Optimizing Packet Reception Rates for Low Duty-Cycle BLE Relay Nodes

Nuno Paulino, Luís M. Pessoa, André Branquinho, Rafael Almeida, and Igor Ferreira

***Abstract*—In order to achieve the full potential of the Internet-of-Things, connectivity between devices should be ubiquitous and efficient. Wireless mesh networks are a critical component to achieve this ubiquitous connectivity for a wide range of services, and are composed of terminal devices (i.e., nodes), such as sensors of various types, and wall powered gateway devices, which provide further internet connectivity (e.g., via Wi-Fi). When considering large indoor areas, such as hospitals or industrial scenarios, the mesh must cover a large area, which introduces concerns regarding range and the number of gateways needed and respective wall cabling infrastructure, including data and power. Solutions for mesh networks implemented over different wireless protocols exist, like the recent *Bluetooth Low Energy* (BLE) 5.1. While BLE provides lower power consumption, some wall-power infrastructure may still be required. Alternatively, if some nodes are battery powered, concerns such as lifetime and packet delivery are introduced. We evaluate a scenario where the intermediate nodes of the mesh are battery powered, using a BLE relay of our own design, which acts as a range extender by forwarding packets from end-nodes to gateways. We present the relay's design and experimentally determine the packet forwarding efficiency for several scenarios and configurations. In the best case, up to 35% of the packets transmitted by 11 end-nodes can be forwarded to a gateway by a single relay under continuous operation. A battery lifetime of 1 year can be achieved with a relay duty cycle of 20%.**

***Index Terms*— BLE, Bluetooth, low-energy, wireless sensor networks, mesh networks.**

I. Introduction

WIRELESS mesh networks can be the platform for many applications. A common use case are sensor networks [1], but others include domotics [2], automated inventory tracking or localization [3]. Specific scenarios include healthcare [4], [5], security [6], [7] and warehouses and industrial facilities [8].

Depending on the application, mesh networks can be built with Wi-Fi devices, for example, but Wi-Fi end-points or routers are more costly, and typically require wall power. On the other hand, BLE devices benefit from a comparatively lower cost, power efficiency, and smaller device sizes. The specification for Bluetooth Mesh networking was introduced in 2017 [9], and is meant to operate over implementations of BLE, which itself was integrated into the Bluetooth specification as of version 4 [10].

A controlled flood routing mechanism allows for global connectivity among all nodes for applications where only advertising channels are used, avoiding the need to establish specific channels. Although this simplifies the network setup and adds some resilience to node failure, it causes not only congestion due to collisions, but also increases energy cost due to the redundant transmissions. Also, nodes that are battery powered may power down periodically to save energy, in which case packet transit may be compromised. To support this, some nodes act as friends to these low-power nodes. Friends will buffer packets meant for the low-power nodes, until these become active. This however implies that the friend nodes must be constantly powered, otherwise the packet delivery would again be compromised. In other works, the majority of the network, i.e., the intermediate nodes acting as relays, must either be equipped with long lasting batteries, or wall-power must be readily available. This is not only costly, but may make the deployment of BLE mesh solutions unviable in legacy locations without this prior infrastructure.

Manuscript received April 21, 2022; accepted May 18, 2022. Date of publication June 7, 2022; date of current version July 1, 2022. This work was supported by the ERDF—European Regional Development Fund, through the Operational Program for Competitiveness and Internationalization—COMPETE 2020 Program under the Portugal 2020 Partnership Agreement, within Project Continental AA"s Factory of the Future (Continental FoF), under Grant POCI-01-0247-FEDER-047512. The associate editor coordinating the review of this article and approving it for publication was Prof. Pierluigi Salvo Rossi. *(Corresponding author: Nuno Paulino.)*

Nuno Paulino and Luís M. Pessoa are with the Institute for Systems and Computer Engineering, Technology and Science (INESC TEC), 4200-465 Porto, Portugal (e-mail: nuno.m.paulino@inesctec.pt; luis.m.pessoa@inesctec.pt).

André Branquinho, Rafael Almeida, and Igor Ferreira are with Wavecom S.A., 3800-125 Aveiro, Portugal (e-mail: abranquinho@wavecom.pt; rtavares@wavecom.pt; iferreira@wavecom.pt).

Digital Object Identifier 10.1109/JSEN.2022.3179622

Therefore, despite the relative maturity of the BLE Mesh specification, and its suitability for the *Internet-of-Things* (IoT), the current specification is not without drawbacks [11]. Some works in the state-of-the-art have identified some limitations or potential improvements which are mostly driven by specifics of the use-case [11]–[13]. In fact, the Bluetooth *Special Interest Group* (SIG) defines operational models at the application-layer, based on the most common use cases suitable for BLE mesh [14].

As pointed out, relays and friend nodes must, in the general purpose use case of the BLE mesh, be powered on at all times. However, for more significant deployment of BLE mesh networking for future IoT applications, devices reliant only on battery power would be more beneficial. So, in this paper, we present an evaluation of a battery-powered relay node design, and of the effect of different operating policies on the relay's lifetime, and the *Packet-Delivery-Rate* (PDR).

The relay's main purpose is two-fold: 1) to allow for installation of BLE mesh networks where wall-power is not available for intermediate nodes, and 2) to allow for packet relaying in indoor environments where *Line-of-Sight* (LOS) is more important than actual transmission range. Although BLE 5 now supports transmission ranges up to 100 m [15], this greater transmission range would mainly allow for a reduction in the number of relays by covering longer distances in LOS with fewer devices. This is typically not the case in most scenarios, with exceptions such as large warehouses.

We consider use cases where the end-nodes periodically transmit sensor data, and rely on the relays to extend their range to the gateways. The relays must be strategically placed such as to, ideally, ensure LOS throughout the indoor space. This contrasts with the conventional friend node and low-power node topology, as the relay/friend nodes must now preserve battery life. We also contemplate that end-nodes may be subject to an application dependent degree of mobility, e.g., if they are attached to mobile equipment, or are intended to verify the existence of assets in storage. In this regard, the end-nodes themselves may or may not also be battery powered, but our focus is on the lifetime and forwarding efficiency of the relay as a function of its configuration.

Continuous operation by the relays would result in an unsuitably short battery life. Therefore, by configuring their listening period with a low duty-cycle, the battery life can be extended. As expected, this results in packet losses, especially as the traffic is uplink (from the end-nodes to the gateway), whereas in the conventional case, the network configuration is designed to assure high PDR. However, some applications may not consider that all data is high priority, and some degree of data loss and/or end-to-end delay may be acceptable.

We present an in-house design for a battery-powered BLE relay, and characterize the system's packet loss in different conditions. Specifically, we vary the number of client nodes, the listening time spent on each BLE channel, and apply two different forwarding policies. Additionally, we subject the system to noise from other Bluetooth devices external to the network. We validate the operation of our BLE relay design by manufacture and assembly, employing some of the units as beacons (so we may configure transmission periods), while another unit performs the relay function under several software configurations which implement our operating policies.

This paper is organized as follows: Section II reviews related work, Section III describes the network topology we addressed, Sectionsec:node presents the design characteristics of the BLE relay node, and the configurable operating parameters, like the duty cycle and forwarding policy. Section V presents experimental evaluation of packet reception rates for different scenarios. Section IV concludes the paper.

II. Related Work

As we mentioned previously, the performance of BLE based mesh networking can vary based on use-case, and on which quality metric we wish to ensure, e.g., longevity, PDR, or resiliance to failure. A comprehensive survey on the research efforts in BLE mesh topologies is presented by Darroudi *et al.* [16]. The survey categorizes and compares nearly 30 approaches to BLE network designs, including standardization solutions proposed by the Bluetooth SIG and the *Internet Engineering Task Force* (IETF), academic solutions, and proprietary solutions. Additional studies address emergent applications, limitations, and potential improvements [11], [13], [17], [18].

A major distinction between mesh approaches is whether data is transmitted by flooding (e.g., using the BLE advertising channels), or through end-to-end connections through specific nodes. A comparison is presented in [19], where the authors compare the Trickle flooding protocol [20] with the FruityMesh connection based protocol [21]. Both are evaluated regarding their multi-hop efficiency, for a network of nine intermediate nodes placed between two source and sink nodes. The packet delivery ratio and the end-to-end delay are measured. Both approaches are comparable in this scenario, with a packet delivery rate of close to 40% when 10 packets are generated per second by the source node. FruityMesh suffers an end-to-end delay which is approximately 9× higher compared to Trickle, but in turn requires 3× less power.

Kim *et al.* [22] present *BLEMesh*. A packet forwarding protocol is proposed to transmit batches of packets. Less transmissions are required in total to transport data end-to-end, through intermediate nodes, relative to naive flooding or routing based approaches. The packets include priority tables used by intermediate nodes to determine if a received packet should be re-transmitted, based on whether or not that packet was already forwarded by a node of higher priority. A downside is that the payload capability of the BLE packet diminishes as the number of nodes and batch size increases. A simulated evaluation for a mesh with 5 nodes, and assuming only one advertising channel, achieves a reduction of 54.5% in the required number of transmissions, relative to flood routing.

Brandão *et al.* [23] propose the Drypp protocol, based on the Trickle flooding protocol [20]. Trickle is a mesh network protocol for BLE where each node captures and attempts to re-transmit data at a later time, unless it meanwhile listens to redundant transmissions sent by other nodes. Drypp introduces a load balancing method which relies on dynamic adaptation of the protocol parameters based on each node's battery level. For three test nodes implementing the Drypp protocol,

an 11% increase in network lifetime was achieved relative to Trickle, in exchange for a 7.5% decrease in throughput.

A BLE mesh network relying on a routing protocol is evaluated in [24]. The proposed mesh network is designed for environmental monitoring and disaster scenarios, and both the edge (sensor nodes) and the Wi-Fi capable gateway nodes are battery powered. Information is propagated based on Trickle routing [20]. To extend battery life, the sensor nodes are periodically shut off, and modifications to the Trickle algorithm are introduced to prevent packet loss due to these power-down periods. Given the periods for listening and transmission time, the authors estimated a lifetime of 589 days for a sensor node, and 511 days for a gateway, when equipped with 6000 mAh and 8000 mAh lithium polymer batteries, respectively.

The work in [25] addresses optimization of the use of Bluetooth relays in mesh networks. Connection-less mesh networks propagate data by controlled flooding between nodes, until the destination node of a particular data packet is reached. However, this leaves the network vulnerable to excessive flooding as a function of the number of nodes used as relays and/or selected to be relays. The authors employ state-of-the-art relay selection algorithms to a BLE mesh network, and evaluate the effect of six different relay selection algorithms to a Connected Dominating Set (CDS) representation of the mesh. Using an in-house simulator, different relays densities were tested with two end nodes exchanging 1000 messages one-way. The lowest packet loss can be achieved by computing the routing with the fewest hops, but the lowest power consumption is possible for a genetic algorithm which finds the minimum CDS of the network, at the cost of suffering the highest packet loss (as high as 80%).

In [26], a method for relay node management is proposed based on a tree representation for the mesh network, together with an integer linear programming formulation which minimizes the number of relay nodes required to ensure connectivity between all nodes. The algorithm requires that the number of nodes and network topology be know to determine the relay routing. Using an in-house simulator, the authors evaluate the routing efficiency and energy consumption of a system composed of up to 100 nodes in an indoor configuration where LOS is not possible for all pairs of nodes. A power consumption reduction of up to $12\times$ is claimed over the conventional case where any relay node can be used as a relay during forwarding (i.e., flooding).

In [27] the same issue of relay selection to avoid excessive flooding and collisions in BLE mesh topologies is addressed. The authors argue that BLE mesh is designed for simple devices, and that therefore relay selection algorithms may be difficult to implement. Regardless, adequate selection is required in order to reduce broadcast flooding and overall power consumption of the mesh by avoiding redundant re-transmissions. Three relay selection algorithms are implemented by simulation of the lowermost layers of the BLE stack specification. They are compared in terms of PDR and resilience of the resulting network when relay failures occur. All algorithms start with a network discovery phase where only some nodes are chosen as relays. The first is a greedy algorithm that selects as relays those nodes which have the most yet undiscovered neighbours, the second algorithm applies pruning to redundant relays, and the third is a modification of a state-of-the-art *ad-hoc* networking algorithm [28]. For meshes with 100 nodes, and several tested node densities, all algorithms outperform the case where all nodes act as relays, which is the default beheviour of the BLE mesh specification. The choice of algorithm is concluded to be application dependant, as node failure resiliency and PDR vary based on node sparsity and the number of relay nodes.

Darroudi *et al.* [12] note that while BLE mesh was proposed to increase coverage over previously existing star topologies, the issue of the resulting power consumption is significantly overlooked in other works. The use case where end-nodes are considered lower-power nodes is addressed. The end-nodes sleep periodically, and the rest of the network assumes wall-powered relays, including friend nodes. At the time of writting, the friendship feature was not supported by the Nordic SDK, therefore the authors implement this behaviour at user level according to the BLE specification. The authors conclude that the design, configuration, and topology of a BLE mesh has significant impact on mesh lifetime and PDR, and since these are mostly application dependant (i.e., determined by requirements) no optimal configuration exists. While the BLE mesh specification allows for low-power end nodes to idle often, it requires that all intermediate relay nodes (i.e., friend nodes) be constantly powered to ensure packet delivery. Therefore, BLE mesh in general does not cover all use cases for IoT, specifically in the absence of wall power for all nodes, or where lifetime of battery-powered nodes is a concern.

In [13] improvements of the friendship mechanism are also addressed. The authors point out that although the friendship feature was introduced to support low-power nodes, the remaining nodes must remain in a constant scanning state. Additionally, the specification is better suited for use cases where the low-power nodes primarily transmit data, being unsuitable for cases where they are desired as data receivers. Also, the friendship mechanism suffers from asynchronicity between the friend node and its connected low-power nodes, and the retransmissions on the multiple advertising channels. The authors briefly evaluate the possibility of very precise time synchronization to reduce request and listening periods, and also transmisssion using only one channel, but discard these possibilities in favor of two modifications to the request and response protocol between friend nodes and low-power nodes. Firstly, the low-power node's listening window is reset per every packet the friend node dispatches, allowing for a burst mode where multiple requests by the low-power nodes are avoided, thereby reducing the amount of time the receiving low-power node must remain active. However, this in turn increases the polling time of other waiting nodes. Therefore, the low-power nodes are also modified in a second strategy, *Burst Transmissions with Listen Before Transmit* (BTLBT), where each low-power node first probes the medium before sending a request. The strategies were evaluated using five Nordic Semiconductor nRF52840 based devices, and by modifying an open-source implementation of the BLE mesh specification. For a configuration of one friend node and four low-power nodes, and a power consumption model presented

TABLE I
SUMMARY OF EXPERIMENTAL RESULTS OF COMPARABLE APPROACHES

Work	Protocol/Strategy	Type*	Total #Nodes	#Sources / #Sinks	Node Dist. (m)	TX Rate (p/second)	End-to-End Delay (ms)	Packet Delivery Ratio (%)	Node Power (mW)
[19]	FruityMesh	phys.	9	1 / 1	1.5	1, 5, 10	$\sim 3.8E3$	100%, ∼90%, 40%	9.4
[19]	Trickle	phys.	9	1 / 1	1.5	1, 5, 10	$0.5E3$	100%, 80%, ∼35%	28.5
[23]	Drypp	phys.	5	1 / 1	1.5	$\sim 24^{\S}$	-	91%‖	21.11‖
[22]	BLEMesh (batching)	sim.	5	1 / 1	-	-	-	-	-
[24]	Trickle (modified)	phys.	$6^{\dagger}$	6 / 1	∼5	1/60	5–10	-	1.85
[25]	MOC-CDS	sim.	77–572	1 / 1	7	$20–33^{\S}$	50–90	∼80%–95%	$N/A^{\ddagger}$
[25]	Genetic algorithm	sim.	77–572	1 / 1	7	$20–33^{\S}$	60–130	∼20%	$N/A^{\ddagger}$
[26]	Minimum Relay Tree	sim.	$50^{\dagger}$	50 / 1	20	$^5/_{60}$, $^2/_6$	97, 99	63%, 56%	1.5, 5.5
[26]	Full Flooding	sim.	$50^{\dagger}$	50 / 1	20	$^5/_{60}$, $^2/_6$	102, 128	90%, 82%	6.2, 22.6
[27]	Greedy Connect	sim.	1000	random	∼1.8–10	1-200	-	∼95%–30%	-
[27]	K2 Pruning	sim.	1000	random	∼1.8–10	1-200	-	∼95%–30%	-
[27]	*Dominator!* (modified)	sim.	1000	random	∼1.8–10	1-200	-	∼95%–20%	-
[13]	BLE Mesh Standard Spec.	phys.	5	1 / 4	1	∼4	248	100%	0.0624
[13]	Burst Tx.	phys.	5	1 / 4	1	∼16	67	100%	*N/A*
[13]	BTLBT	phys.	5	1 / 4	1	∼10	95	100%	0.05
Ours	Immediate Fwd.	phys.	4, 13	2, 11 / 1	∼1–5	1	-	58%, 16%	24.74
Ours	Batching and Fwd.	phys.	13	11 / 1	∼1–5	1	-	35%–9%	24.75–6.19

*Simulation or Physical, †+1 Gateway, ‡Given only as relative decrease, §To the best of our understanding
‖Relative to Trickle, ‖Derived from reported values

by [12], the BLE mesh standard was evaluated versus the proposed burst and BTLBT strategies. Depending on the expected downlink traffic, the lifetime of the low-power nodes can be increased by up to 2.5 months for the specific use case, which affords long idle periods for both the friend and low-power nodes. This approach is similar to the batching and forwarding approach we present, but the relay/friend node takes on an active role rather than a passive role, by holding and forwarding messages while periodically preserving its own battery.

In general, the choice of protocol and network topology is application dependant [11], [12]. Table I summarizes the results from the experimental evaluations shown in this section, including our own. The values reported are our best effort at a comparison of the presented approaches. Depending on the respective experiments, some columns show either scalar, ranges of values, or lists (correspondence between list values is kept column to column). Node power reports the power consumption of each node of the tested mesh, taking into account the entire operating time, including any sleep periods of the nodes (i.e., the average power consumption throughout the experiment lifetime).

The experiments we conducted can be categorized as controlled flooding mechanism, but where we rely on details specific to a class of applications to determine forwarding behaviour. We consider end nodes with a constant packet rate, and envision a topology for the network where a relay is responsible for the end nodes within its range. Additionally, we are not concerned with end-to-end delay, as data is non-critical and given equal importance. We also conduct experiments while introducing real-world noise due to other wireless devices external to the network, which we have not observed in other works we have identified. We implement a behaviour similar to the BLE friend node, but we consider the case of unidirectional data transmissions, where the end-nodes do not idle, and instead the relay (acting as a friend, due to data batching), must sleep in order to preserve its own battery. As for the battery life of end-nodes, if we envision applications where the data to be retrieved originates from mobile equipment or assets in long-term storage, where longer lasting batteries and/or very lower transmission periods can be used to not compromise functionality. Our batching and forwarding approach is not dissimilar to Marco *et al.* [29], also mentioned in [27], where end-nodes randomize advertisement intervals, and only first hops are repeated, but only once. We study the effect of further repetitions.

III. NETWORK TOPOLOGY

The use-case network topology for the evaluation of our relay, and respective forwarding policies, is shown in Figure 1. We target use cases where the end nodes are battery powered, and periodically transmit information about the environment (e.g., sensor data). The gateways are BLE/Wi-Fi devices which synchronize the status of the network with the centralized system. The devices were programmed by resorting to version 15.2.0 of the nRF5 SDK [30], which is certified to be BLE 5 compliant by the Bluetooth SIG.

One of the characteristics of BLE is the transmission range (approximately 20 m). This means that either all nodes placed throughout the site have to be within this range of a wall-powered gateway in order for data to be retrieved by those nodes, or that data is forwarded through nodes. The former is a potentially expensive solution, and the later is the object of study on mesh network routing protocols. However, if the end-nodes are simple sensors and cannot move data to and from each other (or if they are physically placed in such a way that a sequence of hops from end node to gateway cannot be established), more sophisticated battery-powered intermediate nodes are required which do not gather data themselves, but serve as range extenders to the gateways.

This paper presents a design of a relay node, which functions as a packet receiver, gatherer, and re-transmitter.

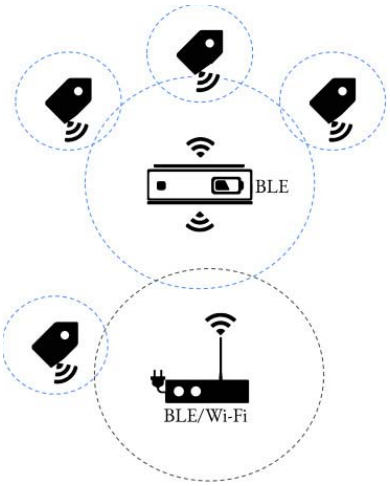


Fig. 1. BLE mesh topology, with battery-powered end nodes and intermediate relay nodes, and wall-powered BLE/Wi-Fi gateways interfacing with an upstream server system.

This makes it possible to extend the network range in situations where the indoor configuration or cost do not allow for a more ubiquitous distribution of wall-powered gateways. It also provides a cheaper solution relative to fully-fledged gateways, since it may replace them where Wi-Fi capabilities are not needed. Additionally, since the relays are battery powered, they are easy to relocate according to changes in the application requirements, or simply to tune the quality of the sensed data.

IV. BLE Relay Node

The purpose of the BLE relay device is to serve as a packet forwarder. It discards (i.e, does not forward) packets originating from devices which are not part of its own network. Currently this is done by MAC address filtering. The only payload sent is the identification of each node.

We implemented two functionally identical relay designs, both based on a single Nordic Semiconductor *nRF52832* micro-controller [31], which performs the packet reception and re-transmission, and idles the relay by going into a low-power mode. The configuration parameters listed earlier, such as listening intervals and periodicity, are controlled by the firmware residing on the non-volatile program memory of the *nRF52832* chip. All relay implementations are composed by one single-layer, dual-sided, FR-4 PCB with a 1 mm thickness.

The first prototype relay contains the *nRF52832* chip, a J-Link type programming header, and a single 3.3 V CR2032 button cell battery. The relay is considerably small, with a 23 mm × 38 mm × 10 mm profile. The antenna for reception and transmission of Bluetooth packets is a co-planar *Inverted F Antenna* (IFA), tuned for 2.4 GHz.

The schematic of the second prototype is shown in Figure 2. It is designed for a longer lifespan, relying on a series of four 3.3 V AA batteries when deployed in a location where wall power is unavailable, which is the primary use-case of the device. Alternatively, a mini-USB connector accepts a 5 V input. An *LTC4419* chip [32] is used as a power selector, which prioritizes the USB power input. A *TPS62125* [33] regulates the chosen input to 3.3 V for

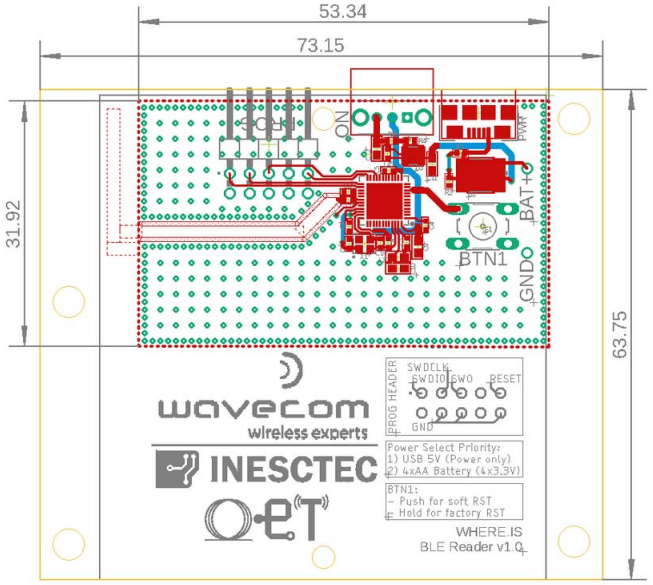


Fig. 2. Schematic view of the BLE relay prototype (dimensions in millimeters).

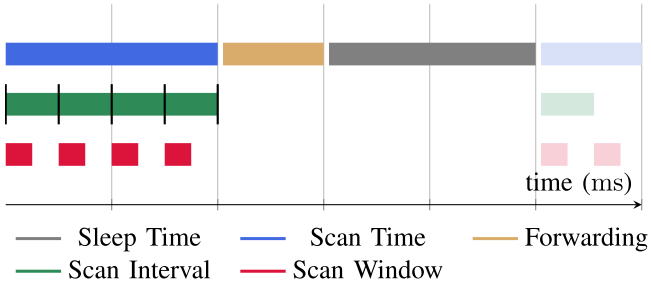


Fig. 3. Temporal diagram of relay operation cycle. The length of the sleep time (radio off, and sleep mode enabled for the *nRF52832*) determines the overall duty cycle. The *forwarding* period applies for one of our forwarding policies (Section V-C).

the *nRF52832*. Finally, the J-Link programming header powers the device in the absence of other power sources. The antenna design is identical to that of prototype *A* (albeit with a longer trace to the PCB edge, of 2.1 cm), and the device is 74 mm × 64 mm × 25 mm. Experimental evaluations in this paper consider only this relay variant.

The relay's software can accept a number of configuration parameters which will be the focus of the experimental evaluation. Figure 3 shows the cyclical operation mode of the relay during scanning. The relay stays in a given channel during a *scan interval*, and listens on that channel during the length of the *scan window*. In our tests we vary the length of the scan interval and set the scan window to an equal value. We evaluate the effects of two forwarding policies and estimate lifetime of the devices as a function of the sleep time (for the best performing scan interval and policy). Only advertising channels are used, and paired connections are not established, which is typical for one-way sensor meshes.

V. Experimental Evaluation

We evaluate the relay's performance regarding packet reception and forwarding, for different scan interval lengths,

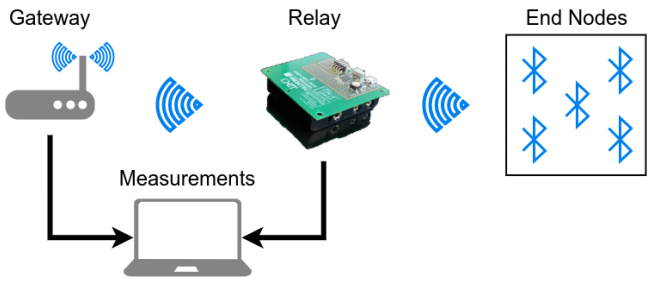


Fig. 4. Experimental setup for relay efficiency evaluation.

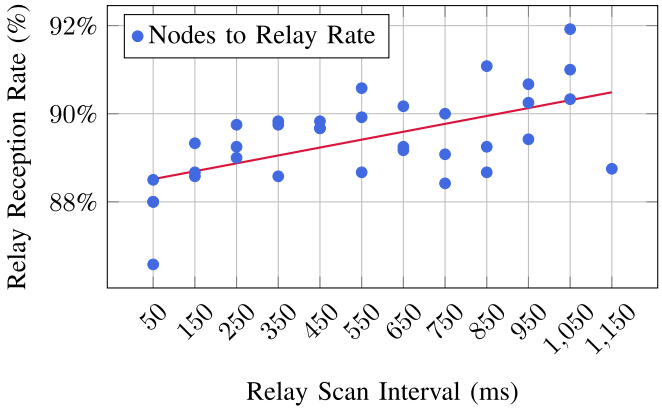


Fig. 5. Reception rate between the nodes and relay, and the relay and gateway, for three runs per relay scan interval (for 1200 packets sent by two nodes, transmitting at 1s intervals), without forwarding, and 17 nodes acting as noise.

policies, and sleep time. We employed the experimental setup show in Figure 4. In addition to the elements of the system shown, additional BLE nodes were placed in the environment, to act as noise, thus subjecting the system to a realistic operating condition. For all our tests, the *scan window* occupies the entire duration of the *scan interval*, in order to evaluate only the effects of the listening time, forwarding policy and sleep time. Exploring the effects of the length of sleep time (i.e., device duty cycle), in conjunction with non-equal scan window and interval lengths, on power savings and performance is out of the scope of this paper.

Given this, we evaluated the following characteristics:

- the rate of packets received by the relay while subject to noise, for different scan intervals (i.e., advertising channel switching periods);
- the forwarding efficiency between the relay and a gateway using an immediate forwarding policy, first with two client nodes, and then with 11 client nodes;
- forwarding efficiency for 11 nodes, under a policy which buffers received packets and forwards replicas to the gateway, to reduce the overhead of switching between radio modes;
- power consumption as a function of device duty cycle (i.e., sleep time).

In order to account for all transmitted and received packets, the relay and the terminal gateway communicate every packet received via serial connection. Each packet is annotated with the originating node. Since the transmission period of the nodes is known, we know the total transmitted packets for a given run time. We can then compute the packet losses in different conditions, between the nodes and the relay, and between the relay and the gateway.

A. Relay Reception Efficiency for 2 Nodes

In this test, the relay's packet reception rate under noise was tested for two client nodes, set to transmit advertising packets with period of 1 s. The test environment contained another 15 BLE nodes, external to the network, advertising at different intervals and thus acting as noise.

We varied the relay's *scan interval* between 50 ms and 1150 ms. The *scan window* occupies the entire period. What is measured in this case is the packet reception rate under noise, and due to the intrinsic loss of packets due to the randomness of the selected transmission and reception channels. The Bluetooth specification outlines a total of 40 channels, three of which (37, 38 and 39) are used for advertising packets.

Figure 5 shows the measured reception rates of the relay. Three runs were performed per configuration. Per run, each of the two nodes transmitted 600 packets. For a transmission rate of 1 packet per second, this totals an experimental time of 90 min per configuration. For all experiments the average reception rate is 88% ($\sigma = 1.02\%$).

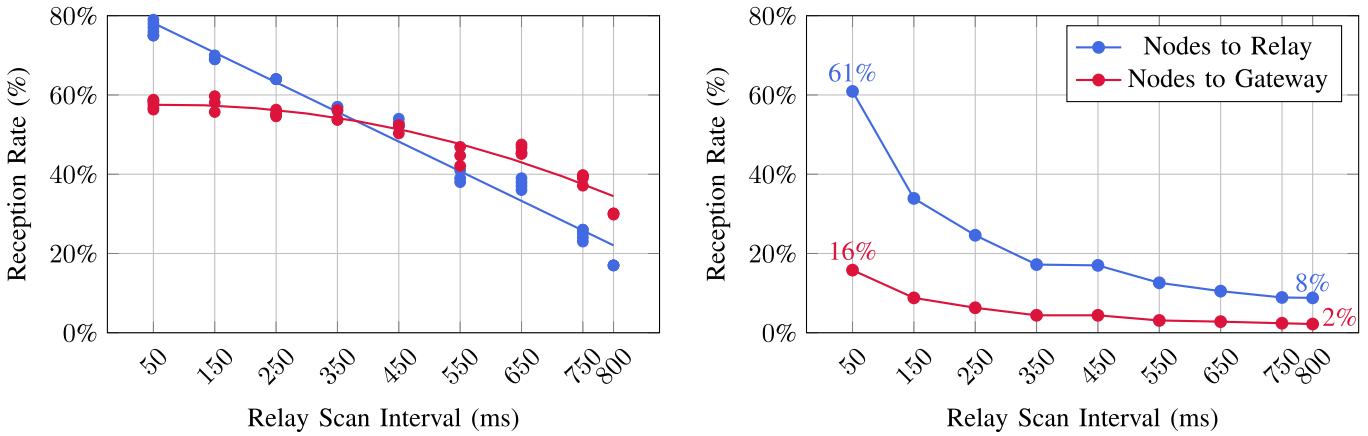
The *scan interval* does not affect the reception rate significantly. Even so, it is marginally more efficient for the relay to stay tuned into a single channel for as long as possible, i.e., longer scan intervals. This might contribute to a slightly reduced packet loss since less time is spent switching radio channels, which contributes to idle time. Also, since the Bluetooth protocol also dictates that an advertising event must be sent by a node on all three channels, the likelihood of the relay capturing a packet is higher by staying on a single channel for a period of time which is greater than the node's transmission period.

Note that in this scenario the relay's radio never transmits, and we evaluated the best case reception rate in a noisy scenario. Since the radio is half-duplex, once the relay begins forwarding packets, its reception rate will consequently decrease, as we present next.

B. Relay Forwarding Efficiency for 2 and 11 Nodes

Figure 6 shows the reception efficiencies between the nodes and the relay, and between the relay and gateway. Figure 6a shows the case with 2 client nodes, and 15 nodes acting as noise, and Figure 6b shows the case with 11 client nodes, and 6 nodes acting as noise. In these experiments, the sleep time is zero, as we wish to evaluate the performance, for a long period of operation, only as a function of the network size, scan interval, and noise introduced by other devices. The relay has an immediate forwarding policy for every packet received.

Figure 6a shows that the relay experiences a greater packet loss relative to the data in Figure 5, since it was configured to interrupt the scan interval and re-transmit immediately. This policy intended to reduce the travel time of the packets to the gateway. However this means that only one packet is relayed per scan interval, which explains the loss of packets from the nodes to the relay. Consequently, the number of packets



(a) relay and gateway reception rate for 2 nodes, for a total of 1200 packets sent by 2 nodes at 1 s intervals, with 17 nodes acting as noise

(b) relay and gateway reception rate for 2 nodes, for a total of 39600 packets sent by 11 nodes at 1 s intervals, with 6 nodes acting as noise

Fig. 6. Reception rate for the relay and gateway for two scenarios under an immediate forwarding policy.

forwarded to the gateway diminishes as the scan interval increases.

For scan intervals greater than 350 ms, the number of packets received by the gateways actually exceeds those forwarded. This is due to two factors. Firstly, for forwarding the relay must be switched to advertising mode for a duration such that only one packet is sent. However, non-deterministic behaviour during channel switching and switching between reception and transmission sometimes produces duplicate packets. Secondly, the gateway may receive packets directly from the nodes, depending on transmission power. This leads to an apparent increase in system performance for lengthier scan intervals, despite the relay's losses.

Figure 6b shows the same metrics when 11 nodes are introduced into the system. For the same reason as before, the reception rate (for both the relay and gateway) decreases with the relay's scan interval. However, this case shows how the relay effectively acts as an intermediate buffer to hold packets. The shorter the scan intervals, the quicker the relay echoes packets, decreasing the likelihood that packets are missed while the gateway is occupied, either by being in a non-listening state, e.g., switching between channels, or by being busy processing beacons received either directly from the nodes or by the relay.

However, even in the best case, only approximately 16% of the total packets sent arrive at the gateway, which implies significant energy expenditure by the beacons without benefit. The next section improves this with a different forwarding policy.

C. Relay Forwarding Efficiency for 11 Nodes & Batching Policy

We programmed the relay with a forwarding policy based on a listening period, and a forwarding period. During the listening period, the relay accumulates the captured packets, e.g., 4 packets from node #1, 10 from node #2, and one from node #3. During forwarding, the relay echoes up to N repetitions of a packet per node, regardless of how many packets were received per node. For instance, for 10 packets received for node #1, five echoes will be transmitted. This reduces the total traffic, and also normalizes the amount of packets sent upstream to the gateways, potentially boosting reception of packets sent by nodes under noisier conditions.

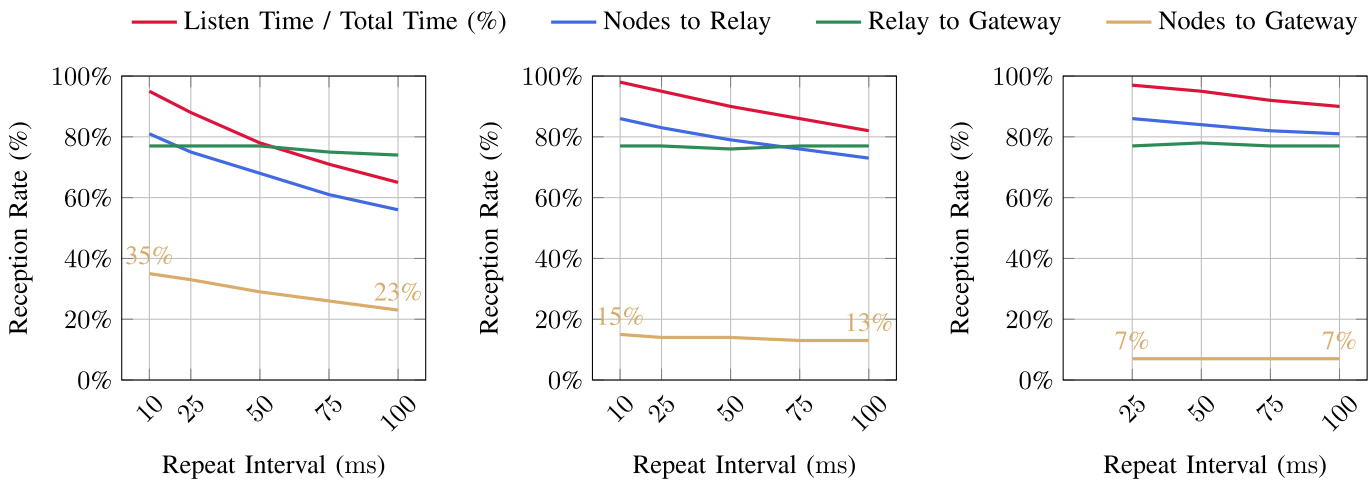
Figure 7 shows the reception rates, this time including also the rate of successful transfer between the relay and gateway. The three scenarios employ a listening period of 10 s, and a different number of packet repetitions each, e.g., five packet repetitions for Figure 7a. For each case, the interval between repetitions is also varied. Once again, the sleep time is zero, and the *scan interval* is 50 ms for all cases.

The *listening time* is also shown, which represents the amount of time during each listen-and-forward cycle that the relay is listening. The relay first listens during the *scan time* (S_{Time}) (switching between channels every *scan interval*), and buffers the packets. Then it enters forwarding mode where each packet is re-sent a given number of times ($Nr_{Repeats}$) at a set interval ($R_{Interval}$). Given that there are 11 nodes, the ratio between listening and forwarding time can be estimated as:

$$L(\%) = \frac{S_{Time}}{S_{Time} + R_{Interval} \times Nr_{Repeats} \times N_{Nodes}} \quad (1)$$

In the best case, with 5 repetitions at 10 ms interval, up to 35% of packets are now successfully forwarded to the gateway, which is 2.2× increase in performance relative to immediate forwarding. Although the relay captures less packets directly from the nodes, due to the lengthier forwarding period, the overall forwarding efficiency is higher.

That is, further energy savings could be obtained by now adjusting the end nodes to a less frequent transmission period, as the effort required to successfully deliver a packet to the gateway has decreased. In a multi-relay scenario, a superior performance should be expected, although the best strategy regarding scan interval, repeat interval, and repeat count would have to be determined. However, a possible approach would be to have each relay in the system forward only a subset of



(a) reception rates for 5 packet repetitions from relay to gateway

(b) reception rates for 2 packet repetitions from relay to gateway

(c) reception rates for 1 packet repetition from relay to gateway

Fig. 7. Reception rates between the several levels of the network, for different forwarding repetitions (i.e., Figures 7a to 7c), and for different intervals between repetitions. The *listening time* indicates the time the relay spends listening to the nodesand the remaining time is spent forwarding. The scan interval and window are 50 ms for all cases.

all nodes, thus reducing its own load and preventing excessive in-system noise. We leave these aspects as future work.

D. Estimated Power Consumption Vs. Sleep Time

This section explores the power consumption in continued operation as a function of sleep time, given that the forwarding rates during uptime are indicated by the previous experiments. To retrieve power consumption, we utilized a power profiler kit from Nordic Semiconductors [34].

We first use the power profiler to measure the current draw during radio operation (i.e., during *scan window* periods). Regardless of configuration values, the relay draws 7.5 mA.

We then evaluate the power consumption for different duty cycles defined by the *scan* and *sleep* times. The *scan interval* and *window* remain equal at 50 ms, and adopt a batching policy with 5 repetitions and a 10 ms repeat time. The efficiency for this case was 35%. The average current draw and efficiency as a function of the duty cycle can be calculated by the product of the cycle and these baseline values of 7.5 mA and 35%, respectively. The battery life is computed based on the relay's four AA batteries totaling 12000 mAh.

Figure 8 shows the resulting efficiencies and battery life. The efficiency is shown based on the experimental runs with 11 nodes transmitting at a 1 s interval. In this case, a duty cycle of 100% leads to the 35% efficiency, but a battery life of only approximately 2.16 months. To attain a battery life of a year, a duty cycle of 20% is required, with an estimated efficiency of 7%. Note that the effective efficiency of forwarding remains 35%, since a duty cycle of 20% implies that, in the best case, 20% of all packets would be forwarded.

Additionally, note that for all experiments, the efficiency is dependant on the total amount of packets sent by the nodes. These experimental runs impose a 1 s period per node. For some applications like sensor networks for temperature or light

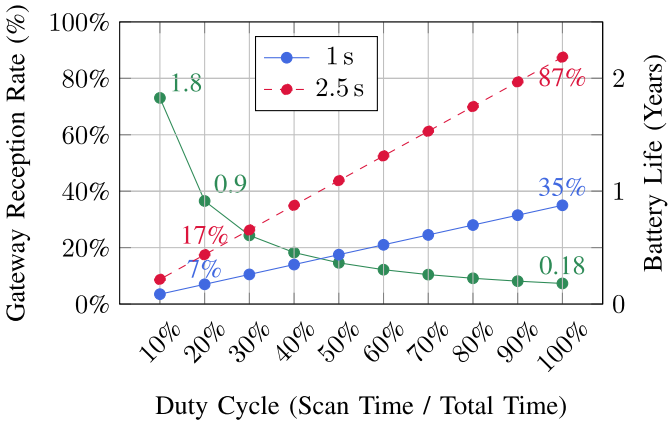


Fig. 8. Battery life and gateway reception rate as a function of duty cycle. Reception rate is shown for a system with 11 client nodes with a transmission period of 1 s and 2.5 s. The relay is configured for a 50 ms scan interval/window, and with a batching policy of 5 repetitions and 10 ms repeat time.

intensity readings with periods of in the order of minutes, longer update periods would be tolerable, especially since a long battery life is also desired for the nodes.

We can extrapolate that for a node transmission period of 2.5 s, the relay could forward 87% of the packets, given the same up-time and fewer packets, a behaviour similar to the one observed for [19] (see Table I). For a duty cycle of 20% to ensure close to a year of battery life, the estimated efficiency would increase by 10 percentage points.

The efficiency and power consumption are still subject to additional parameters such as multiple relays, tweaks to the batching policy, different values for scan and sleep times which resulting the same duty cycle, node transmission period and number of nodes. As other works have illustrated, an optimal networking policy would largely be application dependant. [12]

VI. Conclusion

We have presented an evaluation of a Bluetooth device and packet forwarding policies in mesh networks. The objective of the relay device is to extend the range of transmission between end devices, such as Bluetooth nodes, and the gateway devices, which are wall-powered and communicate with a central server. The relays allow for more area coverage without additional gateways, which are more costly, and without further investment in wall-power infrastructure.

We first evaluated the relay's packet reception with 2 nodes, under noise generated by 17 nodes which were not part of the system, for values of the *scan window* between 50 ms and 1150 ms, and found that the relay can receive up to 90% of the node transmissions, for a node transmission period of 1 s.

We then evaluated the forwarding efficiency, measured as the number of packets received by the gateway versus the total number of packets sent by the nodes. For a policy where the relay immediately forwards a received packet, only 16% of packets sent by 11 nodes are received by the gateway. By employing a policy of deferred forwarding, and multiple packet repetitions per listened node, this increases to 35%, given the setup of a single relay under load of 11 clients transmitting at a period of 1 s. Varying these parameters would influence the delivery ratio, and an acceptable ratio would be application-dependant.

Finally, we measured the power draw of the device using a power analyzer, and estimated the lifetime of the four AA batteries (12 000 mA h) for different duty cycles and node transmission periods.

Future work includes study of different end-node transmission periods, the effect of hops over multiple relays, and the effect of larger relay batch sizes and/or packet repeats. Devices with capabilities antenna arrays for beamforming may provide even more efficient indoor mesh networking by avoiding reflections and optimizing energy usage [11].

References

[1] S.-M. Darroudi and C. Gomez, "Bluetooth low energy mesh networks: A survey," *Sensors*, vol. 17, no. 7, p. 1467, Jul. 2017.

[2] C. Gomez and J. Paradells, "Wireless home automation networks: A survey of architectures and technologies," *IEEE Commun. Mag.*, vol. 48, no. 6, pp. 92–101, Jun. 2010.

[3] F. Zafari, A. Gkelias, and K. K. Leung, "A survey of indoor localization systems and technologies," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 3, pp. 2568–2599, 3rd Quart., 2017.

[4] M. N. K. Boulos and G. Berry, "Real-time locating systems (RTLS) in healthcare: A condensed primer," *Int. J. Health Geograph.*, vol. 11, no. 1, p. 25, 2012.

[5] J. Konecny, M. Prauzek, R. Martinek, L. Michalek, and M. Tomis, "Real-time patient localization in urgent care: System design and hardware perspective," in *Proc. IEEE 20th Int. Conf. e-Health Netw., Appl. Services (Healthcom)*, Sep. 2018, pp. 1–5.

[6] B. Molina, E. Olivares, C. E. Palau, and M. Esteve, "A multimodal fingerprint-based indoor positioning system for airports," *IEEE Access*, vol. 6, pp. 10092–10106, 2018.

[7] R. Giuliano, F. Mazzenga, M. Petracca, and M. Vari, "Indoor localization system for first responders in emergency scenario," in *Proc. 9th Int. Wireless Commun. Mobile Comput. Conf. (IWCMC)*, Jul. 2013, pp. 1821–1826.

[8] A. Rácz-Szabó, T. Ruppert, L. Bántay, A. Löcklin, L. Jakab, and J. Abonyi, "Real-time locating system in production management," *Sensors*, vol. 20, no. 23, p. 6766, Nov. 2020. [Online]. Available: https://www.mdpi.com/1424-8220/20/23/6766

[9] Bluetooth SIG. (2017). *Mesh Profile Specification Version 1.0*. [Online]. Available: https://www.bluetooth.org/docman/handlers/downloaddoc.ashx?doc_id=429633

[10] (2014). *Bluetooth Specification Version 4.2*. [Online]. Available: https://www.bluetooth.com/specifications/specs/core-specification-4-2/

[11] A. Hernández-Solana, D. Pérez-Díaz-De-Cerio, M. García-Lozano, A. V. Bardají, and J.-L. Valenzuela, "Bluetooth mesh analysis, issues, and challenges," *IEEE Access*, vol. 8, pp. 53784–53800, 2020.

[12] S. Darroudi, R. Caldera-Sànchez, and C. Gomez, "Bluetooth mesh energy consumption: A model," *Sensors*, vol. 19, no. 5, p. 1238, Mar. 2019. [Online]. Available: https://www.mdpi.com/1424-8220/19/5/1238

[13] D. Hortelano, T. Olivares, and M. C. Ruiz, "Reducing the energy consumption of the friendship mechanism in Bluetooth mesh," *Comput. Netw.*, vol. 195, Aug. 2021, Art. no. 108172. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1389128621002309

[14] Bluetooth SIG. (2017). *Mesh Model Specification Version 1.0*. [Online]. Available: https://www.bluetooth.org/docman/handlers/downloaddoc.ashx?doc_id=429634

[15] Bluetooth SIG. (2019). *Bluetooth Specification Version 5.1*. [Online]. Available: https://www.bluetooth.com/bluetooth-resources/bluetooth-core-specification-v5-1-feature-overview/

[16] S. Darroudi and C. Gomez, "Bluetooth low energy mesh networks: A survey," *Sensors*, vol. 17, no. 7, p. 1467, Jun. 2017.

[17] J. Yang, C. Poellabauer, P. Mitra, and C. Neubecker, "Beyond beaconing: Emerging applications and challenges of BLE," *Ad Hoc Netw.*, vol. 97, Feb. 2020, Art. no. 102015. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1570870518307170

[18] M. R. Ghori, T.-C. Wan, and G. C. Sodhy, "Bluetooth low energy mesh networks: Survey of communication and security protocols," *Sensors*, vol. 20, no. 12, p. 3590, Jun. 2020. [Online]. Available: https://www.mdpi.com/1424-8220/20/12/3590

[19] 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 *Proc. IEEE 28th Annu. Int. Symp. Pers., Indoor, Mobile Radio Commun. (PIMRC)*, Oct. 2017, pp. 1–6.

[20] P. Levis, T. Clausen, J. Hui, O. Gnawali, and J. Ko, *The Trickle Algorithm*, RFC 6206, Internet Requests for Comments, RFC Editor, Mar. 2011. [Online]. Available: http://www.rfc-editor.org/rfc/rfc6206.txt

[21] M-Way Solutions. (2015). *Fruitymesh*. Accessed: Nov. 29, 2021. [Online]. Available: https://github.com/mwaylabs/fruitymesh

[22] H.-S. Kim, J. Lee, and J. W. Jang, "BLEmesh: A wireless mesh network protocol for Bluetooth low energy devices," in *Proc. 3rd Int. Conf. Future Internet Things Cloud*, Aug. 2015, pp. 558–563.

[23] A. S. Brandao, M. C. Lima, C. J. B. Abbas, and L. J. G. Villalba, "An energy balanced flooding algorithm for a BLE mesh network," *IEEE Access*, vol. 8, pp. 97946–97958, 2020.

[24] A. Gogic, A. Mujcic, S. I. Hodzic, and N. Suljanovic, "Performance analysis of Bluetooth low energy mesh routing algorithm in case of disaster prediction," *World Acad. Sci., Eng. Technol., Int. J. Comput., Electr., Automat., Control Inf. Eng.*, vol. 10, pp. 929–935, Jan. 2016.

[25] M. Reno *et al.*, "Relay node selection in Bluetooth mesh networks," in *Proc. IEEE 20th Medit. Electrotech. Conf. (MELECON)*, Jun. 2020, pp. 175–180.

[26] A. Bęben, A. Bąk, and M. Sosnowski, "Efficient relay node management method for BLE MESH networks," *Int. J. Electron. Telecommun.*, vol. 66, no. 1, pp. 29–35, 2020. [Online]. Available: https://www.infona.pl/resource/bwmeta1.element.baztech-87c2a182-86a1-4308-a876-1e069e99d1ae and http://psjd.icm.edu.pl/psjd/element/bwmeta1.element.oai-journals-pan-pl-113741

[27] E. A. J. Hansen, M. H. Nielsen, D. E. Serup, R. J. Williams, T. K. Madsen, and R. Abildgren, "On relay selection approaches in Bluetooth mesh networks," in *Proc. 10th Int. Congr. Ultra Modern Telecommun. Control Syst. Workshops (ICUMT)*, Nov. 2018, pp. 1–5.

[28] F. Kuhn, R. Wattenhofer, Y. Zhang, and A. Zollinger, "Geometric ad-hoc routing: Of theory and practice," in *Proc. 22nd Annu. Symp. Princ. Distrib. Comput.*, 2003, pp. 63–72, doi: 10.1145/872035.872044.

[29] P. Di Marco, P. Skillermark, A. Larmo, and P. Arvidson, "Bluetooth mesh networking," Ericsson, White Paper 284 23-3310 Uen, 2017. [Online]. Available: https://www.ericsson.com/en/reports-and-papers/white-papers/bluetooth-mesh-networking

[30] Nordic Semiconductor. (2017). *nRF5 SDK Version 15.2.0*. [Online]. Available: https://www.nordicsemi.com/Products/Development-software/nRF5-SDK/Download

[31] (2017). *nRF52832 Product Specification v1.4*. [Online]. Available: https://infocenter.nordicsemi.com/pdf/nRF52832_PS_v1.4.pdf
[32] Analog Devices. (2015). *18 V Dual Input Micropower PowerPath Prioritizer*. [Online]. Available: https://www.analog.com/media/en/technical-documentation/data-sheets/4419fa.pdf
[33] Texas Instruments. (2017). *TPS62125 Step-Down Converter*. [Online]. Available: http://www.ti.com/lit/ds/symlink/tps62125.pdf
[34] Nordic Semiconductor. (2021). *Nordic Power Profiler Kit*. [Online]. Available: https://www.nordicsemi.com/Products/Development-hardware/Power-Profiler-Kit

Nuno Paulino received the M.Sc. degree in electrical and computer engineering from the Faculty of Engineering, University of Porto, in 2011, and the Ph.D. degree in electrical and computer engineering from the University of Porto in 2015. He is currently an Assistant Professor at the University of Porto, and a Researcher with the INESC TEC. His research interests include reconfigurable and embedded systems in FPGAs, computing architectures, and tools for hardware/software codesign.

Luís M. Pessoa received the "Licenciatura" and Ph.D. degrees in electrical and computer engineering from the Faculty of Engineering, University of Porto, in 2006 and 2011, respectively. He is a Senior Researcher at the Centre of Telecommunications and Multimedia, INESC TEC, where he is the Lead of the area of Optical and Electronic Technologies. His research interests include coherent optical systems, radio-over-fiber, RF/microwave devices and antennas, and underwater wireless power/communications.

André Branquinho received the bachelor's degree in physics from the University of Aveiro in 2010, and the master's degree in electrical and communication engineering from the Technische Universität Hamburg in 2014, focusing on signal processing and wireless communication. He has been the Product Manager at Wavecom S.A., Aveiro, Portugal, since 2017. Currently, he is responsible for development of RFID and real-time localization products.

Rafael Almeida received the master's degree in computers and telematics engineering from the University of Aveiro in 2018. He is a Telecommunications and Software Engineer and a Technical Project Manager at Wavecom S.A., Aveiro, Portugal. Previously, he was a Software Engineer at Altice Labs.

Igor Ferreira received the master's degree in electronic and telecommunications engineering from the University of Aveiro in 2009. His experience includes antenna and RF design, and software-defined radio. He is the Director of Development at Wavecom S.A., Aveiro, Portugal, and a Product Development Manager at Wavesys.