BMADS: BLE Mesh Asynchronous Dynamic Scanning

D. Perez-Diaz-de-Cerio, J. L. Valenzuela, Mario Garcia-Lozano,
Á. Hernández-Solana, and A. Valdovinos

***Abstract*—The Bluetooth low-energy (BLE) mesh profile, currently standardized by the Bluetooth Special Interest Group (SIG), assumes that friend and relay nodes should be continuously scanning the channel ready to receive any incoming transmissions (TXs). This is very inefficient in terms of energy consumption. Hence, we present BMADS, an asynchronous dynamic scanning mechanism designed to reduce the overall energy consumption of the mesh network. BMADS is implemented as a new feature for the currently published BLE mesh specifications without modifying its core. This feature reduces the scanning cycles of the nodes, but still ensuring the reliability. To make this possible, BMADS sends a new control message sequence before transmitting the data. This sequence puts the nodes into continuous scan mode, preparing the network for data TX. The proposal can be used in BLE sensor networks that can tolerate a certain TX delay and require especially low-energy consumption. BMADS allows a theoretical reduction in the energy consumption of the relay nodes up to 98%.**

***Index Terms*—Bluetooth mesh, Bluetooth, energy, wireless mesh networks.**

I. Introduction

Wireless sensor networks are being introduced in many application fields: national park monitoring, agriculture, farm and cattle control, mountain hydrology measurements, mobility, and smart cities, etc. [1]–[3]. These systems are also being adopted by the industry to improve the control, monitoring, and automation in terms of robustness, reliability, security, latency, and jitter. In many cases, these systems use wireless technologies as a solution in what is often called the Industrial Internet of the Things (IIoT) [4], [5].

Wireless mesh networking has been proposed as a suitable solution in IIoT applications given its better resilience, reliability of multipath routing, and low infrastructure costs [6]–[8]. Besides these features, IIoT applications usually require ultra long battery life, preferably measured in years [9], [10]. Many of the devices in IIoT need to work autonomously with limited access to energy resources. Moreover, they might be inaccessible and/or the operational costs due to battery handling may be prohibitive. Additionally, data generation can be sparse in time, such as temperature, ambient brightness, motion events, humidity, etc. All these examples present low-throughput and do not have low-latency requirements. Therefore, energy efficiency is essential in the industrial market. For example, [11]–[14]. The first one [11] proposes an industrial scenario where the authors monitor beer fermentation with a minimum time between samples of six hours. Or, for instance, Ojha *et al.* [12] presented a smart agriculture application that requires to transmit at intervals of 60 s, 15 min, or even 60 min. If we analyze sensors for smart health, Hornero *et al.* [13] just needed 2000 transmissions (TXs) per day to enable a communication system for people with speech disabilities. Or, in the smart cities and mobility field, the work of [14] proposes sensors that produce from 10 to 8640 detections in one day.

There are many technological proposals supporting mesh networking, such as ZigBee, WirelessHart, 6LoWPAN, or Thread, all of them based on the IEEE 802.15.4 standard family. In this context, the Bluetooth Special Interest Group (SIG) has developed a new mesh specification [15] based on Bluetooth low-energy (BLE) [16]. Bluetooth mesh adds new capabilities to the BLE standard. Although it is not included as part of the original core specification, this new profile must be adapted to it. This fact presents some drawbacks and compromises that add some complexity to the configuration and deployment process. Some of these design drawbacks affect energy consumption as will be discussed next.

Trying to reduce this consumption, the Bluetooth mesh profile defines low power nodes (LPNs) that include power saving features. These nodes can be inactive for a long period due to infrequent data TX. The use of LPNs can reduce the energy consumption of these devices, but note that every LPN should be connected to a friend node. The friend node receives messages from one or several LPNs that should be transmitted to the mesh network and, vice versa, it holds the messages intended for the LPNs until they wake up and poll the friend node for them. Therefore, a friend node must remain permanently active. Another point is that, due to its low-power nature, LPNs cannot act as relays in a mesh network.

The nodes with the relay feature activated receive and forward packets to distribute them over the network, and

Manuscript received April 22, 2020; revised July 3, 2020; accepted August 9, 2020. Date of publication August 19, 2020; date of current version February 4, 2021. This work was supported in part by the Spanish Ministry of Science with ERDF Funds under Grant RTI2018-099880-B-C32, Grant RTI2018-095684-B-I00, and Grant RTI2018-099063-B-I00; and in part by the Government of Aragon (Reference Group T31_20R). *(Corresponding author: D. Perez-Diaz-de-Cerio.)*

D. Perez-Diaz-de-Cerio, J. L. Valenzuela, and Mario Garcia-Lozano are with the Signal Theory and Communications Department, Universitat Politècnica de Catalunya, 08860 Castelldefels, Spain (e-mail: dperez@tsc.upc.edu; valens@tsc.upc.edu; mariogarcia@tsc.upc.edu).

Á. Hernández-Solana and A. Valdovinos are with the Aragon Institute for Engineering Research, University of Zaragoza, 50018 Zaragoza, Spain (e-mail: anhersol@unizar.es; toni@unizar.es).

Digital Object Identifier 10.1109/JIOT.2020.3018022

are an even more crucial part of the network. However, the BLE mesh standard defines: "A device supporting only the advertising bearer should perform passive scanning with a duty cycle as close to 100% as possible in order to avoid missing any incoming mesh messages or provisioning packet data units (PDUs)" [15]. Therefore, relay and friend nodes must remain constantly alert, which significantly implies high energy consumption. For example, a real device consumes around 5 mA while scanning [17], [18]. In figures, this means that a relay powered by a 10 000-mAh battery would have a battery life under three months. Thus, BLE loses part of its low-energy principles when used in a mesh network.

Therefore, in scenarios like the ones described in [11]–[14] where sensors do not need to transmit very frequently nor present low-delay requirements the sensors are more energy efficient if configured as LPNs. However, having all the friends and relays in the network permanently listening for TXs becomes very inefficient. This issue is remarked in [19], a very interesting study where we can find a model of BLE mesh energy consumption based on measurements performed on a real hardware platform using the nRF51422 chipset from Nordic Semiconductor [20]. This article is useful to evaluate the energy consumption of the current state of the standard [15]. However, their conclusions state that "Bluetooth Mesh cannot be considered as a general-purpose Internet of the Things (IoT) technology because its application is limited to the scenarios where its backbone can be appropriately powered". We think that BLE mesh could still be very interesting for this kind of applications, although it should be optimized in terms of energy consumption.

To the best of our knowledge, the works that propose new strategies for energy optimization in BLE mesh are scarce, but BLE with its new mesh profile is becoming an interesting and hot topic rapidly. For example, in [21], BLE mesh is proposed as one of the enabler technologies in multihop networks among other possibilities. However, none of the analyzed papers addresses the subject of energy savings in BLE mesh. The survey mostly emphasizes this aspect in solutions that change the routing or use a scatternet topology (connection-oriented). The also recently published work of Yang *et al.* [22], surveys several built applications, the problems to adopt BLE in new areas, and current academic and industrial solutions that expand the capabilities of BLE, as can be the new mesh profile. However, once again, its energy optimization is not discussed. This is not the case of [23], where two mesh approaches that fit within BLE operation are compared using power consumption as one of the metrics. They analyze flooding using the Trickle algorithm [24] and a connection-oriented networking solution called Fruitymesh [25] to conclude that the optimal mesh approach depends on the application. Researching further in these proposals, the Trickle algorithm as well as the most recent Drypp algorithm presented in [26], help to reduce the amount of redundant network traffic by adapting TX rates to network density. However, both of them use a continuous scan approach. Thus, the impact over the energy consumption is considerably lower than a duty cycle based proposal.

The FruityMesh solution is based on neighbor-only routing, where no routing tables are stored but a connection is established between two nodes and kept open until the packets reach their final destination. Greenlink [27] or the work discussed in [28] are also very interesting approaches that focus on minimizing the number of central/relay nodes, but both of them, as happens with Fruitymesh, are connection-oriented so the energy savings are moderate. For example, Greenlink reduces the energy consumption a 50% compared with the standard deployment. The Minimum Relay Tree presented in [29], also focuses on minimizing the number of central/relay nodes as happens with [28], but requires a previous analysis of the network and the manual selection of which nodes should act as relays. And, if the topology changes, the network should be reconfigured again manually.

Another strategy to reduce consumption could be to apply low power listening mechanisms like the ones listed for other wireless sensor technologies in [30]. This article presents three main categories: 1) scheduled; 2) protocols with common active periods; and 3) asynchronous MAC. However, from our point of view, the use of these proposals or a store-and-forward solution implies a drastic change to the bearer, network, and transport layers of BLE mesh standard. So, most of them cannot be directly used in BLE mesh.

Thus, all of this background generates the main motivation behind BMADS: to demonstrate that, with some tweaks, BLE mesh can still be considered as a general-purpose technology for the IoT and IIoT. However, this comes with some major challenges. First, the proposal should be based on real devices and measures. Second, the proposal should allow high energy savings in BLE mesh networks. And, third, the proposal should not modify the core of the standard as other approaches do. We consider this an essential point because in this way it can be more easily adopted by the industry.

We leave out of the scope of this article considering other proposals, like RFC7668 [31] which adapts IPv6 to BLE or like [32] which presents a novel architecture for IPv6 over BLE mesh networks. The first is an adaptation based on connection mode and BLE mesh is based on the TX of broadcast messages. And, the second uses continuous scan mode, so the energy consumption is still high.

Therefore, we propose to use the asynchronous dynamic scanning (ADS) mechanism of BMADS to reduce the energy consumption of the mesh network. This mechanism introduces a scanning duty cycle in the relay procedure. It is based on the generation of a control message sequence to put the relays into continuous scan, preparing them for the TX of data. In summary, BMADS main contribution is a reduction in energy consumption of relay nodes up to 98%, much more than the 50% claimed by solutions like Greenlink. And, at the same time, it has been designed as a new feature that can be applied directly to the current published mesh profile unlike [25], [28], etc.

This article is organized as follows. First, we briefly introduce the concepts of BLE mesh which are relevant for understanding our proposal. This includes message frame encapsulation and special focus is set on the publish, network,

TABLE I
BMADS TIMERS LOOKUP TABLE

Random delays	
$ASAP$	As soon as possible
$rand$	Undefined value in the specifications
$rand_{10}$	Uniformly distributed between 0 and 10 ms
Publish retransmit interval	
$pubInt$	Generic publish retransmit interval
$pubInt^{\text{seq}}$	BMADS sequence publish retransmit interval
$pubInt^{\text{data}}$	Data publish retransmit interval
Network transmit interval	
$netInt$	Generic network transmit interval
$netInt^{\text{seq}}$	BMADS sequence network transmit interval
$netInt^{\text{data}}$	Data network transmit interval
Relay retransmit interval	
$relInt$	Generic relay retransmit interval
$relInt^{\text{seq}}$	BMADS sequence relay retransmit interval
$relInt^{\text{data}}$	Data relay retransmit interval
Scanning parameters	
$scanInt$	Generic scan interval
$scanInt^{\text{seq}}$	Scan interval in low power state
$scanWin$	Generic scan window
$scanWin^{\text{seq}}$	Scan window in low power state
Transmission duration	
T_{ADS}	Time in continuous scan to guarantee data reception
T_{ADS1}	Time to propagate the control sequence
T_{ADS2}	Time to transmit the data from source to sink
T^{seq}	Duration of a BMADS sequence
T^{data}_{N}	Time to transmit N messages of data in a single hop
T^{seq}_{AE}	Time of a BLE advertising event

and relay retransmission procedures of the managed flooding mechanism. Then, in Section III, we present the points that define our proposal: the format of the new control message, how to configure the TX sequence, and the adequate selection of the device scanning parameters. Later, we analyze the energy consumption of our proposal in comparison with the standard behavior of the specifications. Section V discusses the performance of the technique under different scenarios. Consumption and delay are quantified and stability issues are also analyzed. The impact of the tunable parameters is also evaluated and some rules-of-thumb are provided for an easy configuration. Finally, conclusions close this article.

II. BLE MESH BASIC OVERVIEW

To fully understand our proposal, two main aspects of the mesh specifications should be explained before: 1) the different timing relationships and 2) how to encapsulate the data to be transmitted.

A. BLE Mesh Packet Transmission Timing

The BLE mesh specification is designed to work on top of the BLE standard. In a BLE mesh network, nodes transmit and relay messages using the advertising bearer [15]. This bearer utilizes the non connectable undirected advertising event of the BLE standard. These events imply the TX from one to three advertising packets on the primary advertising physical channels: 37, 38, and 39. The time between the start of two consecutive non connectable undirected advertising events is configurable between 20 ms and 10.24 s. An extra random value between 0 and 10 ms (rand_{10}) is added to minimize collision occurrence between potentially synchronized nodes.

The specification defines a managed flood-based routing. It implements two main methods to restrict the unlimited relaying of messages over the mesh network: 1) the management of message caches done by relays and 2) the maximum number of times a packet can be relayed, namely, time-to-live (TTL). It also implements a TX procedure to increase reliability in the already crowded Industrial Scientific and Medical (ISM) band. This procedure relies on the inherent redundancy of the three advertising frequencies of BLE and the additional TXs of the mesh specifications.

There are two parameters defined in the source nodes which control how many times a message is transmitted or *published* in terms of the Bluetooth standard, and the interval between publications. These are the publish retransmit count (P_{RC}) and the publish retransmit interval steps (P_{RIS}) parameters. Although the information remains the same for a single message, the radio packets differ in their control content and sequence number. The P_{RC} parameter takes values between zero and seven, i.e., a message can be published between one and eight times. On the other hand, the P_{RIS} defines the publish interval (pubInt) as indicated by the following equation:

$$\text{pubInt} = (P_{RIS} + 1) \cdot 50 \text{ ms}. \tag{1}$$

Since the P_{RIS} is encoded with 5 b, pubInt ranges between 50 and 1600 ms in steps of 50 ms.

The standard also defines a composite state, the network transmit state, with two similar parameters: 1) the network transmit count (N_{TC}) and 2) the network transmit interval steps (N_{TIS}). This state determines the number of times that a network PDU originated in a node should be transmitted. Its behavior is different from the publish procedure. In this case, the repetitions are a verbatim copy of every published message. The N_{TC} is a 3-b field, so a network PDU is transmitted between one and eight times ($N_{TC} + 1$). The N_{TIS} also ranges from 0 to 31, but in this case, the step duration is 10 ms. Additionally, it is necessary to include a random interval between 0 and 10 ms. So, the time in milliseconds between two TXs is calculated as in

$$\text{netInt} = (N_{TIS} + 1) \cdot 10 \text{ ms} \tag{2}$$

the network interval (netInt), plus a rand_{10}. However, the minimum netInt is 20 ms due to the restriction on the advertising interval fixed by the used bearer.

The specifications also define the procedures and parameters of the relay nodes. These parameters are designed to guarantee the network reliability and, at the same time, to keep under control the flooding mechanism proposed in the standard.

On the one hand, the relays have two parameters similar to the ones that control the message publications: 1) the relay retransmit count (R_{RC}) and 2) the relay retransmit interval steps (R_{RIS}). R_{RC} controls how many times a packet is going to be retransmitted (up to eight). And, R_{RIS} controls the time

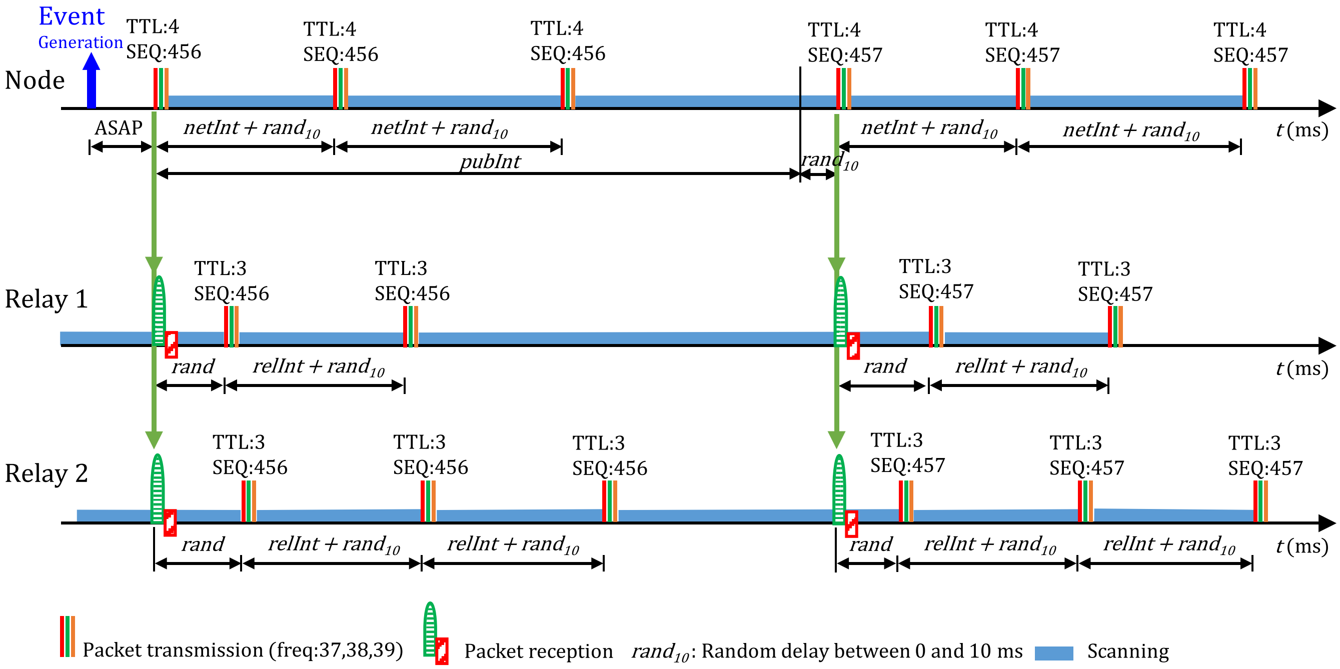


Fig. 1. Example of the publish/relay retransmit and network transmit mechanisms.

between retransmissions with a granularity of 10 ms and a minimum of 20 ms

$$\mathrm{relInt} = (R_{RIS} + 1) \cdot 10 \text{ ms}. \tag{3}$$

In this case, however, the standard recommends to include an undefined random time between the packet reception and the first retransmission to avoid collisions.

On the other hand, the relay nodes should maintain a cache to store previous retransmitted messages to prevent unnecessary retransmissions and overloading the network.

The previous paragraphs explain the procedures proposed by the mesh specification. Nevertheless, we have performed in-lab measurements with real devices (nRF52840 development kits (DKs) from Nordic Semiconductor). These tests show us the need to complement the basic TX model with additional restrictions.

1) The specifications indicate an undefined random delay for the first relay TX. Following our observations with Nordic devices, this vendor sets a maximum value equals to the advertising interval. This interpretation of the standard could lead to extreme cases in which the advertising interval is huge. Given this, we limit this delay to a random value between 0 and 10 ms. This is in concordance with the contention window applied by the rest of the retransmissions.
2) An additional restriction happens in the publish retransmission procedure. When the time gap between the last TX and the new scheduled publication is less than netInt. The TX is delayed to netInt ms. Therefore, this time space is always present between them.
3) The processing time at relay and sink nodes was also measured from the test bed. And, as per the measures, this time is very small in Nordic's devices. However, we consider that this should be taken into account in the equations because other devices present much higher values as we have observed in our previous works [33].
4) We also include the measured time of an advertising event (T_{AE}). We have checked that this time depends on the device implementation. The TX time depends on the payload size. However, the time between TXs within an event can vary for each chipset manufacturer.

Fig. 1 shows an example of different retransmission mechanisms. First, the generating node uses $P_{RC} = 1$ and $N_{TC} = 2$. Note that once an event is generated, the first message should be transmitted as soon as possible. When a new packet is received at the relay, the message is stored in the cache and it is retransmitted $R_{RC} + 1$ times. In the example, Relay 1 is configured with a $R_{RC} = 1$ and Relay 2 is configured with an $R_{RC} = 2$. Both relays are within the influence area of the generating node. If a packet is received several times, due to the transmitter having a $N_{TC} > 0$ or because the packet is received from two different relays, the relay process is not executed again as the packet is already present in the cache. Nevertheless, with the second publication, the process is launched again because the packet is different from the one stored in the cache.

Note that, later, we suppose that these variables, can be configured differently for transmitting the data or the control message. So, hereafter, we use superscripts to indicate this difference. Additionally, from now on, every time-referred equation, variable, or parameter is expressed in ms.

B. BLE Mesh Message Encapsulation

Fig. 2 depicts how a message is encapsulated through the different stack layers. Reading the figure from up to bottom the message goes from the application layer to the physical layer. There are two message types allowed: access and control messages. When the message to be transmitted is large, no matter

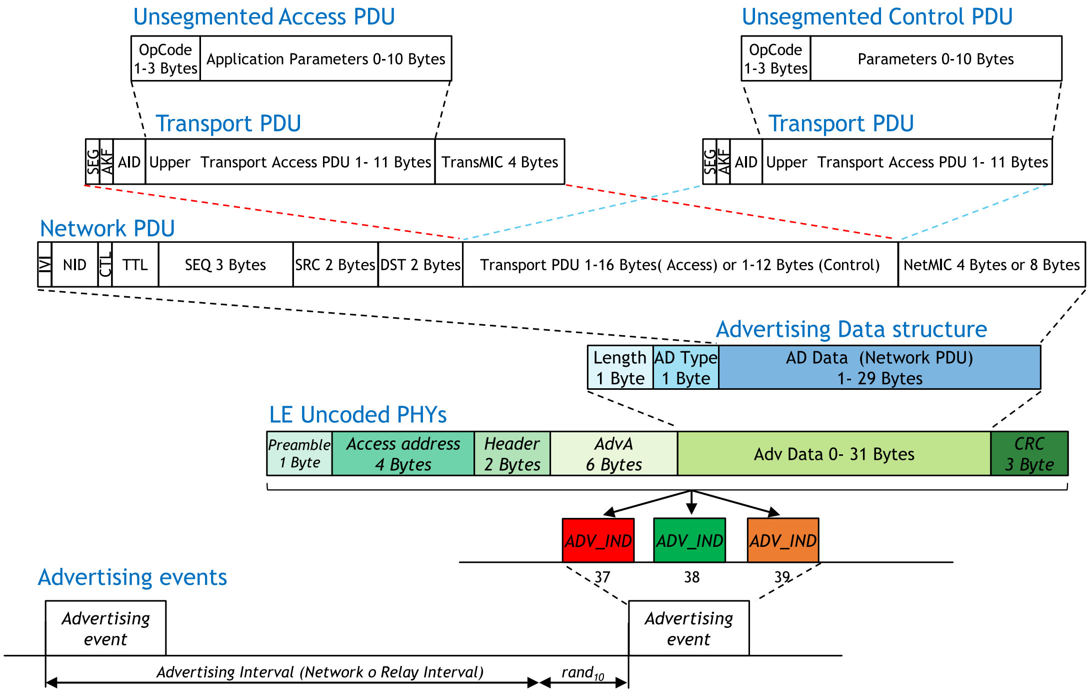


Fig. 2. BLE mesh message encapsulation.

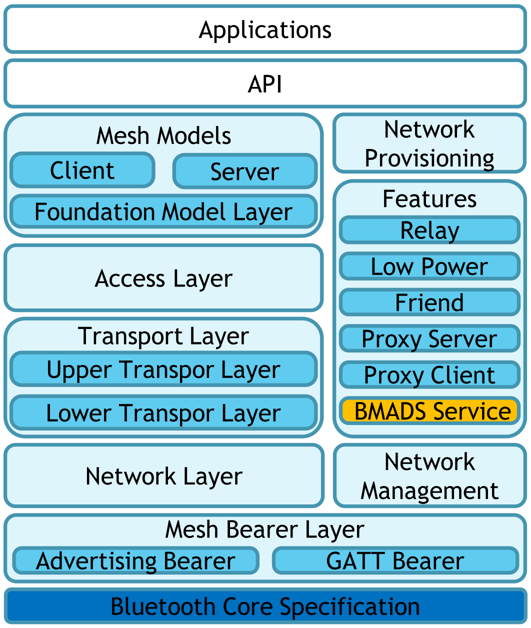


Fig. 3. Changes to the currently published BLE mesh stack.

its type, it can be segmented. An unsegmented access/control message can convey up to 10 bytes of data which is enough for the purposes considered in this article. The difference in the encapsulation of an access versus a control message comes in the transport and network layers. An access message divides the message integrity checks (MICs) in two 4-byte parts, one in the transport layer and the other in the network layer. This is done to allow the encryption of data application messages. For example, a relay may retransmit a message if it belongs to the network (Network MIC OK), but not necessarily has to know its contents (Transport MIC KO). Control messages, instead, are critical messages used to manage the network. Thus, they use also 8 bytes but all of them are dedicated to protecting the network PDU. In this way, both message types have the same length at the network layer. This network PDU is introduced into a standard BLE advertising data structure with a defined advertisement type: $0 \times 2A$ for mesh messages or $0 \times 2B$ for mesh beacons. Here on, the advertisement is passed to the physical layer as any other BLE advertisement.

III. Proposal

The BMADS proposal is a TX scheme adapted to the BLE mesh specifications (see Fig. 3) where a modified BLE mesh stack protocol is depicted. The new proposal is defined as an additional feature in orange, the BMADS service. This optional feature enables applications that are delay-tolerant and/or make infrequent TXs to reduce the consumption in the network relays. An essential characteristic of this new feature is that is added without modifying in any way the physical and MAC layer of the current standard. In fact, it does not either require any change in the protocol stack structure. The only necessary change is the definition of a new control command and its management. This can be implemented using the RFU fields available in the current standard frame definitions. However, like an LPN needs another neighbor node to be a friend, to apply BMADS, all the relays in the network should use this new feature.

The main idea is to adjust the relay scanning parameters to be able to receive a new control message, the ADS message, without losses, and maximizing energy savings. This message switches the relays to continuous scan mode. It should be detected by the relay receivers when they are in the active state of their duty cycle, i.e., scanning. To achieve this, we

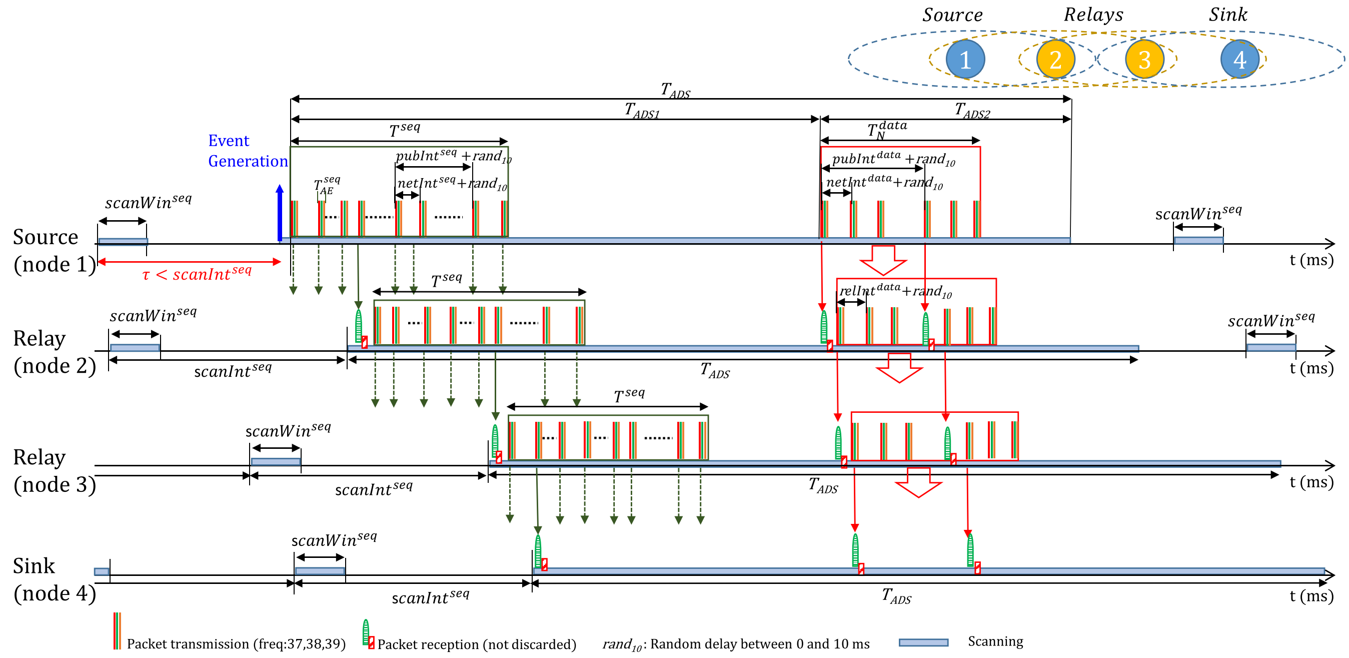


Fig. 4. BMADS mechanism proposal.

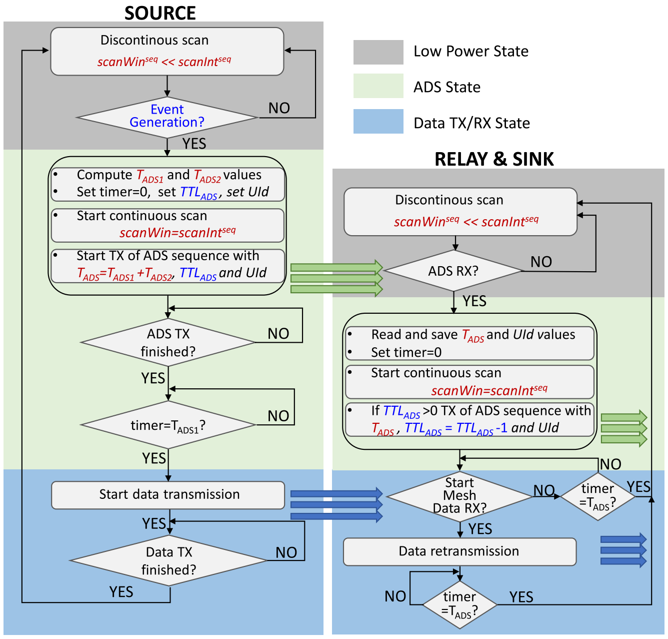


Fig. 5. BMADS algorithm.

use the network transmit and publish retransmit procedures to generate a sequence of packets and ensure its reception. Once a relay processes the message, it equals the scan window to the scan interval. It waits for the data and, if necessary, generates another ADS message to continue its propagation over the network. The objective is to prepare the network for the TX of data. These messages should specify the amount of time that the devices should stay scanning continuously. They should also contain a unique identifier (UId) and a second TTL field (TTL_{ADS}). These three would be the parameters of an ADS message. Next, we illustrate the mechanism with the example depicted in Fig. 4 and the help of the algorithm of Fig. 5. We will delve into the details afterward.

Thus, Fig. 4 represents a scenario with one source, two relays, and the final destination of the data. The topology and coverage areas of each node are shown at the top right corner of the figure. Note that this is just an example, our proposal is valid over any topology. Unlike normal BLE mesh operation, all the devices in the network are configured with a low-energy scanning duty cycle while there is nothing to transmit. This corresponds to the gray areas in Fig. 5. The duty cycle of the scanner is defined by means of the scan window (scanWin) and scan interval (scanInt) parameters. Therefore, scanWin < scanInt. These values should be configured to ensure the reception of at least one of the ADS message sequence packets sent from any neighbor. The discontinuous blue lines of Fig. 4 represents when the devices are actually scanning, i.e., they are in the active part of their duty cycle. Although it is not necessary for the source node to be scanning, we handle all the nodes in the same way for simplicity. Thus, firstly, the source generates a packet and enters into the ADS state depicted in green in Fig. 5. The node initializes all the necessary variables and starts continuous scan. Prior to its TX, it generates a sequence of control packets immediately. This sequence is based on the repetition of the ADS message following the standard publish and network transmit procedures. In the example of Fig. 4, the first TXs of the ADS message reach the second node when it is not scanning and these frames are lost. Once the relay starts scanning and receives one ADS message packet, it temporarily changes its mode to continuous scan and generates a new ADS message sequence. The same situation is resolved in the second relay. Then, when the sink receives the ADS message sent by the third node, it activates itself and the network is completely prepared to transport the data. Now, the network is in the same state as an standard BLE mesh without any modification. After a precalculated time, the source sends the data which is relayed all the path to the sink. This is depicted as the blue area of Fig. 5. This introduces a delay since the event generation, but guarantees that all the nodes implied are ready for data reception/TX. When the process is over, every node in the network returns to its low-power scanning duty cycle. All the time variables shown in Figs. 4 and 5 are explained in Section III-E.

A. ADS Message Parameters

To fulfill the desired behavior, each ADS message should only activate all the neighbors around the node transmitting the message. Thus, we propose to use the standard TTL field to limit this, fixing its value to zero. Therefore, an ADS message is never relayed. It is regenerated in each hop. Additionally, the address field should be set to the broadcast fixed group address 0 × FFFF allow the processing of the message by every node that receives it.

When a node receives an ADS message, it should enter into continuous scan mode for a predetermined amount of time. This time, T_{ADS}, is calculated at the source node and introduced into one of the parameters of the ADS message. This calculation, which is discussed in Section III-E, should consider the worst case scenario: the detection always happens at the last packet of the sequence and the random generators should return their maximum value.

The second parameter that should be included in the ADS message is the TTL_{ADS} value. Recall that the standard TTL, present in all mesh messages, is fixed to zero. Therefore, TTL_{ADS} is necessary to indicate the number of hops required to reach the destination and to limit the range of the mechanism. This value can be determined previously using a heartbeat message, for example. When a relay must generate a new ADS message, it reduces the TTL_{ADS} by one and keeps the rest of the parameters the same.

The third and last parameter of an ADS message must be a UId. A node that receives an ADS message with a previously received identifier should discard the message because it has already been relayed. The identifier is determined and introduced into the parameters of the first ADS message by the source. The relays should use the same identifier as the received message.

In summary, we propose to use 3 bytes for the first parameter (TTL_{ADS}) as we use the resolution step of CLK_1 defined in vol. 2—[16, Sec. 1.1]. Thus, in steps of 625 μs, we can keep the network on, up to 10485.76 s. For the second parameter, TTL_{ADS}, just 1 byte is needed. And, last, for the UId, we use the remaining 6 bytes to reach the maximum of an unsegmented control PDU (see Fig. 2).

B. BMADS Sequence Configuration

To configure the BMADS sequence, we use the N_{TC}^{seq}, N_{TIS}^{seq}, P_{RC}^{seq}, and P_{RIS}^{seq} parameters and calculate the associated values, $\text{netInt}^{\text{seq}}$ and $\text{pubInt}^{\text{seq}}$, as defined in Section II-A. These parameters should be selected in a way that we can create a long enough pattern to ensure that at least one ADS message is received during the active part of the scanner duty cycle.

Remember that the publish retransmission interval is defined in steps of 50 ms and the network TX interval must be a multiple of 10 ms with a minimum of 20 ms. Recall also that the number of network and publish TXs should be in the range between one and eight. Thus the basic structure of the network transmit state can be repeated, *published* using the standard terminology, up to seven more times. Thus, with the maximum values of N_{TC}^{seq} and P_{RC}^{seq}, we can send up to 64 ADS messages distributed over a long time interval, $T_{\text{ADS}}^{\text{seq}}$. This is a matter of discussion in Section V.

C. Redundancy

Once the BMADS sequence is determined, the $\text{scanWin}^{\text{seq}}$ and $\text{scanInt}^{\text{seq}}$ of the devices should be configured to ensure the detection of the ADS message. Additionally, we introduce a new parameter (k) into the model to adjust the reliability of the network. The redundancy of the control sequence and the data is controlled with their correspondents P_{RC} and N_{TC}. However, k represents the minimum number of consecutive ADS packets to be received considering error-free TXs. Thus, we ensure a successful ADS message reception even with up to $k-1$ consecutive packets with errors. Notice that k must accomplish

$$k < \left(P_{RC}^{\text{seq}} + 1\right) \cdot \left(N_{TC}^{\text{seq}} + 1\right) \tag{4}$$

which shows the maximum number of TXs available with the configured parameters of the BMADS sequence.

Although P_{RC}, N_{TC}, P_{RIS}, and N_{TIS} could be different for the data or the BMADS sequence as their target is different, we consider the same level of redundancy for the ADS messages and data ($N_{TC}^{\text{data}} = R_{RC}^{\text{data}} = k - 1$). Tuning these parameters, including k, not only affects the reliability of the TXs but also has a direct impact over the receiver scanning cycles. Hence, energy consumption depends heavily on how they are adjusted. We evaluate this in Section V.

D. Network Transmit State and pubInt^{seq}

The values of T_{ADS}, $\text{scanWin}^{\text{seq}}$, and $\text{scanInt}^{\text{seq}}$ depend on the relationship between $\text{pubInt}^{\text{seq}}$ and the current network transmit state. Hence, before performing their calculation, it is necessary to consider three different scenarios.

1) Scenario A:

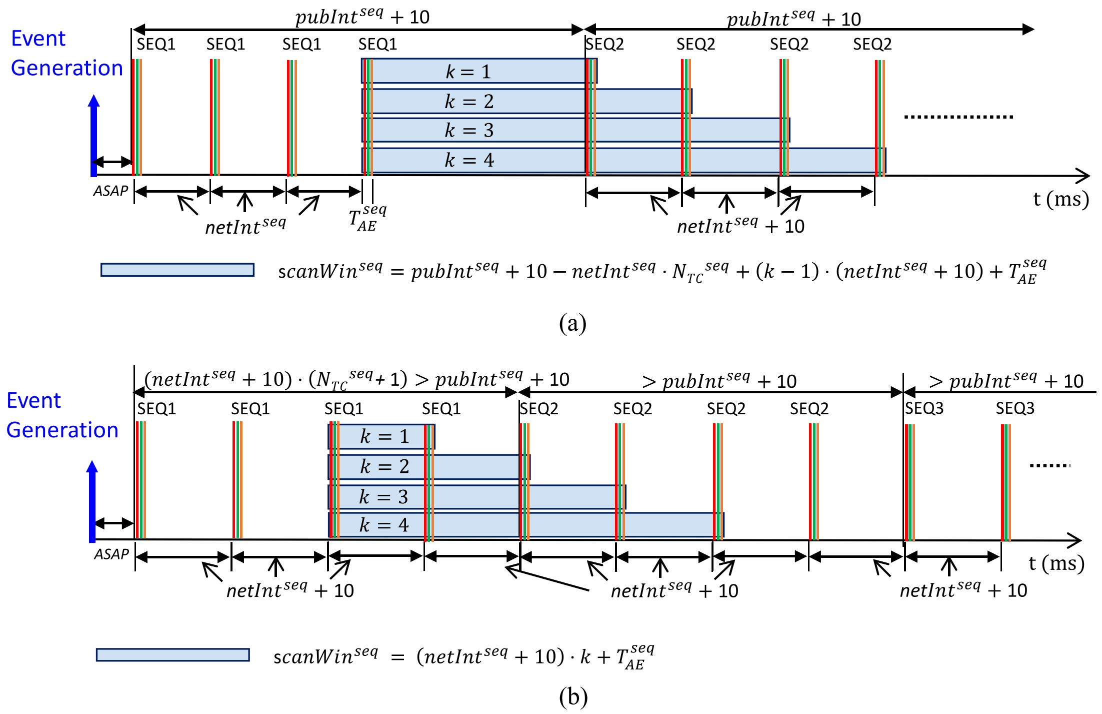
$$\text{pubInt}^{\text{seq}} \leq \text{netInt}^{\text{seq}} \cdot \left(N_{TC}^{\text{seq}} + 1\right). \tag{5}$$

In this case, $\text{pubInt}^{\text{seq}}$ is so short that, when it has to schedule the next publication of an ADS message, the current network transmit state has not finished. Thus, the new message is buffered and all the gaps between TXs follow $\text{netInt}^{\text{seq}} + \text{rand}_{10}$.

2) Scenario B:

$$\begin{aligned}\text{netInt}^{\text{seq}} \cdot \left(N_{TC}^{\text{seq}} + 1\right) < \text{pubInt}^{\text{seq}} \leq \left(\text{netInt}^{\text{seq}} + 10\right) \\ \times \left(N_{TC}^{\text{seq}} + 1\right).\end{aligned} \tag{6}$$

This scenario takes place when $\text{pubInt}^{\text{seq}}$ is between the limits defined for scenarios A and C. The 10 ms added to the $\text{netInt}^{\text{seq}}$ represent the maximum random delay.



(a)

(b)

Fig. 6. Examples to calculate the optimum ScanWin$^{\text{seq}}$. (a) scanWin$^{\text{seq}}$ when publInt$^{\text{seq}}$ $>$ (netInt$^{\text{seq}}$ + 10) $\cdot$ (N_{TC}^{seq} + 1) and $k \leq N_{TC}^{\text{seq}} + 1$. (b) ScanWin$^{\text{seq}}$ when publInt$^{\text{seq}}$ $\leq$ netInt$^{\text{seq}}$ $\cdot$ (N_{TC}^{seq} + 1).

3) Scenario C:

$$\text{publInt}^{\text{seq}} > \left(\text{netInt}^{\text{seq}} + 10\right) \cdot \left(N_{TC}^{\text{seq}} + 1\right). \tag{7}$$

This case is the opposite to scenario A, publInt$^{\text{seq}}$ is large enough to allow the end of the network transmit state. So, when a new publication is scheduled, all the retransmissions of the previous publication have been completed.

E. Calculation of T_{ADS}

T_{ADS} is the amount of time a node should be in continuous scan to guarantee the reception of the data (see Fig. 4).

As the nodes use different configuration parameters for transmitting the BMADS sequence and data, it is necessary to divide T_{ADS} into two parts

$$T_{\text{ADS}} = T_{\text{ADS1}} + T_{\text{ADS2}} \tag{8}$$

T_{ADS1} is the necessary time to propagate the TX of the control sequence through the network. It is defined as

$$T_{\text{ADS1}} = \text{TTL}_{\text{ADS}} \cdot T^{\text{seq}} + (\text{TTL}_{\text{ADS}} - 1) \cdot T_{\text{PROC}}^{\text{relay}} + T_{\text{PROC}}^{\text{sink}} \tag{9}$$

where T^{seq} is the duration of the BMADS sequence and $T_{\text{PROC}}^{\text{relay}}$ and $T_{\text{PROC}}^{\text{sink}}$ are the processing delays at the relays and sink, respectively.

At the same time, T^{seq} depends on the scenarios described in Section III-D. Thus, when the publication retransmissions are buffered because publInt$^{\text{seq}}$ is too short, i.e., scenarios A and B, T^{seq} takes the value of (10). In contrast, scenario C follows (11). In both equations, T_{AE} is the duration of a complete advertising event.

1) Scenarios A and B, (5) and (6):

$$\begin{aligned} T^{\text{seq}} = {} & P_{RC}^{\text{seq}} \cdot \left[\left(N_{TC}^{\text{seq}} + 1\right) \cdot \left(\text{netInt}^{\text{seq}} + 10\right)\right] \\ & + N_{TC}^{\text{seq}} \cdot \left(\text{netInt}^{\text{seq}} + 10\right) + T_{AE}^{\text{seq}}. \end{aligned} \tag{10}$$

2) Scenario C, (7):

$$T^{\text{seq}} = P_{RC} \cdot \text{publInt}^{\text{seq}} + N_{TC}^{\text{seq}} \cdot \left(\text{netInt}^{\text{seq}} + 10\right) + T_{AE}^{\text{seq}}. \tag{11}$$

For its part, T_{ADS2} is the needed time to transmit the data. In this case, we consider that the network transmit is finished when a new publication takes place for simplicity. Accordingly, if we want to transmit N_{data} packets, T_{ADS2} follows:

$$\begin{aligned} T_{\text{ADS2}} = {} & T_N^{\text{data}} + (\text{TTL}_{\text{ADS}} - 1) \cdot T_{REL} \\ & + (\text{TTL}_{\text{ADS}} - 1) \cdot T_{\text{PROC}}^{\text{relay}} + T_{\text{PROC}}^{\text{sink}} \end{aligned} \tag{12}$$

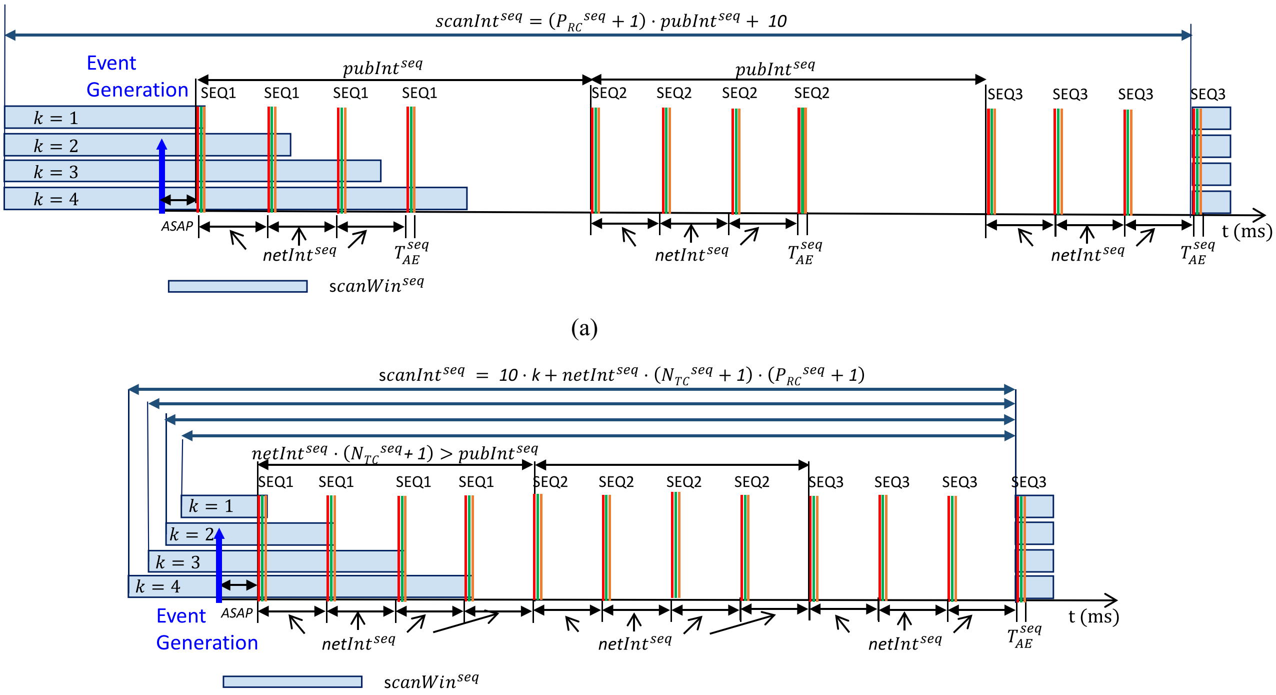
where

$$\begin{aligned} T_N^{\text{data}} = {} & N_{\text{data}} \cdot \left[P_{RC}^{\text{data}} \cdot \text{publInt}^{\text{data}} + N_{TC}^{\text{data}} \cdot \left(\text{netInt}^{\text{data}} + 10\right)\right] \\ & + (N_{\text{data}} - 1) \cdot \left(\text{netInt}^{\text{data}} + 10\right) + T_{AE}^{\text{data}} \end{aligned} \tag{13}$$

$$T_{REL} = \max(rand) + R_{RC}^{\text{data}} \cdot \left(\text{relInt}^{\text{data}} + 10\right) + T_{AE}^{\text{data}}. \tag{14}$$

F. Calculation of scanWin^{seq}

To optimize the consumption, it is necessary to minimize scanWin$^{\text{seq}}$. Remember that this is the period within a scanInt$^{\text{seq}}$ during which the device would be actively scanning for the reception of messages. And, therefore, out of the idle state. However, scanWin$^{\text{seq}}$ should be long enough to receive at least k ADS messages and guarantee the desired redundancy. Thus, the worst case scenario should consider a scanWin$^{\text{seq}}$


scanInt^seq = (P_RC^seq + 1) · pubInt^seq + 10
Event Generation
pubInt^seq
pubInt^seq
SEQ1
SEQ2
SEQ3
k = 1
k = 2
k = 3
k = 4
ASAP
netInt^seq
T_AE^seq
t (ms)
scanWin^seq
scanInt^seq = 10 · k + netInt^seq · (N_TC^seq + 1) · (P_RC^seq + 1)
netInt^seq · (N_TC^seq + 1) > pubInt^seq

(a)

(b)

Fig. 7. Examples to calculate the optimum $\text{scanInt}^{\text{seq}}$. (a) $\text{ScanInt}^{\text{seq}}$ when $\text{pubInt}^{\text{seq}} > \text{netInt}^{\text{seq}} \cdot (N_{TC}^{\text{seq}} + 1)$ and $k \leq N_{TC}^{\text{seq}} + 1$. (b) $\text{ScanInt}^{\text{seq}}$ when $\text{pubInt}^{\text{seq}} \leq \text{netInt}^{\text{seq}} \cdot (N_{TC}^{\text{seq}} + 1)$.

that includes the longest gap between TXs. We use two examples to illustrate how to calculate it for scenarios A and C. To simplify, we consider for these examples that $k \leq N_{TC}^{\text{seq}} + 1$. However, once understood both examples, we can infer the general equations for the three scenarios without restrictions.

The first example is depicted in Fig. 6(a), where it is shown an overlap of the scanner for different redundancy values (blue) with the TXs of the ADS messages. Here, we suppose that we are in scenario C. Thus, $\text{pubInt}^{\text{seq}}$ is long enough to allow the end of the previous network transmit state. This means that all ADS messages with sequence number 1 (SEQ1) are sent before the new publication is triggered. Note that, in this case, the longest gap between TXs comes just before a new publication. This is the gap between the TX of the last SEQ1 message and the first SEQ2 message in Fig. 6(a). To consider the worst case scenario, it is necessary to calculate when this gap reaches its maximum. To determine this, two things must be taken into account. In the first place, we must consider that the new publication obtains the maximum random value when calculating $\text{pubInt}^{\text{seq}} + \text{rand}_{10}$. Thus, the period between publications should be $\text{pubInt}^{\text{seq}} + 10$. In the second place, all the previous TXs of SEQ1 messages should be performed after just a $\text{netInt}^{\text{seq}}$ between them instead of the usual $\text{netInt}^{\text{seq}} + \text{rand}_{10}$. In other words, with the minimum of the extra random value, zero. For $k = 1$, it is necessary to capture just one advertising event. Thus, the $\text{scanWin}^{\text{seq}}$ only needs to cover this gap $(\text{pubInt}^{\text{seq}} + 10 - \text{netInt}^{\text{seq}} \cdot N_{TC}^{\text{seq}})$ and the first advertising event of the new publication $(T_{\text{AE}}^{\text{seq}})$.

Accordingly, if we need to capture more advertising events $(k > 1)$, we should add the next advertising event. This is done to consider the worst case scenario. However, this advertising event should happen after $\text{netInt}^{\text{seq}} + 10$. This supposes that the random number generator has returned its maximum value (worst case). Therefore, while $k \leq N_{TC}^{\text{seq}} + 1$, $\text{scanWin}^{\text{seq}}$ takes the value of the equation shown in Fig. 6(a).

The second example, shown in Fig. 6(b), depicts the scenario A. There, $\text{pubInt}^{\text{seq}}$ is too short, and the second publication should have been sent while there are still SEQ1 messages to be sent. As per the measures done with Nordic devices, in this case, this and the subsequent publications are buffered. This is why every TX in the sequence is done every $\text{netInt}^{\text{seq}} + \text{rand}_{10}$. However, we must consider the worst case again. In this case, this comes when the $\text{scanWin}^{\text{seq}}$ includes as many gaps as k, and these gaps correspond to $\text{netInt}^{\text{seq}} + 10$, the maximum possible random value.

Hence, extrapolating the results from both examples, the general equations for the three scenarios without limitations are as follows.

1) Scenario A, (5):

$$\text{scanWin}^{\text{seq}} = \left(\text{netInt}^{\text{seq}} + 10\right) \cdot k + T_{\text{AE}}^{\text{seq}}. \tag{15}$$

2) Scenario B, (6):

$$\begin{aligned}\text{scanWin}^{\text{seq}} &= \text{pubInt}^{\text{seq}} - \text{netInt}^{\text{seq}} \cdot N_{TC}^{\text{seq}} + 10 \\ &\quad + (k-1) \cdot \left(\text{netInt}^{\text{seq}} + 10\right) + T_{\text{AE}}^{\text{seq}}.\end{aligned} \tag{16}$$

3) Scenario C, (7):

$$\begin{aligned}\text{scanWin}^{\text{seq}} &= \text{pubInt}^{\text{seq}} + 10 - \text{netInt}^{\text{seq}} \cdot N_{TC}^{\text{seq}} \\ &\quad + \left\lfloor \frac{k-1}{\left(N_{TC}^{\text{seq}} + 1\right)} \right\rfloor \cdot \text{pubInt}^{\text{seq}} \\ &\quad + \text{mod}\left(k-1, N_{TC}^{\text{seq}} + 1\right) \cdot \left(\text{netInt}^{\text{seq}} + 10\right) \\ &\quad + T_{\text{AE}}^{\text{seq}}.\end{aligned} \tag{17}$$

G. Calculation of scanInt^{seq}

In this case, to optimize the power consumption, scanInt$^{\text{seq}}$ should be maximized. Thus, unlike in the previous section, the worst case scenario must consider the shortest possible gaps between TXs. Again, we present two examples to explain the procedure of deducting scanInt$^{\text{seq}}$.

If we first consider again a scenario where pubInt$^{\text{seq}}$ is long enough and $k \leq N_{TC}^{seq} + 1$, the example would be depicted as in Fig. 7(a). Note that when looking for the worst case, it is necessary to minimize scanInt$^{\text{seq}}$. Therefore, the random number generator should return its minimum value when applied. This makes that scenarios B and C share the same results. If we calculate the value of scanInt$^{\text{seq}}$ when $k = 1$, the result could be expressed as in

$$\begin{aligned}\text{scanInt}^{\text{seq}} = \text{netInt}^{\text{seq}} \cdot N_{TC} + \text{pubInt}^{\text{seq}} \cdot P_{RC} - T_{\text{AE}}^{\text{seq}}\\ + \text{scanWin}^{\text{seq}}.\end{aligned} \tag{18}$$

It may be easier to deduce its value starting from the right side of the figure. Then, considering that in this case scanWin$^{\text{seq}}$ follows

$$\text{scanWin}^{\text{seq}} = \text{pubInt}^{\text{seq}} - \text{netInt}^{\text{seq}} \cdot N_{TC}^{\text{seq}} + 10 + T_{\text{AE}}^{\text{seq}} \tag{19}$$

then the result is

$$\text{scanInt}^{\text{seq}} = \left(P_{RC}^{\text{seq}} + 1\right) \cdot \text{pubInt}^{\text{seq}} + 10. \tag{20}$$

Note that if k is increased, the gaps between TXs entering in the left scanWin$^{\text{seq}}$ now should consider that the random number associated with them must be maximum to reduce the gap at the right. This minimizes scanInt$^{\text{seq}}$. Note also that, for $k > N_{TC}^{\text{seq}} + 1$, it is necessary to suppose the same for the publish retransmissions that fall inside the scanWin$^{\text{seq}}$.

The second example depicted in Fig. 7(b) is valid in all cases where pubInt$^{\text{seq}} \leq$ netInt$^{\text{seq}} \cdot (N_{TC}^{\text{seq}} + 1)$, i.e., scenario A. In this case, the publish retransmissions are buffered. All the TXs are made every netInt$^{\text{seq}}$ except the ones that are inside the left scanWin$^{\text{seq}}$, which should add the maximum value of the random number generator. Then, for example, for $k = 1$, scanInt$^{\text{seq}}$ can be expressed as

$$\begin{aligned}\text{scanInt}^{\text{seq}} = \text{netInt}^{\text{seq}} \cdot N_{TC} + P_{RC} \cdot \text{netInt}^{\text{seq}} \cdot (N_{TC} + 1)\\ - T_{\text{AE}}^{\text{seq}} + \text{scanWin}^{\text{seq}}.\end{aligned} \tag{21}$$

The corresponding scanWin$^{\text{seq}}$ is shown in

$$\text{scanWin}^{\text{seq}} = \text{netInt}^{\text{seq}} + 10 + T_{\text{AE}}^{\text{seq}} \tag{22}$$

and the final result in

$$\text{scanInt}^{\text{seq}} = 10 + \text{netInt}^{\text{seq}} \cdot \left(N_{TC}^{\text{seq}} + 1\right) \cdot \left(P_{RC}^{\text{seq}} + 1\right). \tag{23}$$

Remember that for higher values of k all the gaps inside the scanWin$^{\text{seq}}$ should add 10 ms each. So, finally, the general equations for calculating scanInt$^{\text{seq}}$ are as follows.

1) Scenario A, (5):

$$\text{scanInt}^{\text{seq}} = 10 \cdot k + \text{netInt}^{\text{seq}} \cdot \left(N_{TC}^{\text{seq}} + 1\right) \cdot \left(P_{RC}^{\text{seq}} + 1\right). \tag{24}$$

2) Scenarios B and C, (6) and (7):

$$\begin{aligned}\text{scanInt}^{\text{seq}} = \left(P_{RC}^{\text{seq}} + 1\right) \cdot \text{pubInt}^{\text{seq}} + 10k \leq N_{TC}^{\text{seq}} + 1\\ \text{scanInt}^{\text{seq}} = \left(P_{RC}^{\text{seq}} + 1\right) \cdot \text{pubInt}^{\text{seq}}k > N_{TC}^{\text{seq}} + 1.\end{aligned} \tag{25}$$

IV. Energy Consumption

Now that we have completely presented our proposal, in this section we analyze the different consumption of the devices when they are transmitting or not, with or without applying our proposal.

A. Standard Mode

When working in standard mode, i.e., without applying our proposal, a device in standby but scanning continuously has a daily energy consumption (E_{stby}) in Joules of

$$E_{stby} = V \cdot I_{\text{scan}} \cdot T_{\text{day}} \tag{26}$$

where V is the working voltage, T_{day} is the duration of a day, and I_{scan} is the current consumption of the device under analysis while it is scanning.

If we consider N_{TX} TX events per day and we send N_{data} messages without errors, assuming $N_{TC} = R_{RC}$ the consumed energy (E_{std}) following the standard BLE mesh specifications is:

$$\begin{aligned}E_{std} = V \cdot \Big\{I_{\text{scan}} \cdot \Big[T_{\text{day}} - N_{TX} \cdot N_{\text{data}} \cdot \left(P_{RC}^{\text{data}} + 1\right)\\ \times \left(N_{TC}^{\text{data}} + 1\right) \cdot T_{AE}^{\text{data}}\Big] + I_{TX}\\ \times \Big[N_{TX} \cdot N_{\text{data}} \cdot \left(P_{RC}^{\text{data}} + 1\right) \cdot \left(N_{TC}^{\text{data}} + 1\right) \cdot T_{AE}^{\text{data}}\Big]\Big\}\end{aligned} \tag{27}$$

where I_{TX} is the current consumption of the device in a TX event. Nevertheless, note that in all the equations of this section we have considered that the current consumption during the duration of an advertising event (T_{AE}) is constant. Hence, I_{TX} should be averaged. It is possible to use tools like [34] to estimate this value for different TX power configurations of the devices. To determine E_{std} we calculate the consumed energy while the device is scanning and the consumed energy when the device is transmitting. In standard mode, the device remains always in one of these two states.

B. ADS Mode

When considering our proposal, it is necessary to differentiate between two cases: being in continuous scan because we saturate the system ($(T_{\text{day}} - T_{\text{ADS}} \cdot N_{TX}) \leq 0$) or not. Note that it only makes sense to apply this proposal when the system is not saturated. So, in that case, the consumed energy (E_{ADS}) is

$$\begin{aligned}E_{\text{ADS}} = V \cdot \Big[I_{\text{scan}} \cdot \Big\{T_{\text{ADS}} - \Big[\left(P_{RC}^{\text{seq}} + 1\right) \cdot \left(N_{TC}^{\text{seq}} + 1\right) \cdot T_{AE}^{\text{seq}}\\ + N_{\text{data}} \cdot \left(P_{RC}^{\text{data}} + 1\right)\\ \times \left(N_{TC}^{\text{data}} + 1\right) \cdot T_{AE}^{\text{data}}\Big]\Big\} \cdot N_{TX}\\ + I_{TX} \cdot \Big[\left(P_{RC}^{\text{seq}} + 1\right) \cdot \left(N_{TC}^{\text{seq}} + 1\right) \cdot T_{AE}^{\text{seq}} + N_{\text{data}}\\ \times \left(P_{RC}^{\text{data}} + 1\right) \cdot \left(N_{TC}^{\text{data}} + 1\right) \cdot T_{AE}^{\text{data}}\Big]\\ \times N_{TX} + I_{\text{scan}} \cdot \left[T_{\text{day}} - T_{\text{ADS}} \cdot N_{TX}\right] \cdot \frac{\text{scanWin}^{\text{seq}}}{\text{scanInt}^{\text{seq}}}\\ + I_{\text{idle}} \cdot \left[T_{\text{day}} - T_{\text{ADS}} \cdot N_{TX}\right]\\ \times \left(1 - \frac{\text{scanWin}^{\text{seq}}}{\text{scanInt}^{\text{seq}}}\right)\Big]\end{aligned} \tag{28}$$

where I_{idle} is the current consumption of the device while it is idle.

Similar to the previous case, we calculate the total consumed energy considering the different states of the devices. In saturation there are only two states, like in standard mode: scanning and transmitting both the data or the messages of the control sequence. However, if the system is not saturated, there are four consumption states: 1) scanning when there is data to be transmitted; 2) transmitting the BMADS sequence or the data; 3) scanning waiting for an ADS message; and 4) idle.

V, I_{scan}, I_{TX}, and I_{idle} can be found in the datasheets of the chipset manufacturers, for example, in [17], [18], or [35].

The energy overhead (E_{OVR}) introduced by our proposal can be calculated as (28) and (27) when this difference is positive. For example, if we consider the continuous scan case, this results in

$$E_{OVR} = V \cdot (I_{TX} - I_{\text{scan}}) \cdot N_{TX} \cdot \left(P_{RC}^{\text{seq}} + 1\right) \cdot \left(N_{TC}^{\text{seq}} + 1\right) \times T_{AE}^{\text{seq}}. \tag{29}$$

V. Discussion and Results

In order to evaluate the achievable gains of the ADS, different scenarios have been tested. The energy consumption of the network has been computed, and it will be expressed relative to a standard Bluetooth mesh network. That is to say, as a percentage of the continuous scan option. The evaluated scenarios differ in their traffic volume, measured in packet transmission (TX) per time unit, and in the network size, measured in the number of hops required in each TX and equal to the TTL (TTL = TTL$_{\text{ADS}}$). The proposal does not impose any restriction on the valid topology so any layout can be assumed and the network size is determined by the TTL, with one node acting as a source. Power consumption has been realistically considered and obtained from commercial off-the-shelf Bluetooth devices. These values can be read in Table II, in particular, we have the following.

1) The consumption data are extracted from [34] and validated experimentally using the nRF52840 DK.
2) The processing time at relay and sink nodes was also measured from the test bed.
3) The rest of the variable parameters are taken from the BLE mesh specifications and have been complemented with real measurements, as previously described.
4) Note that the standard indicates that various parameters can be configured with a range of values. These ranges are given in the table, indicating that all possible combinations have been simulated.

From the previous sections, it is learnt that scanning cycles are determined by using the Bluetooth mesh quadruplet that controls TX redundancy, $\{P_{RIS}, P_{RC}, N_{TIS}, N_{TC}\}$, or equivalently $\{\text{pubInt}, P_{RC}, \text{netInt}, N_{TC}\}$. The manipulation of these parameters allows to configure different scanning cycles with different degrees of energy saving. Scanning cycles also depend on the number k of control packets to be captured for redundancy purposes. Considering the different ranges of these parameters (see Table II), an important number of combinations can be configured. Hence, in order to provide a first insight on the achievable gains of the technique, the performance of all possible solutions has been computed. Results are initially plotted without redundancy. This means that scanning cycles must be able to capture $k = 1$ control packet. Consequently, the mesh network is also initially configured with a quadruplet such as data packets are transmitted once. The effect of increasing redundancy will be analyzed later on.

TABLE II
Common Parameters Used in Calculations

Current consumption at 8 dBm	16.4 mA
Current consumption receiving	4.60 mA
Current consumption in idle mode	0.01 mA
Working voltage	3 V
Processing time at relays and sink	20 μs
Scanned channels	37, 38 and 39
BMADS sequence parameters	
Publish retransmit count, P_{RC}^{seq}	$[0, 7]$ in steps of 1
Publish retransmit interval, $pubInt^{\text{seq}}$	$[50, 1600]$ ms in steps of 50
Network transmit count, N_{TC}^{seq}	$[0, 7]$ in steps of 1
Network transmit interval, $netInt^{\text{seq}}$	$[20, 320]$ ms in steps of 10
Relay retransmit count, R_{RC}^{seq}	N/A
Relay retransmit interval, $relInt^{\text{seq}}$	N/A
Time of advertising event, T_{AE}^{seq}	1.694 ms
Redundancy level, k	$[1, 8]$ in steps of 1
Data parameters	
Publish retransmit count, P_{RC}^{data}	0
Publish retransmit interval, $pubInt^{\text{data}}$	N/A
Network transmit count, N_{TC}^{data}	$k - 1 = [0, 7]$ in steps of 1
Network transmit interval, $netInt^{\text{data}}$	20 ms
Relay retransmit count, R_{RC}^{data}	$k - 1 = [0, 7]$ in steps of 1
Relay retransmit interval, $relInt^{\text{data}}$	20 ms
Time of advertising event, T_{AE}^{data}	1.694 ms

Given this, Fig. 8 shows the relative energy consumption with respect to a standard Bluetooth mesh operation. This is done versus the maximum potential extra delay that TXs might experience. Note that such extra delay may represent a higher or lower percentage of the total TX time. It depends on the volume of data that is transmitted in each TX event. Results for two sample scenarios are plotted in different colors, where each point represents the performance of a particular quadruplet. The yellow points indicate a particular solution and show how its performance changes under different conditions.

The light blue case is a small mesh network in which any source of information can reach its destination in four hops (TTL = 4). The traffic is also low, the source generates just 1 TX/day. Notice that, among all possible configurations, the non dominated ones have been outlined. That is to say, solutions in which the energy consumption cannot be further improved without degrading the maximum delay. Such solutions are Pareto optimal and constitute the configurations to

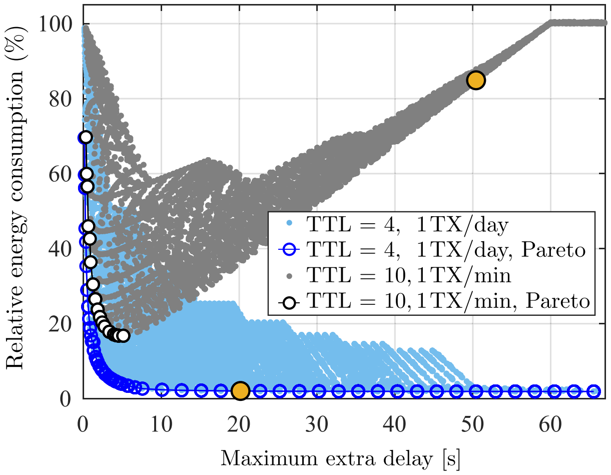


Fig. 8. Performance of all configurations for two different scenarios ($k = 1$).

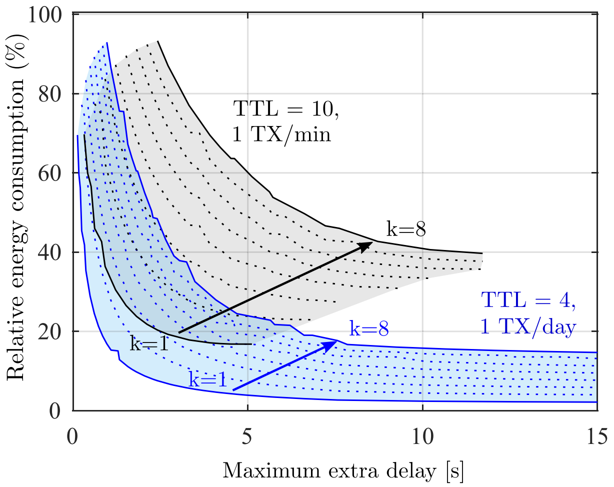


Fig. 9. Pareto optimal solutions for increasing redundancy level.

consider. The others always represent a worse compromise. In this case, there is an obvious trade off between energy consumption and delay. Indeed, the configuration that introduces more potential delay is the one that allows more energy savings. However, it is remarkable that the relative energy consumption reduces fast and sharply to 4.6% with just a potential delay of 4 s. The consumption can be as tiny as 1.92%, meaning more than 98% of energy savings with respect to a standard Bluetooth mesh operation. However, TXs might increase its delay up to 65 s. Note that for a node having a battery of 4400 mAh, the operational life would be extended from 39.8 days to 5.7 years. Consequently, we can conclude that the approach is a very interesting option to achieve power efficient Bluetooth mesh networks.

From the plot, it may seem that the energy gains reach a plateau at around 10 s of delay. However, the relative energy consumption at that point equals 2.38%, noticeable higher than the best case of 1.92% (98.08% gain), which would imply a lifetime increase of more than one additional year. This result is interesting for applications where energy preservation is really critical and are very elastic to the delay.

It can be anticipated that traffic volume and network size impact on the performance of the different quadruplet configurations. In fact, the gray scenario corresponds to a 10 hop sized network generating 1 TX/min. Still, the energy consumption could be as low as 16.78% of a Bluetooth mesh network without the proposed mechanism. However, the reduction is apparent in the number of Pareto-optimal solutions. Indeed, it can be seen that the trade off is not maintained in the full range of delays and there is a point at which both energy and delay are simultaneously degraded. However, solutions with the lowest delays are always present.

As seen in the previous sections, the scan window grows with the desired level of redundancy. Thus, it is logical to anticipate that both energy saving and delay degrade if the parameter k is increased. Fig. 9 quantifies such degradation in the two scenarios. It represents the Pareto fronts for k ranging from 1 (no redundancy) to 8. This means that just the optimum parameter combinations are represented by different points. Comparing both cases, increasing redundancy has a higher impact in large scenarios with higher traffic. This is because continuous scanning is kept active for a longer time when the network is larger and such reconfiguration happens with every new TX. This means that the effect of network size and traffic volume is multiplicative. Note that the redundancy has been swept until $k = 8$. This means that 8 repeated packets must be captured in each radio link by all relays and sink nodes. Despite being such a pessimistic scenario, important energy savings are still observed with respect to normal Bluetooth mesh operation. Without loss of generality, subsequent plots are presented for $k = 1$.

As traffic increases, the mechanism logically tends to a continuous scan option. Therefore, the Pareto front collapses into that unique solution. At this point, it would make no sense to use the approach. The control information would not be necessary and would simply imply unnecessary overhead and energy consumption.

Given these considerations, Fig. 10 compares the optimal solutions for increasing traffic volume and network size. Therefore, the points in the plot just represent the combination of parameters that belong to the set of non dominated solutions. Note that in the very worst case, 4 TX/min happen and a packet experiences 127 hops between the generator and the sink. This is the maximum TTL for a packet in current Bluetooth mesh networks. Even under these circumstances, it is possible to find a configuration that drops the energy consumption to 74%. Gains are almost identical in the range from 1 TX/day to 1 TX/15 min. It is also very evident how the Pareto fronts are displaced up and to the left when the network size increases. This is because once a device needs to transmit a packet, all nodes participating in the TX change their scanning behavior. The new temporary scan window is proportional to the number of hops, meaning more energy consumption. And also, because the TXs are delayed a longer time.

From the previous discussion, we can conclude that different scenarios can be optimized by different quadruplet sets. Besides, a solution that was optimal for one situation can become a non interesting one for another. This was already

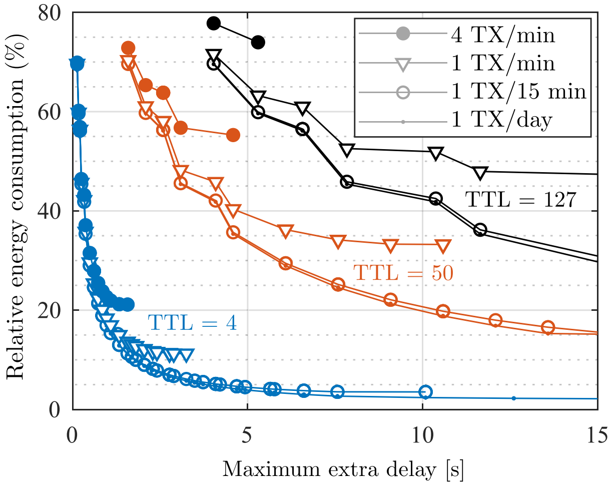


Fig. 10. Optimal solutions for different traffic volumes and network sizes.

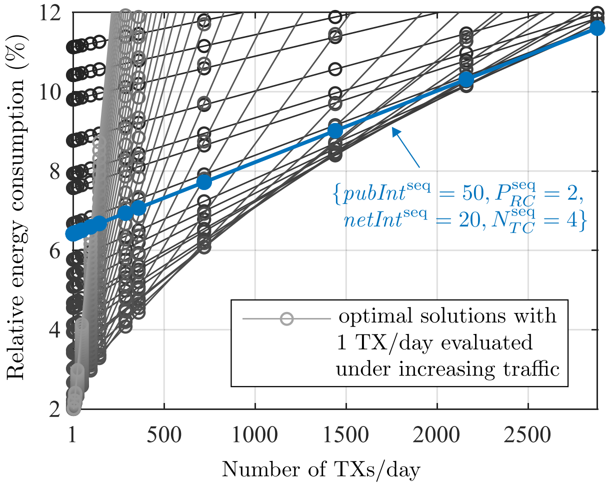


Fig. 12. Performance sensitivity to variations in traffic (zoomed in).

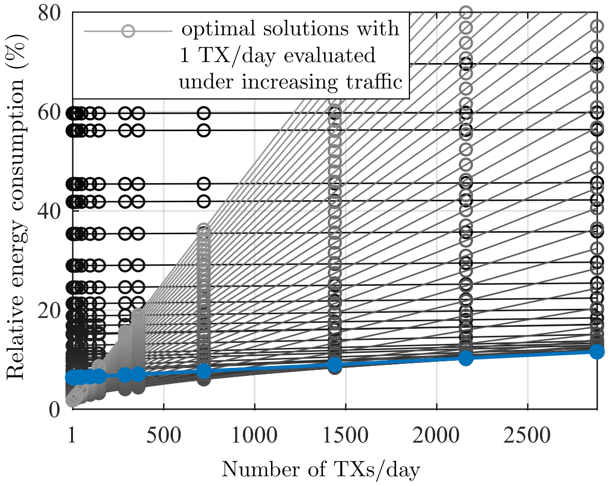


Fig. 11. Performance sensitivity to variations in traffic volume.

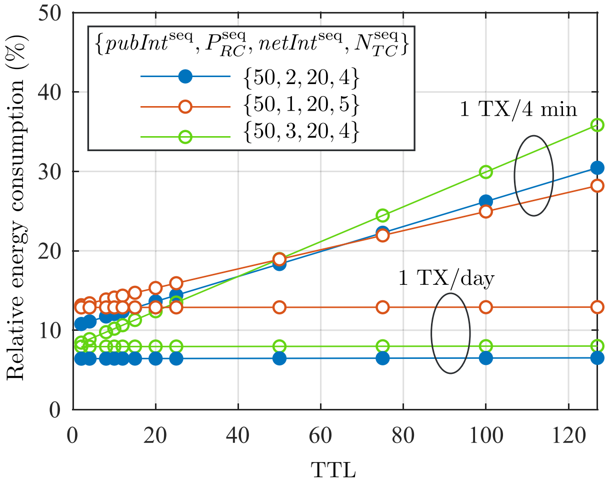


Fig. 13. Performance sensitivity to variations in the network size.

outlined in Fig. 8, where we could observe the position of a certain configuration in the two scenarios. Whereas it was optimal in the low traffic one, it would be a bad choice in the second case.

From the previous comments, it seems necessary to evaluate the sensitivity of our proposal to changing traffic conditions. We take the Pareto optimal solutions with the lowest traffic (1 TX/day). Next, we evaluate how their performance changes with increasing traffic volume. This is represented in Fig. 11, where each line represents the relative energy consumption of a particular quadruplet. For the sake of clarity, different tones of gray have been used in the different lines and Fig. 12 is a supporting zoom of the lower part. It can be seen how different solutions show very different dependency on traffic. It is revealed that certain configurations can be very interesting for sporadic traffic conditions. However, these settings are very sensitive to traffic variations and become the worst option when traffic increases. On the other hand, there are cases in which the gains are barely altered. Hence, two implementation paths are possible. The first one is to choose a sub optimal configuration not being sensitive to traffic variations. An example is outlined in the plot, where the quadruplet $\{\mathrm{pubInt}^{\mathrm{seq}} = 50\text{ ms},\ P_{RC}^{\mathrm{seq}} = 2,\ \mathrm{netInt}^{\mathrm{seq}} = 20\text{ ms},\ N_{TC}^{\mathrm{seq}} = 4\}$ ranges from 6.4% to 11.6% in relative energy consumption. This is a very interesting solution and could be taken as an all-purpose rule-of-thumb for non optimized plug-and-play configurations. The second option is to implement a self-organized mesh network that dynamically configures the quadruplet based on traffic status. Nevertheless, note that the traffic variations represented in the plot are considerable, from 1 TX/day to 2 TX/min. Therefore, if the fluctuations happen in a narrower range, the first approach could perform optimally in all cases.

It has already been argued that the performance of the configurations is also dependant on the network size. The optimal solutions should be computed considering both the traffic and the number of relays participating in the TX. Thus, we also analyze the sensitivity of the performance to variations in the number of hops. This is represented in Fig. 13 for three quadruplets, including the previous $\{50, 2, 20, 4\}$. Two traffic situations are plotted, with low and high traffic activity. From the figure, it can be seen that the performance remains stable for any number of relays as long as the traffic is low. In fact, this is true up to around 1 TX/15 min. When the traffic is

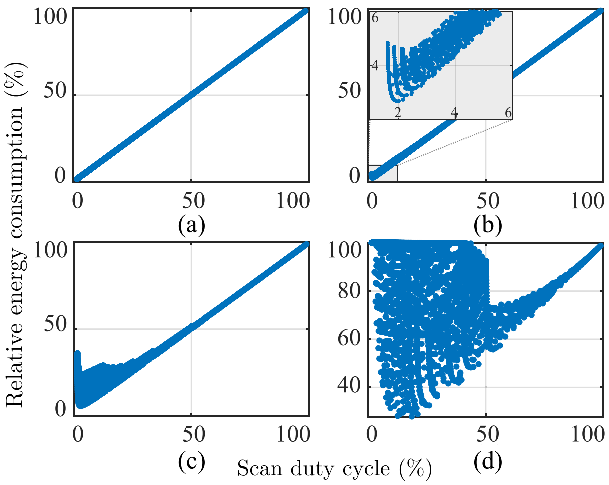


Fig. 14. Impact of scan duty cycle in the relative energy consumption. (a) TTL = 3, 1 TX/day. (b) TTL = 127, 1 TX/day. (c) TTL = 3, 1 TX/4 min. (d) TTL = 127, 1 TX/4 min.

higher, different levels of sensitivity arise. It can be seen how {50, 1, 20, 5} remains more stable with the number of relays at the cost of performing worse in small networks. Nevertheless, it can be thought that the network size is expected to be rather static in the short term. Thus, it seems wise to choose a solution optimized for a particular network deployment rather than a sub optimal all-purpose one.

To end the discussion, we dive into the behavior of the different solutions. Fig. 14 shows the relationship between the scan duty cycle and the power consumption relative to the continuous scan. In this case, each point represents a feasible quadruplet for four scenarios having low/high traffic and low/high TTL. This allows us to analyze the correlation between energy consumption and duty cycle. It can be seen that when the traffic and the size of the network are small [subplot (a)], there is a linear relationship between both variables. In other words, in this scenario, the most interesting option is to use the shortest work cycle possible. This corresponds to solutions with high energy saving but also high delay, as already shown in Fig. 8 and meaning solutions with high $\text{pubInt}^{\text{seq}}$ and/or $\text{netInt}^{\text{seq}}$. However, the specific values in the Pareto front cannot be anticipated and need to be computed. As already discussed, they depend on the traffic and network features. It is also important to recall that there is not a unique *best* solution but a set of optimal ones. Thus, the configuration should be adapted to the particular needs of the application in terms of energy saving and allowable delay. When the traffic volume and network size increase [subplots (b) and (c)], the smallest duty cycles start to be less interesting. For networks with many hops and traffic [subplot (d)], a total loss of correlation can be observed for the lower duty cycles. Indeed, the solutions with the smallest duty cycle become as bad as the ones with large cycle. Finally, it is logical that solutions with high duty cycles do always perform badly in terms of energy consumption.

VI. Conclusion

The recent Bluetooth mesh specifications rise as an interesting enabler for IIoT and other low-energy sensor-based applications. Its managed flooding strategy provides a robust TX mechanism but requires that several nodes scan the channel continuously. This is very inefficient in terms of energy consumption. Therefore, this article presents a novel strategy to reduce the overall consumption. This is done by using dynamic scanning cycles while still ensuring the network reliability. To make this possible, we introduce a new control message sequence that puts the nodes into a continuous scan and prepares the network prior to data TX. This proposal has the added benefit that can be introduced as a new feature in the current standard.

The results show that the approach provides very power efficient Bluetooth mesh networks. The strategy has been assessed in scenarios with different characteristics and it shows different levels of energy gain in all cases. There is not a single optimal configuration for the mechanism but a set of non dominated solutions to choose from. This range of solutions shows a compromise between the energy saving and the additional delay introduced. For applications that can tolerate delays of tens of seconds, substantial energy savings have been observed, over 98%. Therefore, it is possible to extend a 4400-mAh-powered node from less than 40 days to more than 5.7 years. Extreme cases of networks requiring 127 hops and having very high traffic can still benefit of the mechanism and show energy savings of 25%. The results also indicate that it is possible to deploy the mechanism with the level of redundancy required by the propagation and interference conditions. It is also concluded that the solutions that have the lowest duty cycles are very interesting in situations of sporadic traffic.

The results indicate that it is possible to find all-purpose configurations able to accommodate varying traffic volumes. Since they would act in a sub optimal manner, an interesting future work is to investigate how to extend this approach to deploy a self-optimized Bluetooth mesh network. This extension should pursue a dynamic adaptation of the parameters to varying traffic conditions.

References

[1] L. Chettri and R. Bera, “A comprehensive survey on Internet of Things (IoT) toward 5G wireless systems,” *IEEE Internet Things J.*, vol. 7, no. 1, pp. 16–32, Jan. 2020.

[2] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, “Internet of Things: A survey on enabling technologies, protocols, and applications,” *IEEE Commun. Surveys Tuts.*, vol. 17, no. 4, pp. 2347–2376, 4th Quart., 2015.

[3] S. A. Malek, S. D. Glaser, and R. C. Bales, “Wireless sensor networks for improved snow water equivalent and runoff estimates,” *IEEE Access*, vol. 7, pp. 18420–18436, 2019.

[4] G. A. Akpakwu, B. J. Silva, G. P. Hancke, and A. M. Abu-Mahfouz, “A survey on 5G networks for the Internet of Things: Communication technologies and challenges,” *IEEE Access*, vol. 6, pp. 3619–3647, 2017.

[5] C. Perera, C. H. Liu, S. Jayawardena, and M. Chen, “A survey on Internet of Things from industrial market perspective,” *IEEE Access*, vol. 2, pp. 1660–1679, 2014.

[6] J. Weiss. (2015). *Wireless Sensor Networking for The Industrial IoT*. [Online]. Available: www.linear.com/smartmesh

[7] J. M. Williams *et al.*, “Weaving the wireless Web: Toward a low-power, dense wireless sensor network for the industrial IoT,” *IEEE Microw. Mag.*, vol. 18, no. 7, pp. 40–63, Nov./Dec. 2017.

[8] D. Raposo, A. Rodrigues, S. Sinche, J. Sá Silva, and F. Boavida, “Industrial IoT monitoring: Technologies and architecture proposal,” *Sensors*, vol. 18, no. 10, p. 3568, Oct. 2018. [Online]. Available: http://www.mdpi.com/1424-8220/18/10/3568

[9] S. Jacobs. (2013). *Power Your Wireless Sensors for 40 Years*. [Online]. Available: https://www.electronicdesign.com/power-management/article/21795789/power-your-wireless-sensors-for-40-years

[10] *Saft Batteries for The Internet of Things*, Saft Batteries, Levallois-Perret, France, 2019. [Online]. Available: https://bit.ly/3eMbNx8

[11] A. Kutyła-Olesiuk, M. Zaborowski, P. Prokaryn, and P. Ciosek, "Monitoring of beer fermentation based on hybrid electronic tongue," *Bioelectrochemistry*, vol. 87, pp. 104–113, Oct. 2012.

[12] T. Ojha, S. Misra, and N. S. Raghuwanshi, "Wireless sensor networks for agriculture: The state-of-the-art in practice and future challenges," *Comput. Electron. Agr.*, vol. 118, pp. 66–84, Oct. 2015.

[13] G. Hornero *et al.*, "A wireless augmentative and alternative communication system for people with speech disabilities," *IEEE Access*, vol. 3, pp. 1288–1297, 2015.

[14] E. Sifuentes, O. Casas, and R. Pallas-Areny, "Wireless magnetic sensor node for vehicle detection with optical wake-up," *IEEE Sensors J.*, vol. 11, no. 8, pp. 1669–1676, Aug. 2011.

[15] *Mesh Profile Bluetooth Specification V1.0.1*, Bluetooth SIG, Kirkland, WA USA, 2019. [Online]. Available: https://www.bluetooth.com/specifications/mesh-specifications/

[16] *Bluetooth Core Specification 5.1*, Bluetooth SIG, Kirkland, WA USA, 2019. [Online]. Available: https://www.bluetooth.com/specifications/bluetooth-core-specification/

[17] *nRF52840 Product Specification V1.1*, Nordic Semicond., Trondheim, Norway, 2019. [Online]. Available: https://infocenter.nordicsemi.com/pdf/nRF52840_PS_v1.1.pdf

[18] *nRF52832 Product Specification V1.4*, Nordic Semicond., Trondheim, Norway, 2017. [Online]. Available: https://infocenter.nordicsemi.com/pdf/nRF52832_PS_v1.4.pdf

[19] 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

[20] *nRF51422—Bluetooth Low Energy, ANT and 2.4 GHz SoC—Nordicsemi.com*, Nordic Semicond., Trondheim, Norway, 2020. [Online]. Available: https://www.nordicsemi.com/Products/Low-power-short-range-wireless/nRF51422

[21] N. Todtenberg and R. Kraemer, "A survey on Bluetooth multi-hop networks," *Ad Hoc Netw.*, vol. 93, Oct. 2019, Art. no. 101922.

[22] 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.

[23] 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 Int. Symp. Pers. Indoor Mobile Radio Commun. (PIMRC)*, Montreal, QC, Canada, Oct. 2017, pp. 1–6.

[24] P. Levis, P. Clausen, J. Hui, O. Gnawali, and J. Ko, "The trickle algorithm," Internet Eng. Task Force, RFC 6206, 2011. [Online]. Available: https://tools.ietf.org/html/rfc6206

[25] *FruityMesh—The First Completely Connection-Based Open Source Mesh on Top of Bluetooth Low Energy (4.1/5.0 or higher)*, M-Way Solutions GmbH, Stuttgart, Germany, 2019. [Online]. Available: https://github.com/mwaylabs/fruitymesh

[26] 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.

[27] X. Wang, K. Xu, and B. Mao, "GreenLink: An energy efficient scatternet formation for BLE devices," *Wireless Commun. Mobile Comput.*, vol. 2018, Nov. 2018, Art. no. 1836198. [Online]. Available: https://doi.org/10.1155/2018/1836198

[28] R. T. E. Dvinge, A. Stalmach, and L. Nalpantidis, "Connection-based Bluetooth mesh network as a low energy solution for off-grid data networks," in *Proc. 8th Int. Conf. Mod. Circuits Syst. Technol. (MOCAST)*, Thessaloniki, Greece, May 2019, pp. 1–6.

[29] A. Bęben, A. Bąk, and M. Sosnowski "Efficient relay node management in BLE MESH networks," *Int. J. Electron. Telecommun.*, vol. 66, no. 1, pp. 29–35, 2020. [Online]. Available: http://www.ijet.pl/index.php/ijet/article/view/10.24425-ijet.2019.130262

[30] C. Cano, B. Bellalta, A. Sfairopoulou, and M. Oliver, "Low energy operation in WSNs: A survey of preamble sampling MAC protocols," *Comput. Netw.*, vol. 55, no. 15, pp. 3351–3363, Oct. 2011.

[31] J. Nieminen, T. Savolainen, M. Isomaki, B. Patil, Z. Shelby, and C. Gomez, "IPv6 over BLUETOOTH (R) low energy," Internet Eng. Task Force, RFC 7668, 2015. [Online]. Available: https://tools.ietf.org/html/rfc7668

[32] B. Luo *et al.*, "Neighbor discovery for IPv6 over BLE mesh networks," *Appl. Sci.*, vol. 10, no. 5, p. 1844, Mar. 2020. [Online]. Available: https://www.mdpi.com/2076-3417/10/5/1844

[33] D. Perez Diaz de Cerio, A. Hernandez, J. L. Valenzuela, and A. Valdovinos, "Analytical and experimental performance evaluation of BLE neighbor discovery process including non-idealities of real chipsets," *Sensors (Switzerland)*, vol. 17, no. 3, p. 499, 2017.

[34] *Online Power Profiler*, Nordic Semicond., Trondheim, Norway, 2020. [Online]. Available: https://devzone.nordicsemi.com/nordic/power

[35] *CC2650MODA SimpleLink^{TM} Datasheet*, Texas Instrum., Dallas, TX, USA, 2018. [Online]. Available: http://www.ti.com/lit/ds/symlink/cc2650moda.pdf

D. Perez-Diaz-de-Cerio received the M.Sc. degree in telecommunications engineering and the Ph.D. degree in telecommunication from the Technical University of Catalonia, Barcelona, Spain, in 2003 and 2010, respectively.

In 2003, he joined the WiComTec Research Group, Signal Theory and Communications Department, Universitat Politècnica de Catalunya, Castelldefels, Spain, as a Collaborating Lecturer. All his lectures are held with the Escola d'Enginyeria de Telecomunicació i Aeroespacial de Catalunya, Castelldefels, Spain, for Bachelor's degree, second-cycle and MAST master students. He has participated as a consultant in several local and European projects in addition to other public and private funded projects. His research interests are wireless communications systems, especially those based on the IEEE 802.X standards and their use in e-health and IoT.

J. L. Valenzuela received the M.Sc. degrees in telecommunication engineering and the Ph.D. degree from the Universitat Politècnica de Catalunya (UPC), Barcelona, Spain, in 1993 and 1997, respectively.

He is currently an Associate Professor with the Signal Theory and Communications Department, UPC. He was concerned with equalization techniques for digital systems. He has also been working on the field of digital communications with particular emphasis on digital radio and its performance under multipath propagation conditions. His research interests are in the field of wireless sensor networks and wireless communications systems.

Mario Garcia-Lozano received the Ph.D. degree in telecommunications engineering from the Universitat Politècnica de Catalunya (UPC), Barcelona, Spain, in 2009.

He has 20 years of experience in different radio network planning and optimization activities both at academia and industry. He is currently an Associate Professor with UPC. He has actively participated in 25 competitive research projects and contracts with the industry. His research activities are focused in the field of radio resource management and the optimization wireless networks.

Dr. Garcia-Lozano is a recipient of three best paper awards and was the advisor of the student team that won the international competition for mobile network planning organized by the company ATDI.

Á. Hernández-Solana received the M.Sc. degree in telecommunications engineering and the Ph.D. degree from the Universitat Politècnica de Catalunya (UPC), Barcelona, Spain, in 1997 and 2005, respectively.

She has been working with UPC and the University of Zaragoza, Zaragoza, Spain, where she has been an Associate Professor since 2010. She is a member of the Aragón Institute of Engineering Research, UPC. Her research interests include 5G/4G technologies, heterogeneous communication networks and mission-critical communication networks, with emphasis on transmission techniques, radio resource management and quality of service, mobility management and planning, and dimensioning of mobile networks.

A. Valdovinos received the M.Sc. degree in telecommunications engineering and the Ph.D. degree from the Universitat Politècnica de Catalunya (UPC), Barcelona, Spain, in 1990 and 1994, respectively.

He has been working with UPC and the University of Zaragoza, Zaragoza, Spain, where he has been a Full Professor since 2003. His research interests include 5G/4G technologies, heterogeneous communication networks, and mission-critical communication networks, with emphasis on transmission techniques, radio resource management and quality of service, mobility management and planning, and dimensioning of mobile networks.