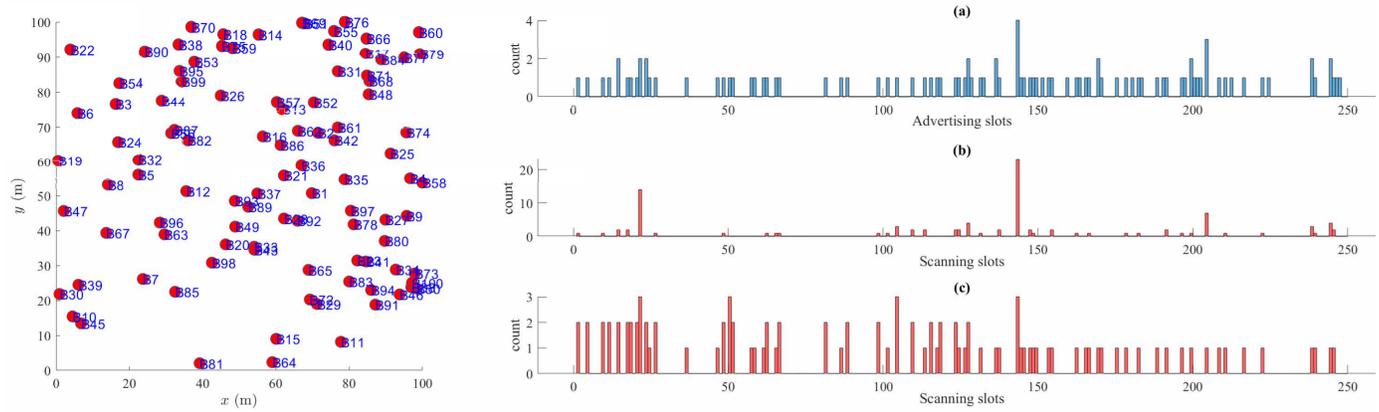


Fig. 12: Illustration of network size with 100 beacons: (a) the advertising time slot initiated by the 100 beacons, (b) the initial scanning time slots taken by the 100 beacons according to policy 1, (c) the rescheduled time slots according to policy 2 and 3, enabling by Algorithm 2.

Algorithm 2: Overlay Protocol - Policy 3

Input : advertising packets

Output: index of a single time slot

```

1 initiate the scanning on all the  $N - 1$  time slots
2 set  $L$  to an empty dictionary object
3 set  $\kappa$  equals to the size of the network
4 current selected slot = the slot that can hear the most
  packets
5 while no packet received do
6   do scanning
7   if at least one packet received and  $\kappa > 0$  then
8     Update  $\kappa = \kappa/2$ 
9     Parse the packet header to acquire the
      scanning time slot of the corresponding
10    store the information of the scanning time
      slot of the receiving node into a list  $L$ 
11    foreach  $l$  in  $L$  do
12      if the current selected slot equals to  $l.slot$ 
13        then
14          | increase the count;
15        end
16        if count  $< \kappa$  then
17          | schedule the scanning event to the
            current selected slot
18          | include this information in its own
            header
19        end
20        else
21          | schedule the scanning event to the
            second best time slot
22          | update this information in its own
            header
23        end
24      end
25    end

```

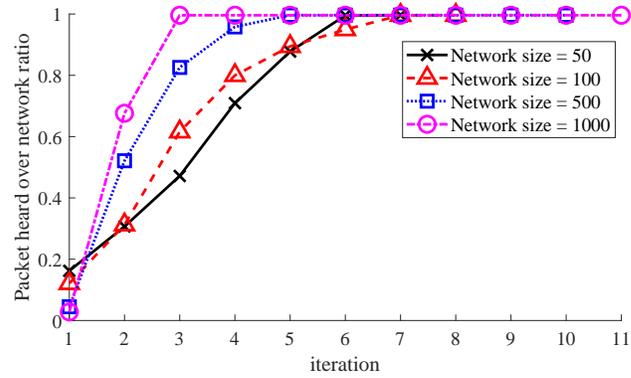


Fig. 13: The ratio of packet heard over network size at each iteration.

beacons can be heard by at least one beacon (i.e., there is at least one beacon listening to the incoming packet from a beacon), then the ratio is one. If the packets transmitted by some of the beacons are not received by any beacon in the network, the ratio is less than 1. Note that it is impossible to achieve ratio one when there are some packet collision events. We ran the simulation 50 times for each of the network sizes, and the results for network size equals 50, 100, 500, and 1000 are averaged and plotted, as shown in Fig. 13.

Note that the final value is not one, but 0.9964, 0.9964, 0.9963, 0.9960 for network size 50, 100, 500, and 1000, respectively. The reason is that at some runs, the packet heard ratio is one, but sometimes it cannot reach one due to packet collision. It is clear that the packet collision increases when the network size increases. The plot also shows that the algorithm required more iterations to converge when the network size is small. This is because the beacon needs more time to find the next slot that contains the incoming packets. Whereas when the network size increases, almost most of the time slot containing at least one packet, thus it took lesser iterations for the beacon to find the next slot. The results proved that our overlay protocol can better schedule the scanning time slot to ensure the packet transmitted by the beacon is at least heard by one of the beacons.

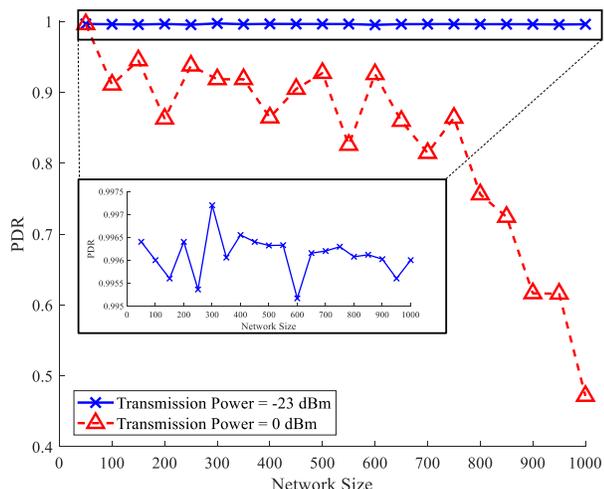


Fig. 14: The effect of network size and transmission power on packet dissemination.

6.6.2 Effect of Network Size and Transmission Power

After each beacon has rescheduled its scan slot, we further examined the effect of network size and transmission power on PDR. Fig. 14 shows the PDR for network size ranges from 50 to 1000, and for transmission power equals 0 dBm and -23 dBm. When the transmission power is equaled to 0 dBm, the packet can travel for a longer distance. However, when the network size increased, such a long transmission range can cause more packet collisions and eventually affect the PDR. On the other hand, we can see that the PDR is quite consistent regardless of the network size when transmission power equals -23 dBm. When we zoomed in on the plot, we can see that there is a slight variation in PDR for different network sizes, and the highest PDR is achieved when the network size is equaled to 500. One possible reason is that when the network size is small, some beacons might be located very far from each other and their transmission range is not long enough to deliver the packet to the beacon located farther away. When the network size increased, packet collision occurred and affected the PDR. However, the PDR is still quite high compared with the one with 0 dBm transmission power. This is because when the transmission range is short, the number of packet collisions also decreased. The results show that PDR performance is quite consistent when each beacon selects the scanning slot following our overlay protocol. In general, low transmission power is preferred when the network size increased as compared to high transmission power.

7 EVALUATIONS OF APPLICATIONS

We built a proof-of-concept prototype with off-the-shelf beacons¹, in which each beacon is built with the CC2541 chipset from Texas Instrument. Two different applications, smart control, and smart relay were developed to demonstrate the feasibility of our overlay mesh. Fig. 15 illustrates these two applications.

- 1) **Smart control** (Fig. 15(a)): The Bluetooth mesh allows us to control multiple devices using a single

command. In this application, we show how we can control multiple LEDs with a single button click. These LEDs were distributed randomly in a large area, and some of the LEDs are outside the range of the beacon that initiates the control command.

- 2) **Smart relay** (Fig. 15(b)): The path diversity of the mesh network enables the packet to be delivered to the receiving end even some nodes fail to function properly. In this application, we show how we can forward the proximity information from a remote location (underground) that does not have any Internet connectivity to the receiver located in the upper floor.

Besides c4 and c8, the smartphone N8 also functions as a beacon. This can be done by programming the smartphone to work as a beacon. The reason to use the smartphone as a beacon is because we can demonstrate the smart control and smart relay applications by manipulating the rich hardware embedded within the smartphone. Note that this can also be done by adding a button, or a sensor to the existing beacon hardware. However, we opt to use the smartphone to provide a quick demonstration. Fig. 15(a) shows that we can light up the LEDs connected to the two beacons by clicking the button on N8; whereas Fig. 15(b) shows that we can relay the ambient light information via the beacons to the receiver. Another smartphone on the left-hand side is programmed to work as a receiver, at the same time logging the information regarding the advertising packet broadcast by the beacon. The purpose is to show that all beacons can still retain their own advertising events while being part of the overlay mesh network.

7.1 Demonstrations with Large Scale Testbeds

We further set up two testbeds to demonstrate the two applications described above. For the smart control application, we randomly scattered the beacons around a big laboratory. The purpose is to show that the node can successfully disseminate the lighting command to all the nodes even though some of them are outside the transmission range. Fig. 16 shows the experimental environment with a total of 12 nodes. The parameters setting of each beacon is similar to the one described in Table 1. As illustrated in Fig. 16(a) and (b), we can change the LED color with single button control. The experiment was conducted at least 100 times and the total time required to turn on all the LEDs were jotted down. The same experiment was repeated by varying the number of nodes from 12 to 2, with 2 decrements each step. The corresponding results are discussed in the next subsection.

For the smart relay application, we deployed a large testbed on a multi-floor building, as depicted in Fig. 17. The lower floor is assumed to be in the underground that does not have Internet connectivity, whereas the upper floor is a smart environment that can connect to the Internet. A group of beacons deployed in the lower floor was working as a source beacons that broadcast proximity information about the possible nearby object, and the beacons deployed along the staircase and along the corridor of the second floor were working as a relay that can forward the packet they received from the source beacons. We randomly walked along the

1. "Ghostyu Beacon", <http://ghostyu.com>



Fig. 15: Two applications were developed to demonstrate the feasibility of overlay mesh: a) controlling multiple nodes with a single command, and b) relaying the ambient light information.

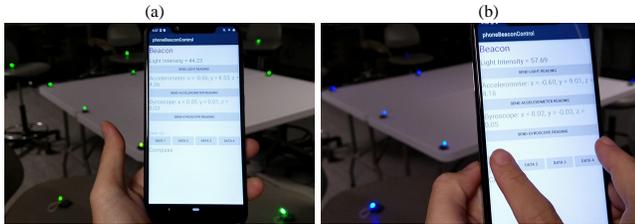


Fig. 16: There are a total of 12 nodes scattered around the laboratory. The transition from (a) and (b) show that we can control multiple LEDs with a single button click.

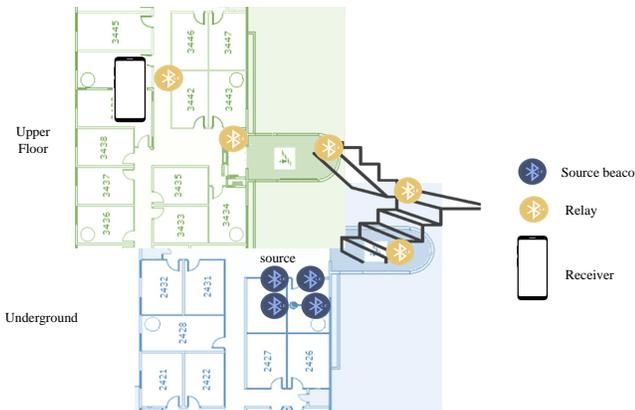


Fig. 17: The overlay mesh testbed consists of a group of beacons acting as a source, a group of beacons acting as a relay, and a smartphone as a receiver.

corridor in the upper floor with the smartphone to check for the received packet. The smartphone is also programmed to log the packet delivery time. The experiment was repeated by adding the number of relays and the number of beacons to the environment, and the experimental result is summarized in Table 3. The time indicated the average time to receive all the packets broadcast by the deployed beacons. So, if there are 20 source beacons, we average the arrival time of all the 20 packets. The overlay mesh testbed consists of a group of beacons acting as a source, a group of beacons acting as a relay, and a smartphone as a receiver.

7.2 Latency Evaluation and Discussion

Let t_d denote the time duration. Specifically, for the smart control application, t_d refers to the total time required to disseminate the packet to all the nodes; whereas for the

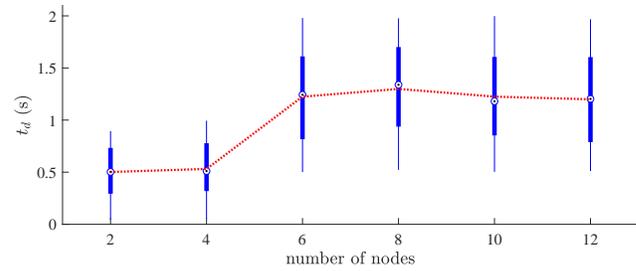


Fig. 18: The total time required to turn on all the LEDs.

smart relay applications, t_d measures the time for the packet to travel from the mesh node to the receiver.

7.2.1 Latency of Packet Dissemination to all the Nodes

Fig. 18 describes the results to disseminate the lighting command to all the nodes. It is interesting to note that the time required to turn on all the 12 LEDs is almost the same and sometimes shorter than the time required by 8 or 6 LEDs. This can be explained by the fact that more nodes were attempting to forward the packet when the number of nodes increased. Hence, some nodes outside the range might be able to receive the command quicker through their immediate neighbors.

7.2.2 Latency of end-to-end Packet Traveling

From Table 3, we can see that the more relays, the less time for all the packets to reach to the destination. This is especially through when the number of source beacons is between 35 to 50. This can be explained by the fact when the number of relays increases, it provides more paths for the packet to travel to the receiver. However, when the number of source beacons is less (i.e., between 5 to 15), all the packets can be delivered faster with fewer relays. The problem here is that when there are only a few packets from the source beacons to be delivered, and there are many relays to help to deliver the same packet, the same packet might experience packet collision and thus increasing the average packet delivery time (here, the average packet delivery time is computed by averaging the packet arrival time for all the packets from all the source beacons). The result in Table 3 shows that more relays are not necessarily useful to the packet delivery. Future work can cast this as an optimization problem so as to find the best number of relays for different scenarios containing a different number of source beacons.

TABLE 3: The average time (in ms) required to forward the packet from the source beacons to the receiver.

		Number of source beacons									
		5	10	15	20	25	30	35	40	45	50
Number of relays	20	412	480	497	511	521	580	733	845	897	913
	40	306	378	411	450	481	564	701	754	874	889
	60	310	388	397	427	468	510	654	695	761	784
	80	308	376	403	408	422	487	554	625	677	681
	100	312	388	401	457	492	503	511	519	612	623

7.2.3 Lifetime of Underlying Beacons

According to the theoretical results discussed in Section 5, the underlying beacon can only last for no more than 31 days when continuous scanning is employed. On the other hand, we can improve the lifetime of the underlying beacon to at least 7.4 months if the lowest feasible duty cycle is used. To verify the theoretical results, on 10 March 2021, we purposely set aside two beacons (beacon A and beacon B) powered up by a 620 mAH battery to examine how long they can last. Beacon A is configured to scan continuously with an advertising interval equal to 1 s, whereas beacon B is configured to scan with the lowest feasible duty-cycle with the same advertising interval. We used the smartphone to check the advertising packet broadcast by the beacon every day. If the smartphone can still receive the packet from the beacon, it means the beacon is still working. The smartphone can receive the advertising packet from beacon A in the first two days. However, no more advertising packet from beacon A is detected on day 3, day 4, and so on. For beacon B, the smartphone can still receive the advertising packet at the time of writing (14 June 2021). The current results verify that the beacon can last for more than 3 months (and it is still working now) when using the lowest feasible duty cycle unveiled from our previous analysis while improving the performance of packet dissemination rate.

8 CONCLUSIONS

BLE beacon can be easily deployed in a harsh environment (e.g., underground mining site, cave, etc.) where no WiFi nor cellular signals are available. Hence, we can use BLE beacons to monitor the remote location and disseminate emergency information, if any. This is only possible if we deployed a gateway to collect the packet broadcast by BLE beacons. The gateway can then upload the data to the cloud server for remote monitoring. However, as discussed, it is hard to have Internet connectivity in these remote locations. On the other hand, all the beacons are selfish nodes that keep broadcasting their own advertising packet and never listen to their neighboring nodes.

This paper presents a BLE-based overlay mesh enabling each node to listen to one another and help to forward the packet whenever there is a packet forwarding request. We propose an energy-efficient overlay protocol to minimize the energy consumed by advertising and scanning events while improving the packet dissemination rate. Our proposed protocol enables the overlay mesh to leverage the already deployed beacons for packet forwarding without affecting its low power feature. With our protocol, each node can retain its periodic advertising event while facilitating the

intermittent packet forwarding request over the overlay mesh network. Extensive experiments with two applications demonstrate the feasibility of our overlay mesh for real-world use cases.

Our overlay protocol is only possible when every beacon has its time slots synchronized. Currently, we used a manual approach by leveraging a BLE App installed in a smartphone to configure the beacon’s parameters and ensure all of them initiate their time slots according to the command sent from the smartphone. Manual routine reinitialization is required from time to time to avoid synchronization errors due to time drift and device issues. Rather than routine reinitialization, future work can be conducted to address the synchronization issue automatically across large-scale beacon networks. Besides the synchronization issue, another possible direction we can consider is to integrate the network coding to jointly transmit the advertising packet and multiple packets containing the sensing information. If an efficient network coding approach can be developed to encode and decode different payloads into a single packet and then transmit the encoded packet through the same advertising slot, it can further enhance the packet dissemination rate while retaining the original message that should be broadcast by the beacon network.

ACKNOWLEDGMENT

This work was supported by HKUST-NIE Social Media Laboratory. The BLE beacons and chipsets were sponsored by CyPhy Limited, Hong Kong.

REFERENCES

- [1] The full story of thailand’s extraordinary cave rescue. [Online]. Available: <https://www.bbc.com/news/world-asia-44791998>
- [2] P. C. Ng, J. She, and S. Park, “High resolution beacon-based proximity detection for dense deployment,” *IEEE Transactions on Mobile Computing*, vol. 17, no. 6, pp. 1369–1382, June 2018.
- [3] P. Spachos and K. N. Plataniotis, “Ble beacons for indoor positioning at an interactive iot-based smart museum,” *IEEE Systems Journal*, pp. 1–11, 2020.
- [4] P. C. Ng, J. She, and S. Park, “Notify-and-interact: A beacon-smartphone interaction for user engagement in galleries,” in *2017 IEEE International Conference on Multimedia and Expo (ICME)*, July 2017, pp. 1069–1074.
- [5] P. C. Ng and J. She, “A novel overlay mesh with bluetooth low energy network,” in *2019 IEEE Wireless Communications and Networking Conference (WCNC)*, April 2019, pp. 1–6.
- [6] R. R. Rout and S. K. Ghosh, “Enhancement of lifetime using duty cycle and network coding in wireless sensor networks,” *IEEE Transactions on Wireless Communications*, vol. 12, no. 2, pp. 656–667, February 2013.
- [7] S. Wu, J. Niu, W. Chou, and M. Guizani, “Delay-aware energy optimization for flooding in duty-cycled wireless sensor networks,” *IEEE Transactions on Wireless Communications*, vol. 15, no. 12, pp. 8449–8462, Dec 2016.

- [8] K. Han, Y. Liu, and J. Luo, "Duty-cycle-aware minimum-energy multicasting in wireless sensor networks," *IEEE/ACM Transactions on Networking*, vol. 21, no. 3, pp. 910–923, June 2013.
- [9] A. Liendo, D. Morche, R. Guizzetti, and F. Rousseau, "Efficient bluetooth low energy operation for low duty cycle applications," in *2018 IEEE International Conference on Communications (ICC)*, May 2018, pp. 1–7.
- [10] K. E. Jeon, J. She, P. Soonsawad, and P. C. Ng, "Ble beacons for internet of things applications: Survey, challenges, and opportunities," *IEEE Internet of Things Journal*, vol. 5, no. 2, pp. 811–828, April 2018.
- [11] F. Yin, Y. Zhao, F. Gunnarsson, and F. Gustafsson, "Received-signal-strength threshold optimization using gaussian processes," *IEEE Transactions on Signal Processing*, vol. 65, no. 8, pp. 2164–2177, April 2017.
- [12] P. C. Ng, J. She, and R. Ran, "A compressive sensing approach to detect the proximity between smartphones and ble beacons," *IEEE Internet of Things Journal*, pp. 1–1, 2019.
- [13] A. Montanari, S. Nawaz, C. Mascolo, and K. Sailer, "A study of bluetooth low energy performance for human proximity detection in the workplace," in *2017 IEEE International Conference on Pervasive Computing and Communications (PerCom)*, March 2017, pp. 90–99.
- [14] M. Wang and J. Brassil, "Managing large scale, ultra-dense beacon deployments in smart campuses," in *2015 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, April 2015, pp. 606–611.
- [15] P. C. Ng, J. She, P. Spachos, and R. Ran, "A fast item identification and counting in ultra-dense beacon networks," in *GLOBECOM 2020 - 2020 IEEE Global Communications Conference*, Dec 2020, pp. 1–6.
- [16] P. C. Ng and J. She, "Denoising-contractive autoencoder for robust device-free occupancy detection," *IEEE Internet of Things Journal*, vol. 6, no. 6, pp. 9572–9582, Dec 2019.
- [17] P. C. Ng, J. She, and R. Ran, "A reliable smart interaction with physical thing attached with ble beacon," *IEEE Internet of Things Journal*, vol. 7, no. 4, pp. 3650–3662, April 2020.
- [18] R. Faragher and R. Harle, "Location fingerprinting with bluetooth low energy beacons," *IEEE Journal on Selected Areas in Communications*, vol. 33, no. 11, pp. 2418–2428, Nov 2015.
- [19] W. He, P. Ho, and J. Tapolcai, "Beacon deployment for unambiguous positioning," *IEEE Internet of Things Journal*, vol. 4, no. 5, pp. 1370–1379, Oct 2017.
- [20] E. Vergetis, R. Guerin, S. Sarkar, and J. Rank, "Can Bluetooth succeed as a Large-scale Ad Hoc Networking Technology?" *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 3, pp. 644–656, March 2005.
- [21] M. Chernyshev, C. Valli, and M. Johnstone, "Revisiting Urban War Nibbling: Mobile Passive Discovery of Classic Bluetooth Devices Using Ubertooth One," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 7, pp. 1625–1636, July 2017.
- [22] G. Shan and B.-h. Roh, "Performance model for advanced neighbor discovery process in bluetooth low energy 5.0-enabled internet of things networks," *IEEE Transactions on Industrial Electronics*, vol. 67, no. 12, pp. 10965–10974, Dec 2020.
- [23] G. Shan and B.-H. Roh, "Advertisement interval to minimize discovery time of whole ble advertisers," *IEEE Access*, vol. 6, pp. 17817–17825, 2018.
- [24] L. Leonardi, G. Patti, and L. L. Bello, "Multi-hop real-time communications over bluetooth low energy industrial wireless mesh networks," *IEEE Access*, vol. 6, pp. 26 505–26 519, 2018.
- [25] M. Baert, J. Rossey, A. Shahid, and J. Hoebeke, "The bluetooth mesh standard: An overview and experimental evaluation," *Sensors*, vol. 18, no. 8, 2018.
- [26] H. Kim, J. Lee, and J. W. Jang, "Blemesh: A wireless mesh network protocol for bluetooth low energy devices," in *2015 3rd International Conference on Future Internet of Things and Cloud*, Aug 2015, pp. 558–563.
- [27] P. C. Ng, J. She, and R. Rong, "Compressive rf fingerprint acquisition and broadcasting for dense ble networks," *IEEE Transactions on Mobile Computing*, pp. 1–1, 2020.
- [28] Y. K. Reddy, P. Juturu, H. P. Gupta, P. R. Serikar, S. Sirur, S. Barak, and B. Kim, "A connection oriented mesh network for mobile devices using bluetooth low energy," in *Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems*. ACM, 2015, pp. 453–454.
- [29] Z. Guo, I. G. Harris, L. Tsaur, and X. Chen, "An on-demand scatternet formation and multi-hop routing protocol for ble-based wireless sensor networks," in *2015 IEEE Wireless Communications and Networking Conference (WCNC)*, March 2015, pp. 1590–1595.
- [30] Y. Murillo, B. Reynders, A. Chiumento, S. Malik, P. Crombez, and S. Pollin, "Bluetooth now or low energy: Should ble mesh become a flooding or connection oriented network?" in *2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, Oct 2017, pp. 1–6.
- [31] J. DeCuir, "Introducing bluetooth smart: Part 1: A look at both classic and new technologies," *IEEE Consumer Electronics Magazine*, vol. 3, no. 1, pp. 12–18, Jan 2014.
- [32] M. Collotta, G. Pau, T. Talty, and O. K. Tonguz, "Bluetooth 5: A concrete step forward toward the iot," *IEEE Communications Magazine*, vol. 56, no. 7, pp. 125–131, JULY 2018.
- [33] L. Wang, P. Wan, and B. Banks, "Constant approximations for beaconing scheduling in wireless networks with duty-cycled scenarios," *IEEE Transactions on Wireless Communications*, vol. 15, no. 3, pp. 2328–2334, March 2016.
- [34] F. Wang and J. Liu, "On reliable broadcast in low duty-cycle wireless sensor networks," *IEEE Transactions on Mobile Computing*, vol. 11, no. 5, pp. 767–779, May 2012.
- [35] R. Schrader, T. Ax, C. Röhrig, and C. Fühner, "Advertising power consumption of bluetooth low energy systems," in *2016 3rd International Symposium on Wireless Systems within the Conferences on Intelligent Data Acquisition and Advanced Computing Systems (IDAACS-SWS)*, Sep. 2016, pp. 62–68.
- [36] P. H. Kindt, M. Saur, M. Balszun, and S. Chakraborty, "Neighbor discovery latency in ble-like protocols," *IEEE Transactions on Mobile Computing*, vol. 17, no. 3, pp. 617–631, March 2018.
- [37] S. Kamath and J. Lindh, "Measuring bluetooth low energy power consumption," *Texas instruments application note AN092*, Dallas, 2010.



Pai Chet Ng obtained her Ph.D. degree in electronic and computer engineering from the Hong Kong University of Science and Technology. She is currently a Post-doctoral Researcher with the University of Toronto. Her research interests include Proximity-based Sensing and Networking, Physiological Signals with Mobile and Wearable Devices, and IoT Systems with Artificial Intelligence.



James She is an assistant professor in the Department of Electronic and Computer Engineering at the Hong Kong University of Science and Technology (HKUST), and a visiting research fellow at the University of Cambridge. He is also the founding director of Asia's first social media lab, HKUST-NIE Social Media Lab, and spearheads multidisciplinary research and innovation in analytics and systems for social media and multimedia big data, smart/wearable device based interactive technologies in cyber-physical system or IoT infrastructure, and new media technologies and productions for art and culture. Since 2016, James became an associate editor for ACM Transaction on Multimedia Computing, Communications and Applications.



Petros Spachos received the Diploma in electronic and computer engineering degree from the Technical University of Crete, Chania, Greece, the M.A.Sc. and Ph.D. degrees in electrical and computer engineering from the University of Toronto, Canada. He was a Postdoctoral Researcher with the University of Toronto from September 2014 to July 2015. He is currently an Assistant Professor with the School of Engineering, University of Guelph, Guelph, ON, Canada. His research interests include experimental wireless networking and mobile computing with a focus on wireless sensor networks, smart cities, and the Internet of Things.