2019 IEEE Wireless Communications and Networking Conference (WCNC)

A Novel Overlay Mesh with Bluetooth Low Energy Network

Pai Chet Ng and James She
Department of Electronic and Computer Engineering
The Hong Kong University of Science and Technology
Email: {pcng, eejames}@ust.hk

***Abstract*—While Bluetooth Low Energy (BLE) beacons have been massively deployed to broadcast their advertising packets to any receivers in their vicinity, it is relatively difficult, if not impossible, for a beacon to report the packet back to the server in the absence of a receiver. This paper proposes a novel BLE-based overlay mesh (BOM) that enables the mesh functionality to existing beacon networks without introducing new infrastructure. However, it is an elusive challenge to jointly manage the beaconing and flooding events. To this end, BOM employs 1) best-effort scheduling (BES) to minimize the packet collision rate (PCR) while scheduling the time slots for beaconing events, and 2) received signal strength (RSS)-based bounded flooding (RBF) to maximize the packet delivery ratio (PDR) for the *advertising packet* while forwarding the *relaying packet* across the BOM network. Extensive simulations indicate the substantial performance gain of our proposed approach in comparison to the legacy approaches. Specifically, BES reduces the PCR to 66.67%, whereas RBF improves the PDR for the *advertising packet* to 52% while maintaining approximately the same PDR for the *relaying packet*. The practical experiment with a real network testbed further demonstrates the feasibility of BOM.**

I. Introduction

Bluetooth Low Energy (BLE) beacon is an active broadcaster that disseminates its *advertising packet* to the surroundings periodically. Any Bluetooth-compatible receiver (e.g., smartphone, tablet, etc.) can pick up the *advertising packet* when it is in the proximity of the beacon. To date, BLE beacons have been widely deployed in many locations, including airports and shopping malls, to promote proximity marketing [1] [2] and provide localization services [3] [4]. Besides the typical *advertising packet*, BLE beacon can also broadcast a packet that contains useful sensing information for remote monitoring. However, it is almost impossible to send that packet back to the server when there is no receiver in the vicinity of the beacon.

The above-mentioned problem is illustrated in Fig. 1. Let $G = (N, L)$ denote the beacon network, where set $N = \{1, 2, ..., n\}$ defines the list of nodes and set $L = \{l_{uv} | 1 \leq v \neq u \leq n\}$ describes the relationship between each node in N. Suppose that node u would like to send a specific packet $x_u(t)$ back to the server, the packet would be left unattended in the air since there is no receiver within the broadcasting range of node u (i.e., r_u), as shown in Fig. 1(a). Fig. 1(b) shows that this problem can be resolved if each node can help relay the packet until the packet arrives at a receiver. It is almost impossible, however, for the connectionless beacon to relay the packet unless extra infrastructure is added to the existing network. For example, a few Bluetooth mesh nodes (i.e., the BLE devices working with the new mesh protocol introduced by Bluetooth SIG in July 2017) can be added between each beacon node to help to relay the packet. Nonetheless, such a solution is costly and inefficient given the massive number of deployed beacons.

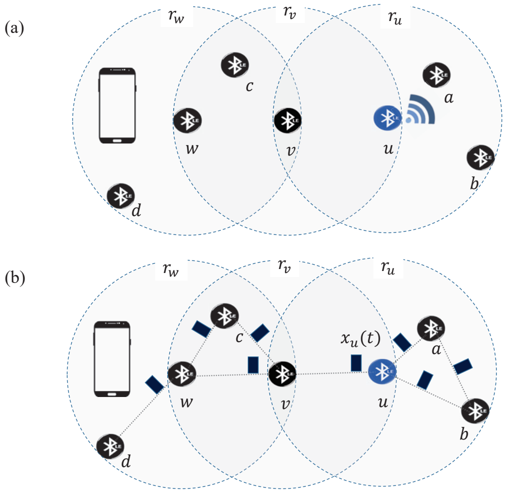


Fig. 1. (a) The packet broadcast by node u would be left unattended in the air since there is no receiver within r_u. (b) Such a problem could be resolved if there is a way to relay the packet from node u to node w.

This paper presents a novel BLE-based overlay mesh (BOM) that enables the mesh functionality to existing beacon networks by upgrading the firmware of each node without introducing new infrastructure. Hence, each node can still serve their proximity marketing or localization service via the periodic beaconing event, and are only required to participate in the flooding event when there is a relaying request. Flooding approach always assumes that all nodes are idle when a packet arrives. Such an assumption could not be applied to our BOM network considering the periodic beaconing event at every node. This paper unveils 2 research challenges toward the realization of BOM by jointly considering the periodic beaconing and intermittent flooding events.

Two types of packet are defined: *advertising* and *relaying packet*. The *advertising packet* refers to the general packet for the periodic beaconing event, whereas the *relaying packet* refers to the packet with the relaying request. To this end, we

978-1-5386-7646-2/19/$31.00 ©2019 IEEE

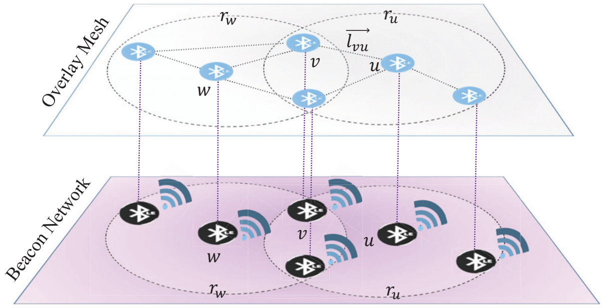


Fig. 2. BOM enables the mesh functionality to existing beacon networks without adding new infrastructure.

propose 1) best-effort scheduling (BES) algorithm to schedule the time slot for beaconing event, and 2) received signal strength (RSS)-based bounded flooding (RBF) algorithm to selectively determine which nodes should participate in the flooding event. To the best of our knowledge, our proposed BOM is the very first work to realize the beaconing and flooding events over the same network without adding new infrastructure. The main contributions of this paper are summarized as follows:

- A generic BOM protocol data unit (PDU) is defined to distinguish between the *advertising* and *relaying packet*.
- BES reduces the packet collision rate (PCR) by strategically scheduling the time slot for each beaconing event;
- RBF improves the packet delivery rate (PDR) for *advertising packets* by minimizing excessive relaying redundancy during the flooding event;
- Practical experiments with a real protoype demonstrates the backward compatibility of our proposed BOM with all versions of BLE.

The rest of the paper is organized as follows. Section II presents our novel BOM network model. Section III describes the algorithms applied by our proposed BOM. Section IV discusses the simulation results. Section V demonstrates our BOM with the real network testbed. Section VI concludes the paper with future works.

II. BLE-BASED OVERLAY MESH (BOM)

Before presenting our BOM network model, we first highlight recent works related to beacon and mesh network. Currently, there are 4 variants of BLE, namely Bluetooth v4.0, v4.1, v4.2 and v5.0 [5]. Beacon networks can be constructed with any version of Bluetooth mentioned above. Each node in the beacon network broadcasts their *advertising packet* periodically according to the pre-defined advertising interval T_a [6]. The official mesh network has been introduced by the Bluetooth SIG in July 2017 [7]. There have been active research works on mesh network prior to the introduction of the official mesh network. For example, BLE Mesh Network (BMN) [8] employs Directed Acyclic Graph to construct the routing table; whereas [9] eliminates the used of a routing table with an on-demand routing approach. However, both beacon and mesh networks are using different transmission protocol. While there are a lot of works addressing the challenges for either beacon and mesh networks, so far, there is no work jointly address both challenges over the same network. In other words, there is no single network which can function as both beacon and mesh network at the same time.

Our BOM network is a novel solution that superimpose the mesh functionality on top of the beacon network. The network architecture of our proposed BOM is illustrated in Fig. 2. There are 2 main operation routines with the BOM network, i.e., initialization and operation routines. The beaconing time slot is scheduled during the initialization routine, and each node simply repeats the beaconing event during the operation routine. Let $S = (u, t_u)$ be a 2-tuple, a node u which successfully initialize its beaconing event in the time slot t_u would be added to S. There are 2 *transmission events* during the operation routine:

*1) **Beaconing Event:*** Based on S, each node in the network repeats their beaconing event to broadcast the *advertising packet* $a_u(t)$ every multiple of T_a. More precisely, $a_u(t)$ is scheduled in the time slot $t = t_u + kT_a$, for all $u \in N$. The beaconing is considered unsuccessful when 2 or more nodes schedule their beaconing event in the same time slot, resulting in packet collision. Let T_a be fixed for all n nodes in the network, node u can broadcast $a_u(t)$ at time t successfully when the following condition is satisfied:

$$a_u(t) + \sum_{\forall v \in NB(u)} a_v(t) = 1, \quad \forall u \in N \tag{1}$$

where

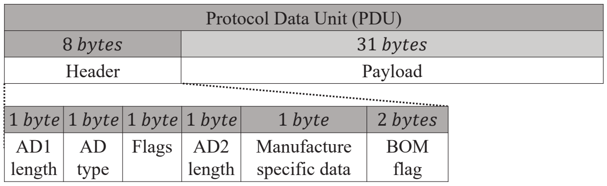
$$a_u(t) = \begin{cases} 1, & mod(t - t_u - T_t - T_{\Delta_{min}}, T_a) = 0 \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

where T_t is the transmission time, which can be computed by dividing the packet length $\mathcal{L}(a_u)$ by the data rate $\mathcal{R}$. For example, $T_t = \frac{\mathcal{L}(a_u)}{\mathcal{R}} = \frac{47 \times 8}{1 \times 10^6} = 0.376ms$ given a typical *advertising packet* with length of $47bytes$ and BLE v4.0-4.1 with $\mathcal{R} = 1Mbps$ $T_{\Delta_{min}}$ is the minimum time spacing between each subsequent packets. In the next section, we present our BES which is designed to schedule the beaconing event such that to minimize the possible collision.

*2) **Flooding Event:*** Each node in the same neighborhood can participate in the flooding event when there is a relaying request. Node v and w are considered to be in the same neighborhood to u when both v and w are located within r_u. Let NB be the set of neighboring nodes, we have:

$$NB(u) = \{v | d(u, v) \leq r_u, \forall v \in N\} \tag{3}$$

where $d(u, v)$ is a function computing the distance between u and v. It is impossible for any node in the existing beacon network to participate in the flooding event since none of these nodes are capable of listening to the incoming packet. The following subsection presents our BOM PDU, a key-enabler towards the realization of BOM.



Protocol Data Unit (PDU)	
$8\,bytes$	$31\,bytes$
Header	Payload

$1\,byte$	$1\,byte$	$1\,byte$	$1\,byte$	$1\,byte$	$2\,bytes$
AD1 length	AD type	Flags	AD2 length	Manufacture specific data	BOM flag

Fig. 3. Our BOM PDU allocates $31bytes$ for the payload, which can be customized according to iBeacon, Eddystone or any sensor data format.

A. Protocol Data Unit defined by BOM

BOM defines a PDU that enables the BLE device to function as the broadcaster and scanner concurrently such that the device retains its function similar to a conventional beacon and at the same time is able to listen to the incoming packets. Our BOM PDU is based on the standard BLE stack, rather than introducing a new stack and concepts, as compared to the official Bluetooth Mesh 1.0 [10]. Fig. 3 depicts the PDU of BOM. According to BLE standard, the maximum packet size is $47bytes$, in which the first $5bytes$ are reserved for the preamble and MAC address. For the remaining $39bytes$, our BOM PDU uses the first $8bytes$ to define the header; in particular, the 6th $byte$ and 7th $byte$ are used to indicate the BOM flag. For example, $byte(6,7) = (42H, 50H)$ denotes an *advertising packet*, whereas $byte(6,7) = (53H, 50H)$ denotes a *relaying packet*. The remaining $31bytes$ from the available data payload can be customized using either the iBeacon, Eddystone, or any sensor data format.

III. PROPOSED APPROACHES

In this section, we present our BES for scheduling the time slot for beaconing event and RBF for selecting the node for participating in the flooding event.

A. Best-effort Scheduling (BES)

The typical approach employed by the existing beacon networks is random access scheduling (RAS) [11], in which the beaconing time slot is initialized randomly as soon as the nodes are added to the network. While RAS is proven to be efficient, the likelihood of packet collision increases when the number of nodes in the same neighborhood increases. Eq. (1) clearly states that collision happens when two or more nodes in the same neighborhood trigger the beaconing event at the same time. Let $NC(u) \subset NB(u)$ be a set of non-collided nodes, the maximum number of nodes that can be placed around u without causing packet collision is

$$|NC(u)| \leq \frac{T_a}{\mathcal{L}(a_u) + T_{\Delta_{min}}} \mathcal{R} - 1 \tag{4}$$

where $|NC(u)| \leq m = |NB(u)|$. Intuitively, if each node within the same neighborhood can ideally divide their beaconing time slots within the fixed interval $[t, t+T_a)$, then at most $|NC(u)| + 1 \leq m + 1$ nodes inside $r_u, \forall u \in N$ are capable of broadcasting their packet successfully.

Instead of randomly initializing the time slot, BES strategically schedules the time slot for each beaconing event within the interval defined by T_a. When the nodes are added to a network, BES assigns a random probability $P(u)$ to all the nodes $u \in N$, then, each node will try to initialize their beaconing event according to $P(u)$. As soon as a node successfully initializes its beaconing event, all the other nodes will keep quiet for $T_t + T_\Delta$ before attempting to initialize their beaconing event again. Such a quiet time is required to avoid interfering with the previously broadcasted packet. A node u that successfully initializes its beaconing event in time slot t_u would be added to S. The collided nodes would need to re-randomize their probability whenever there is a collision. According to the condition defined by Eq. (1), the set of collide nodes is obviously

$$C = \{u, v | \sum_{\forall v \in NB(u)} a_v(t) \geq 1\} \tag{5}$$

BES will always try its best to schedule the beaconing time slot for each node. In case there are still some empty time slots, BES will try to reschedule for those nodes in C, if $|C| \neq 0$. Let $\bar{n}$ be the number of nodes that are not yet initialized, and T be a set consists of a list of empty time slots, then BES can be described with Algorithm 1.

Algorithm 1: Best-effort Scheduling (BES)

```
Input  : $P(u), \forall u \in N$
Output: A set $S$ of tuples $\{(u, t_u) | u \in N\}$
while $t \leq T_a$ do
    if Eq. (1) then
        $S \leftarrow \{(u, t_u)\}$
        $\bar{n} = |N| - |S|$
        $t = t + T_t + T_\Delta$
    else
        $P(v) = rand(0,1), \forall v \in C$
        $t = t + T_\Delta$
    end
end
if $\bar{n} > 0$ then
    $T = \{\bar{t} | \bar{t} = t_v - t_u - T_\Delta > T_t, \forall u, v \in S\}$
    $S \leftarrow \{(v, T) | \forall v \in N \setminus S\}$
end
Beaconing: $t_u = t_u + kT_a, \forall (u, t_u) \in S$
```

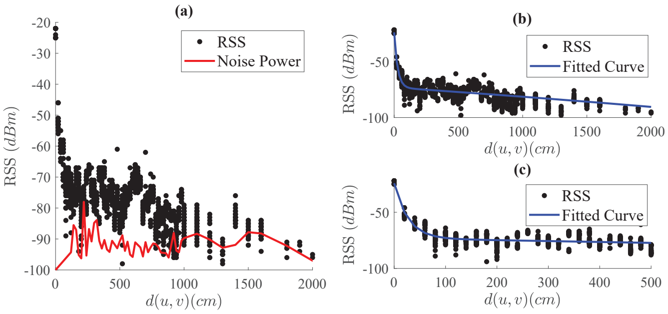


Fig. 4. (a) The RSS values are above the noise floor when $d \leq 10m$. (b) Least square fitting is applied to obtain the fitted curve. (c) The view is zoomed in to $d = [0, 5]m$ to observe the fluctuation of RSS values when $d > 1m$.

Algorithm 2: RSS-based Bounded Flooding (RBF)

```
Input  : Source packet: $x_s(t)$
Output: $x_{r_k}(t_{r_i}), \forall (r_i, t_{r_i}) \in B$
Obtain set $\tilde{Z}$ by Eq. (7)
Assign $P(\tilde{z}_i|r_k) = \frac{RSS_{z_i|r_k}}{\alpha e^{\kappa_1} + \beta e^{\kappa_2}}$
$t_w \leftarrow rand(t_{w,min}, t_{w,min}(1 - P(z_i|z_{k-1})))$
while $t_w$ do
    if $x_{z_j}(t)$ equal $x_s, \forall j \neq k$ then
        $TTL(x_s) = TTL(x_s) - 1$
        if $TTL(x_s)$ equal $0$ then
            Discard: $x_s$
            break
        end
    end
    if $t_w$ equal $0$ then
        Overwrite: $a_{z_i}(t_{z_i}) \leftarrow x_s$
    end
end
```

B. RSS-based Bounded Flooding (RBF)

A node which broadcast a *relaying packet* $x_u(t)$ is denoted as source node. Let Z be a set of nodes that receives the packet, obviously $Z = NB(u)$. The legacy probabilistic flooding (PF) [12], in general, assigns probability $P(z_i)$ for each node $z_i \in Z$ such that each node will wait according to $P(z_i)$ before attempting to retransmit the packet.

This paper proposes RBF which enables each node to compute their relay probability according to the measured RSS. However, RSS is unreliable when $d > 1m$. According to Bluetooth standard, a typical BLE node with transmit power $P_T = 0dBm$ can have a transmission range of up to $10m$. To better observe the relationship between the distance between 2 nodes $d(u, v)$ and RSS, we conducted an experiment by varying $d(u, v)$ from $0cm$ to $2000cm$. The experimental results shown in Fig. 4(a) clearly indicate that RSS values are high (i.e., $\approx (-30, -20)dBm$) when $d(u, v) \leq 10$, and start to decrease when $d(u, v)$ increases. Fig. 4(b) and (c) illustrates the fitted curve obtained through the least square fitting. The resulting fitted curve can be described as follows:

$$RSS(d) = \alpha e^{\kappa_1 d(u,v)} + \beta e^{\kappa_2 d(u,v)} \tag{6}$$

where α, β, κ_1 and κ_2 are the fitting coefficients. From the zooms in view depicted in Fig. 4(c), we can observed that RSS decreases linearly with d when $d < 1m$, and the values start to fluctuate when $d \geq 1m$. Based on such observations, we argue that the RSS values are unreliable when $d > 1m$.

In light with the above observation, our proposed RBF only considers the nodes that are located within the $1m$ distance from the source node. Accordingly, we refine set $\tilde{Z}$ to filter out the elements that do not meet bounded condition θ, i.e.,

$$\tilde{Z} = \{z_i | \theta = \frac{RSS_{z_i|r_k}}{\alpha e^{\kappa_1} + \beta e^{\kappa_2}} > 1, \forall z_i \in Z\} \tag{7}$$

where the denominator is obtained by setting the $d(u, v)$ in Eq. (6) to 1, $RSS_{z_i|r_k}$ indicates the RSS measured by z_i, and $r_k \in R$ denotes the node that has successfully relayed the packet in the previous k hops.

Each node in $\tilde{Z}$ then attempts to relay the packet according to the relay probability $P(\tilde{z}_i|r_k)$. To minimize the packet redundancy, RBF adopts the same time-to-live (TTL) counter used by the official mesh model to manage the flooding. Moreover, RBF allows the nodes in $\tilde{Z}$ to withdraw from the relaying attempt when they overhear the same packet has been transmitted successfully by one of their neighboring nodes. Otherwise, each node will wait for a random time t_w before attempting to relay the packet. Algorithm 2 describes the overall flow of RBF.

IV. SIMULATIONS AND EVALUATIONS

Our simulations can be separated into 2 parts according to the following routines: 1) the initialization, and 2) the operation routine.

A. The Initialization Routine

The nodes are generated randomly within the confined space $100cm \times 100cm$. Each node is configured to behave like a beacon operated according to the specification defined by Bluetooth v4.0/4.1, and have the same advertising interval and transmit power, i.e., $T_a = 100ms$ and $P_T = 0dBm$. The minimum transmission duration is governed by T_t and $T_{\Delta_{min}}$.

The number of neighboring nodes is varied from 10 to 1500, with 10 increments each step. The results achieved by RAS and BES for $T_a = 100, 500$ and $1000ms$ are shown in Fig. 5(a), (b) and (c), respectively. It is clear that our proposed BES suffers less collision compared to RAS. The PCR increases when m increases; whereas PCR decreases when T_a increases from $100ms$ to $1000ms$. In other words, more nodes can be squeezed into the same neighborhood when T_a increases. Fig. 5(c) shows that our proposed BES is approximating to the upper bound defined by Eq. (4), which further indicates the superiority of our proposed BES in comparison to RAS.

B. The Operation Routine

Following the beaconing schedule obtained from the previous simulation, the second simulation randomly selected a few nodes to act as a source node to generate the *relaying packet*. We implemented both our proposed RBF and the legacy PF approach. The performance of both approaches is evaluated based on the PDR, which computes the number of packets successfully delivered in $1s$. *Note that the PDR computation for the advertising packet and the relaying packet* is different. The PDR for the *advertising packet* is computed by taking the ratio of the number of successfully broadcasted packets over the expected number of packets to be broadcast in $1s$, i.e.,

$$\begin{aligned} PDR(a) &= \frac{\mathcal{N}(a_u, t = 1s)}{\mathbb{E}(\mathcal{N}(a_u, t = 1s))} \\ &= \mathcal{N}(a_u, t = 1s)T_a, \quad \forall u \in N \end{aligned} \tag{8}$$

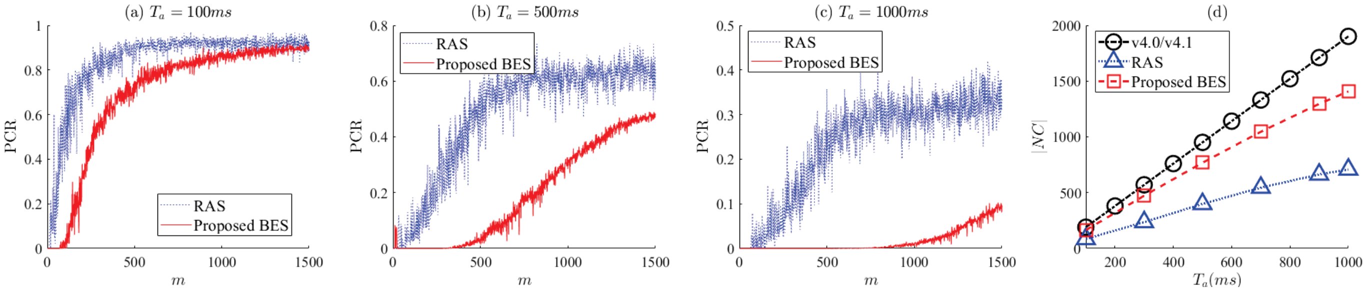
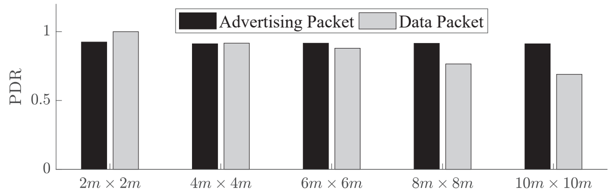


Fig. 5. PCR(s) achieved by our proposed BES vs the legacy RAS for (a) $T_a = 100ms$, (b) $T_a = 500ms$ and (c) $T_a = 1000ms$, given $m = 10 : 10 : 1500$. (d) The maximum number of m achieved by BES and RAS for $Ta = [100, 1000]ms$ in comparison to the upper bound described by Eq. (4).


(a) Our Proposed RBF

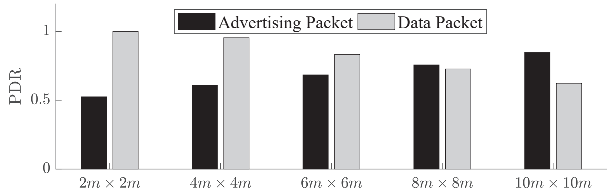

(b) The Legacy PF

Fig. 6. Comparison between the PDR for the *advertising packet* and the *relaying packet* achieved by (a) our proposed RBF and (b) the legacy PF.

where $\mathcal{N}(a_u)$ denotes the number of *advertising packets* broadcast by node u in $1s$. Note that $\mathbb{E}(\mathcal{N}(a_u, t = 1s))$ is dependent on the number of packets that depart from node u, which is equivalent to the packet departure rate $\mathbb{E}(\mathcal{N}(a_u, t = 1s)) = \mu_u = \frac{1}{T_a}$, assuming all the nodes have the same T_a. On the other hand, the PDR for the *relaying packet* is computed by taking the ratio of the number of nodes which successfully received the *relaying packet* in $1s$ to the total number of nodes $n = |N|$, given only 1 *relaying packet* is transmitted. Without loss of generality, the PDR for the *relaying packet* when more than one *relaying packet* is transmitted can be described as follows:

$$PDR(x) = \frac{\mathcal{N}(u_{x_s}, t = 1s)}{n\mathcal{N}(x_s, t = 1s)} \tag{9}$$

where $\mathcal{N}(u_{x_s}, t = 1s)$ denotes the number of nodes that have already received the *relaying packet* x_s. Such information can be examined by checking the packet cache of each node.

The results achieved by RBF and PF are shown in Fig. 6(a) and (b), respectively. We varied the size of the space from $200cm \times 200cm$ to $10m \times 10m$, and the number of nodes tabulated for each space were $[12, 48, 108, 192, 300]$. From the bar charts, it is obvious that the legacy PF trades off the PDR for the *advertising packet* in order to forward the *relaying packet* across the network. Meanwhile, our proposed RBF is able to balance the PDR for both *advertising* and *relaying packets*. Most importantly, our proposed RBF is able to ensure the beaconing event of each node while dealing with the intermittent relaying request.

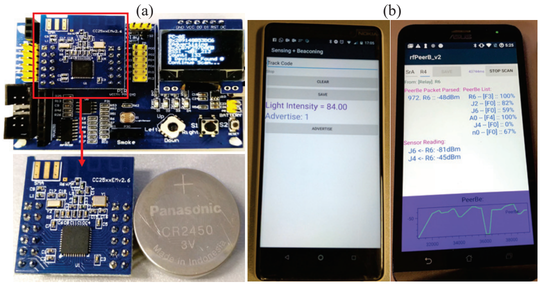


Fig. 7. (a) An RF smartboard is used during the development and debugging phase. The final prototype is a small BLE device that can be powered up with a coin-cell battery. (b) 2 Android Apps were developed: (1) the "*sensing* + *beaconing*" App, and (2) the "*receiving*" App.

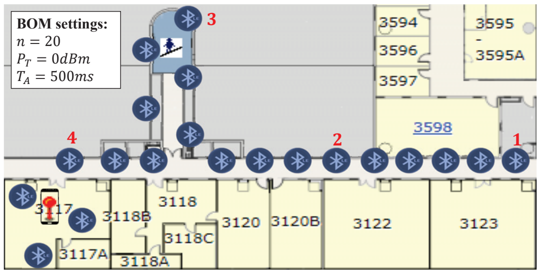

BOM settings:
$n = 20$
$P_T = 0dBm$
$T_A = 500ms$

Fig. 8. The BOM network consists of 20 nodes. The intermittent *relaying packets* are generated randomly at the 4 locations indicated by the red digits.

V. PROTYPING AND REAL NETWORK IMPLEMENTATION

A prototype is built to demonstrate the feasibility of our proposed BOM in a practical environment. Our prototype is based on the *CC2541* chipset, which is a BLE chipset provided by *Texas Instrument*. Fig. 7(a) shows an RF smartboard from *Texas Instrument* that we used during the development and debugging phase. The final prototype is a small device with a size similar to a coin cell battery. Upon validating the functionalities of the prototype, we packaged our code and implemented it on over 25 BLE devices.

A. Real Network Testbed

Out of the 25 devices, 20 were used to construct the BOM network. The remaining 5 were kept as a backup. Fig. 8 depicts our BOM testbed consisting of a total of 20 nodes, with each

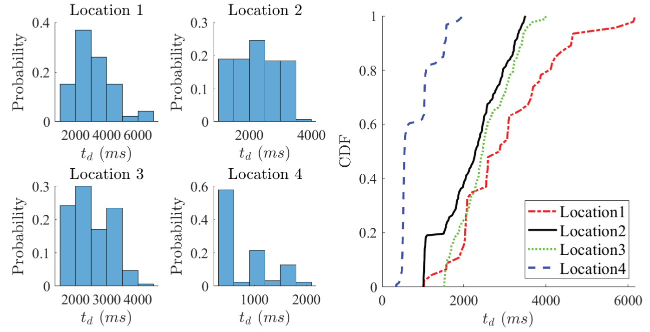


Fig. 9. The time duration for the *relaying packet* to travel from the location indicated by Fig. 8 to the receiver residing in the laboratory.

node configured to have $P_T = 0dBm$ and $T_a = 500ms$. The testbed was set up along the corridor outside the laboratory and extended all the way out to the outdoor staircase. The server is located inside the laboratory.

As shown in Fig. 7(b), 2 Android Apps were developed to facilitate the experiment. The "*sensing + beaconing*" App allows us to measure the sensor information (in this experiment, we opt to measure the light intensity), and then trigger the beaconing function by clicking the "*Advertise*" button. Using this App, we can manually control when to send out a *relaying packet*. We placed the smartphone at the 4 locations indicated in Fig. 8, and then randomly generate the *relaying packet*. Each successful generated packet was logged as a .csv file and saved inside the smartphone.

Another App - the "*receiving*" App enables the smartphone to function as the receiver. The smartphone was placed inside the laboratory, as shown in Fig. 8. Upon receiving the packet, the smartphone can distinguish the types of the packet (i.e., *advertising packet* or *relaying packet*) by inspecting the BOM flag. If it is a *relaying packet*, the smartphone will send the packet to the server, otherwise it will just retrieve the related service associated with the *advertising packet*. Similarly, all the information regarding the received packet is logged as a .csv file. By comparing the logged file from the "*sensing + beaconing*" App with the logged file saved by the "*receiving*" App, the end-to-end delay t_d can be computed.

B. Experimental Results

About 500 *relaying packets* were generated at each location. The time required by the packets to travel from the corresponding locations to the receiver is shown in Fig. 9.

Since the packets generated at location 4 generally required 1 hop to arrive at the receiver, we have $t_d \leq 500ms$. Sometimes, the packet also arrived by hopping through another 2 nodes nearby the receiver. Packets generated at location 2 and 3 required approximately the same t_d to arrive at the receiver, whereas the packets generated at location 1 required longer t_d since multiple hops were required before the packet could arrive at the receiver. It was observed that every node could still constantly broadcast their *advertising packet* at a rate of $\leq 2packets/s$. Hence, BOM is proven to be a feasible solution, which can forward the *relaying packet* on top of the existing beacon network, without significantly affecting the normal functionality of the beacon.

VI. CONCLUSIONS

This paper presents a novel BOM that enables the mesh functionality to existing beacon networks without adding extra infrastructure. Furthermore, we jointly address the beaconing and flooding problem with BES and RPF. Extensive simulations indicate the substantial performance gain of our proposed BBF in comparison to the legacy approaches, whereas practical experiments with a real network testbed demonstrates the feasibility of our proposed BOM. For future work, advanced coding methods (e.g., sparse coding) can be applied to enable the node to superimpose both *advertising* and *relaying packet* in the same length-constrained packet. Furthermore, BOM presents many research opportunities. One of the challenges would be on how to select the node to help to forward the relaying packet without affecting the normal beaconing event. To conclude, enabling the mesh functionality on top of the existing beacon network is a novel and cost-efficient solution for many IoT applications.

ACKNOWLEDGMENT

This work was supported by HKUST-NIE Social Media Laboratory.

REFERENCES

[1] T. Ojala, F. Kruger, V. Kostakos, and V. Valkama, "Two field trials on the efficiency of unsolicited bluetooth proximity marketing," in *Proceedings of the 11th International Conference on Mobile and Ubiquitous Multimedia*, ser. MUM '12, 2012, pp. 38:1–38:4.

[2] "Enhancing the shopping experience," *IEEE Pervasive Computing*, vol. 10, no. 2, pp. 44–47, April 2011.

[3] 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.

[4] L. Xiang, T. Y. Tai, B. Li, and B. Li, "$tack$: learning towards contextual and ephemeral indoor localization with crowdsourcing," *IEEE Journal on Selected Areas in Communications*, vol. 35, no. 4, pp. 863–879, April 2017.

[5] 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.

[6] 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.

[7] 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.

[8] 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.

[9] 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.

[10] M. Baert, J. Rossey, A. Shahid, and J. Hoebeke, "The bluetooth mesh standard: An overview and experimental evaluation," *Sensors*, vol. 18, no. 8, 2018.

[11] C. Joo and N. B. Shroff, "Performance of random access scheduling schemes in multi-hop wireless networks," *IEEE/ACM Transactions on Networking*, vol. 17, no. 5, pp. 1481–1493, Oct 2009.

[12] R. Gaeta and M. Sereno, "Generalized probabilistic flooding in unstructured peer-to-peer networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 22, no. 12, pp. 2055–2062, Dec 2011.