



**HAL**  
open science

## A new Dynamic Multicores Neighbors Discovery approach in BLE for Low Power Systems

Damien Wohwe Sambo, Nathalie Mitton, Robin Delafaitte, Laurent Clavier,  
Rédha Kassi

► **To cite this version:**

Damien Wohwe Sambo, Nathalie Mitton, Robin Delafaitte, Laurent Clavier, Rédha Kassi. A new Dynamic Multicores Neighbors Discovery approach in BLE for Low Power Systems. International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob), Oct 2024, Paris, France. hal-04684912

**HAL Id: hal-04684912**

**<https://hal.science/hal-04684912v1>**

Submitted on 3 Sep 2024

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# A new Dynamic Multicores Neighbors Discovery approach in BLE for Low Power Systems

Damien Wohwe Sambo

*ARTS group, CERI-SN*

*IMT Nord Europe*

Lille, France

damien.wohwe-sambo@imt-nord-europe.fr

Nathalie Mitton

*FUN team*

*INRIA*

Lille, France

nathalie.mitton@inria.fr

Robin Delafait

*CNRS, UMR 8520 - IEMN*

*University of Lille*

Lille, France

robin.delafait@imt-nord-europe.fr

Laurent Clavier

*ARTS group, CERI-SN; CNRS, UMR 8520 - IEMN*

*IMT Nord Europe; University of Lille*

Lille, France

laurent.clavier@imt-nord-europe.fr

Rédha Kassi

*IRCICA - USR-3380*

*University of Lille*

Lille, France

redha.kassi@univ-lille.fr

**Abstract**—Due to their limited resources, IoT devices are constantly striving to find a balance between energy consumption and performance. In this ongoing endeavor, the communication module frequently emerges as the foremost contributor to energy consumption. Among various communication technologies, Bluetooth Low Energy (BLE) stands out as a promising candidate due to its lightweight architecture and excellent performance. Despite recent advances in BLE optimization, the Neighbor Discovery Process (NDP) remains a highly energy-cost task. To enhance energy efficiency during this operation, we propose a new approach for the NDP that leverages the multicore architecture of a recent microcontroller to dynamically adjust the scanning duration based on the number of devices to discover. Intensive simulations have shown that a fixed scanning duration can be avoided by ending the NDP after discovering the required number of devices. Our proposal demonstrates a reduction in energy consumption through intensive simulations and a robust validation methodology. Furthermore, real-world experiments indicate that our approach can achieve up to 50% energy savings compared to a 1-second scan when only one core is used during the NDP.

**Index Terms**—Internet of Things (IoT), energy efficiency, Bluetooth Low Energy (BLE), Neighbor Discovery Process, and Symmetric MultiProcessing.

## I. INTRODUCTION

Over the past decade, the Internet of Things (IoT) has experienced a notable surge in popularity, primarily propelled by its versatile applications and the progression of micro-electronics. A typical IoT network consists of a multitude of low-power devices adept at collecting and transmitting data via lightweight wireless communication technologies. These devices, often powered by limited resources like batteries, need to be compact and cost-effective. Consequently, IoT devices are constrained to basic operations, including gathering physical parameters with sensors, performing fundamental computations using microcontrollers, and transmitting or receiving

data through communication modules. Due to their restricted resources, IoT devices face challenges, particularly in energy consumption, with communication modules, especially during transmission or listening mode. Among the various wireless technologies, Bluetooth Low Energy (BLE) stands out due to its low energy consumption and high-quality service, offering features like low latency and high data rates of up to 1 Mbps or 2 Mbps [1]. However, before any BLE communication, a transmitter must identify receivers through the Neighbor Discovery Process (NDP). Multiple beacons may be exchanged between nodes during this process, leading to increased energy consumption. Various research efforts aim to optimize energy consumption during NDP by identifying optimal parameters. However, the challenge lies in the fact that these parameters may not be universally applicable due to variations in the number of available devices in different scenarios.

Several works aim to optimize the NDP parameters to minimize the overall latency during the NDP [2]–[7]. Although “*optimized values*” proposed in recent papers, the NDP remains highly susceptible to various unpredictable parameters, including the number of nodes within the discovery area, cross-technology interferences etc. Consequently, depending on the environmental conditions (real scenarios), the efficacy of the “*optimized values*” for NDP may yield diverse outcomes.

To address the challenge posed by fixed parameters in the NDP, particularly when the required number of devices to discover is known, we present a novel approach that considers the Symmetric MultiProcessing (SMP) architecture of the microcontroller used in IoT. The core concept is to dynamically adjust the overall scanning duration based on the number of devices to be discovered. The multiple cores of the microcontroller are leveraged to parallelize the scanning process, executing two distinct tasks simultaneously. The first

task involves the scanning process itself, while the second task, executed by another core, checks if the required number of devices has been found. Communication between the cores is facilitated through a shared memory space. Contrary to a single core architecture that can implement multithreading, the SMP architecture can allow a core to dynamically halt another core on demand through the utilization of high-priority instructions. Our proposal demonstrates its capability to achieve more than a 50% reduction in energy consumption compared to a 1-second scan when only one core is dedicated for the NDP. Moreover, intensive simulations highlight the limitations of a fixed scanning duration by revealing that all devices can be discovered before the predetermined scanning duration, with variations based on the number of available devices within the communication range.

The rest of the paper is organized as follows: Section II provides useful background on NDP; related works are presented in Section III; our approach is described in Section IV; its evaluation is presented in Sections V and VI; Section VII concludes with future works.

## II. NEIGHBORS DISCOVERY PROCESS IN BLE

During the Neighbor Discovery Process (NDP), each advertiser periodically transmits advertisement messages (Protocol Data Unit PDU) over the different advertising channels (e.g., Fig. 1 describes the discovery process in BLE 4, where there are three dedicated advertising channels 37, 38, and 39). Following this transmission, a random delay is generated. On the receiving end, the scanner sequentially monitors each advertising channel. Two types of events may occur: (i) Direct advertising, which is exclusive to devices already known (having undergone a successful NDP before); (ii) Undirected events, which involve unknown devices. In Bluetooth Low Energy (BLE), undirected NDP can occur through either a connectable or a non-connectable manner. In connectable discovery, the scanner can respond to the advertiser after receiving an advertisement. This leads to the establishment of a master/slave connection between the scanner and the advertiser. However, in non-connectable discovery, no connection request is initiated during advertising. Consequently, the scanner only receives advertisements without the ability to interact with the advertiser [8].

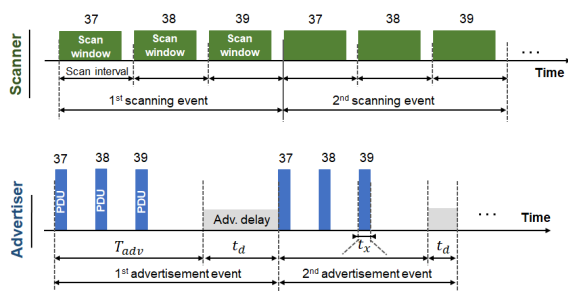


Fig. 1: Standard process of the BLE Neighbour Discovery (with 3 dedicated advertising channels)

## III. RELATED WORKS ON NDP OPTIMIZATION IN BLE

This section presents the related works on NDP optimization of the BLE.

Wei et al. [9] proposed a novel deterministic and one-way Neighbor Discovery Process (NDP) named Lightning. To reduce discovery delay, the authors assumed that in each period, only a few active slots exist, with the remaining time designated as sleep slots. Based on a duty cycle, the proposal determines the optimal number of periods required to achieve the NDP. The authors minimized the duration of the active slot to enable the transmission of a single packet, ensuring the discovery of a neighbor in a one-way communication model. To address collision probability and overall latency issues in BLE's discovery process, Yang et al. [10] introduced TCWS, incorporating a novel wait slot scheme in a two-way communication during NDP. TCWS replaces the three-way handshake communication of standard BLE NDP with a two-way approach, ignoring the advertiser's response (*SCAN\_RSP*) after receiving the scanner's scan request (*SCAN\_REQ*). This reduces collision probability in dense BLE networks by distributing *SCAN\_REQ* packets through the new wait slot. Jeon et al. [2] conducted a mathematical analysis of certain NDP parameters to minimize NDP delay. They assumed equal lengths for advertising events and scan windows, deriving latency and energy consumption metrics using the Chinese Remainder Theorem (CRT). Shan et al. [5] focused on minimizing the advertisement interval, concluding that it should be a multiple of 0.625 ms, ranging from 20 ms to 10.24 s for any advertisement delay between 0-10 ms. They compared the optimal advertisement interval obtained through mathematical analysis with simulation results. Luo et al. [4] conducted an advanced study to optimize NDP parameters according to IoT applications, similar to Jeon et al. [2]. Their solution, based on CRT, identified critical metrics affecting NDP performance (AdvInterval, ScanInterval, and ScanWindow) and determined optimal values. They expressed the average delay of NDP and resulting energy consumption.

Similar to the previous work, recent research aims to evaluate and improve the advanced-NDP introduced in Bluetooth V5 [11] by looking for the optimized values of NDP parameters [3], [5], [12].

We can observe from these approaches (TABLE I) that the primary focus is on optimizing BLE parameters within the Neighbor Discovery Process (NDP) to reduce the overall duration and, consequently, lower energy consumption. However, in each presented work, the scanning duration is ultimately fixed with "optimized" values. Nonetheless, only a few studies, such as the works of Mikhaylov et al. [14] and Djidi et al. [15], have proposed considering additional components like a Wake-up Radio (WuR) in enhancing the NDP. Furthermore, to the best of our knowledge, there is no published solution that utilizes the microcontroller's hardware architecture to dynamically adapt the scanning duration based on the required number of nodes

TABLE I: Comparison of some existing studies on NDP in BLE

Works	Contribution	Version	Year
[10]	Replacing the 3-step communications of standard BLE by a two-way approach during the NDP.	BLE4	2017
[2]	Optimization of the NDP parameters based on the Chinese Remainder Theorem.	BLE4	2017
[3] & [4]	Optimization of the NDP parameters	BLE4	2019
[5]	Optimization of the Advanced-NDP parameters.	BLE5	2020
[13]	Analytical analysis of the multichannel NDP.	BLE4	2021
[6]	Optimization of parameter sets based on any specified duty cycle.	Any	2022
[7]	Analytical analysis of the relationship between NDP parameters and the overall latency.	Any	2023
Proposal	Dynamic adaptation of the NDP at lower levels. Compatible with all optimization parameters.	Any	-

to be discovered

#### IV. PROPOSED APPROACH

1) *Problem statement and assumptions:* The NDP in IoT poses a significant challenge due to the device’s resource constraints. Indeed, the scanner tends to consume excessive energy during this process. In the subsequent sections, we introduce a novel approach that dynamically adjusts the scanning duration according to the number of nodes needed to be discovered. our approach is based on the SMP architecture, which is widely used in microcontrollers on the market.

In our proposal, we assume that :(i) the microcontroller is based on the SMP architecture with at least two cores; (ii) the IoT device uses at least the BLE 4 (the first release of BLE).

Our proposal relies on a multicore architecture rather than a simple multithreaded approach. This choice stems from the limitation inherent in multithreaded architectures, where a thread lacks the capability to halt another running thread on demand. In contrast, interaction between different cores is feasible within a multicore architecture, particularly when they are based on the SMP architecture.

2) *Multicores NDP for a dynamic scanning:* Energy efficiency in the IoT remains the main challenge when designing an application. The IoT node should consume as little as possible to extend its lifetime. Recent works showed that during NDP in BLE, the scanning process consumes more energy than necessary due to the fixed duration of the scan [15]. To reduce the power consumption of the BLE module during the NDP, we propose a new approach that dynamically adjusts the scanning duration by exploiting the microcontroller’s SMP architecture (Fig. 2).

Thanks to the SMP architecture, executing several tasks in parallel is easily possible. During the scan action of the NDP, we can identify two independent tasks: scanning and checking

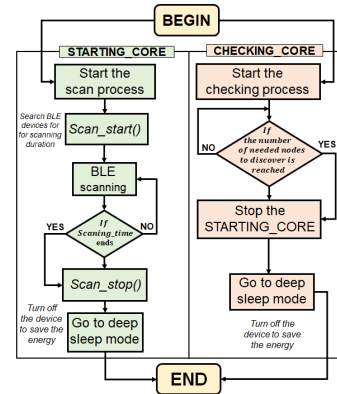


Fig. 2: Flow chart describing the functioning of the proposed approach for the dynamic scanning time during NDP

processes. As we can see in Fig. 2, we use two cores of the microcontroller to adapt the scanning duration during the NDP dynamically. The *STARTING\_CORE* initializes the scanning process by calling the blocking function *Scan\_start()*. At the same time, the *CHECKING\_CORE* starts by looking for the number of newly discovered devices. The exchange between the two cores is allowed by a shared variable *found\_nodes* between them. Thus, while the *STARTING\_CORE* looks for new devices, the *found\_nodes* variable is incremented and the *CHECKING\_CORE* checks the number of the found nodes according to the needed number (input). If the needed number of nodes to be found is reached before the end of the fixed scanning duration, the *CHECKING\_CORE* will stop the scanning core (*STARTING\_CORE*) after saving the different found nodes in non-volatile memory. In this case, the scanning duration will be reduced according to the number of different nodes needed to be found. The NDP duration is then reduced, thus minimizing the scanner’s energy consumption. However, if the number of nodes to be found is not reached before the end of the scanning duration, the (*STARTING\_CORE*) will automatically stop its process without any interruption. the scanning process will normally finish after the fixed duration.

#### V. EXPERIMENTATIONS AND SIMULATION

##### A. Simulation assessment

Based on the official specifications of the BLE 4 [16], we consider the parameters given in TABLE II. The simulations were carried out in MATLAB\_R2023b to study the number of collisions, received PDUs, and the percentage of nodes found in relation to the number of available advertisers. For each scenario, the mean values over 2000 repetitions are reported.

1) *Evaluation of the number of collisions:* In general, a collision occurs when at least two advertisers send their PDU in the same communication channel simultaneously. For a more realistic simulation, we consider the overlapping of at least two PDUs in a scan window to be a collision. The average number (after the 2000 repetitions) of collisions between PDUs per

TABLE II: BLE parameters for the simulations

<b>Scan windows</b>	25 ms	<b>Scan int.</b>	50 ms
<b>Adv. int. (<math>T_{adv}</math>)</b>	100 ms	<b>Adv. delay (<math>t_d</math>)</b>	[0 10] ms
<b>PDU int.</b>	10 ms	<b>PDU transmission <math>t_x</math></b>	0.376 ms
<b>Adv. events</b>	15	<b>Repetitions</b>	2000

channel is given in Fig. 3a. We observe that the highest number of collisions occurs on the first advertising channel (37) and the number of collisions on channel 39 is the lowest. The resulting packet collision probability follows the Poisson distribution and has been approximated by Ghamari et al. in [17]. Based on their approximation, we compared the theoretical probability  $\hat{P}_{coll}$  (1) to the simulation collision probability for different numbers of advertisers ( $N$ ).

$$\hat{P}_{coll} = 1 - e^{\left(-2Nt_x \frac{1}{(T_{adv}+t_d)}\right)}, \quad (1)$$

$\frac{1}{(T_{adv}+t_d)}$  denotes the overall packet transmission rate.

We evaluate the errors between the theoretical collision probability and the simulation probability by using the following gap indicators (2): the Root Mean Square Error (RMSE), the Mean Absolute Error (MAE), and the Mean Absolute Percentage Error (MAPE). For  $N$  measurements,  $\hat{P}_{coll_i}$  and  $P_{coll_i}$  denote respectively the theoretical and the computed collision probabilities when  $i$  advertisers are available.

$$\begin{aligned} MAPE(\%) &= \frac{100}{N} \sum_{i=1}^N \left| \frac{\hat{P}_{coll_i} - P_{coll_i}}{P_{coll_i}} \right| \\ MAE &= \frac{1}{N} \sum_{i=1}^N \left| \hat{P}_{coll_i} - P_{coll_i} \right| \\ RMSE &= \sqrt{\frac{1}{N} \sum_{i=1}^N \left( \hat{P}_{coll_i} - P_{coll_i} \right)^2} \end{aligned} \quad (2)$$

From figures 3b and 3c, we observe that the distance between the theoretical and the simulation collision probabilities decreases while the number of advertisers increases. Thus, the simulations observed are close to the theoretical results, even when the number of advertisers increases.

2) *Number of received PDUs*: A reception occurs when there is no collision and no synchronization issue between the advertiser and the scanner. The average number of PDUs received per channel over 2000 repetitions is shown in Fig. 4. We observe that the number of PDUs received is slightly similar in channels 37 and 38 compared to channel 39. In addition to Fig. 3a, we can conclude that the number of PDUs lost due to synchronization issues in channel 39 is higher than in the other channels. This is because of the lower number of collisions in channel 39 and the lower number of PDUs received despite of the same number of PDUs sent in each channel.

3) *Evaluation of the Neighbor discovery*: The average percentage of nodes discovered per scan window is shown in Fig. 5. As expected, we observe that the time needed to discover nodes increases with the number of advertisers in the network.

However, after the 4th scan window, all nodes present in the network were discovered; thus, there was no point in continuing the scan after that windows. This means that 20% of the total number of scan windows can be skipped without affecting the discovery of all available nodes.

### B. Set-up and micro-benchmark

For the experimental part, we consider the esp32-devkit-1 (Fig. 6a) because it is widely used, affordable, easy to find in the market, and has two cores that share the same memory. Another important motivation for choosing this board is that it directly integrates a BLE chip. To assess the energy consumed by the board, we measured the current drawn during the scanning process using the KEYSIGHT CX3324A current waveform analyzer (Fig. 6b).

When applying parameters from TABLE II, the energy consumed by the board after 1-second scan (Fig. 7) shows the different steps during and before the scanning process: (i) Initialization of the board; (ii) Initialization of the BLE (with the related parameters); (iii) Scan process during the fixed 1-second. The numerical value of the current consumed after the 1-second scan is obtained by evaluating the area under the curve and revealing that the ESP32 board consumes around 0.0208 mA (during a 5-second scan, it consumes 0.0619 mA).

## VI. RESULTS AND DISCUSSIONS

The validation of our proposal considers the setup presented in Fig. 2. We provide an analysis of the core usage in terms of the energy consumption when the scanning duration is fixed. Furthermore, we evaluate the energy efficiency of our approach according to a fixed number of nodes to discover.

### A. Validation process

The scanning duration is fixed (1 to 5 seconds), and we compare the energy consumed by the node when it is equipped with a mono-core or a dual-core (utilizing either one core or the proposed dual-core configuration) microcontroller. To enhance the validation process, we consider the energy consumed during the scan of a native single-core microcontroller (ESP32-C3 [19]). In order to analyze the energy impact of the microcontroller's core during a fixed scanning duration, we evaluate the energy cost while discovering a fixed number of advertisers. In practice, the number of neighbors discovered during the scan is stored in the RTC (non-volatile shared) memory of the ESP32. This enables access by both cores of the microcontroller. We use the RTC memory to store the shared variable because its value can be read after a *Go\_to\_deep\_sleep()* function (Fig. 2). Given that the ESP32 task scheduler operates based on priority levels, the starting core must initiate the scanning process with a lower priority than the checking core. Consequently, the checking core has the ability to halt the starting core when the required number of nodes to discover is reached. During our experimentation, Core "1" corresponds to the *STARTING\_CORE*, and Core "0" corresponds to the *CHECKING\_CORE*.

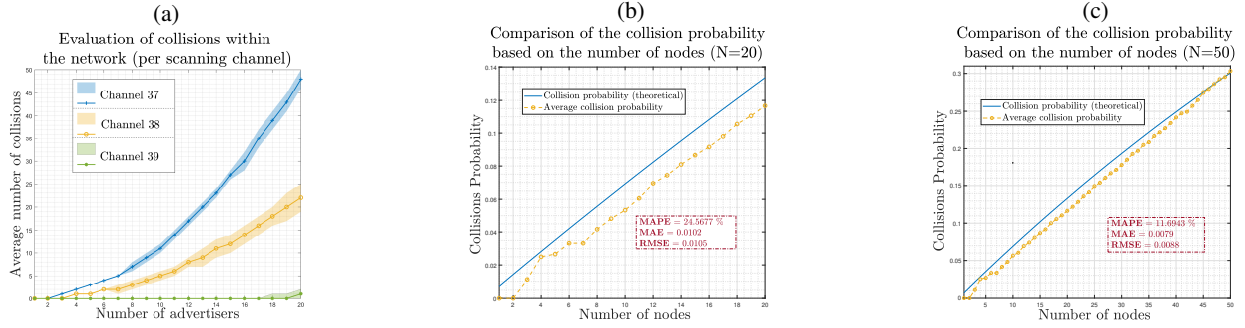


Fig. 3: (a) Evaluation of the number of collisions w.r.t. the number of advertisers. (b) Comparison of the collision probabilities with 20 advertisers (MAPE = 24.5677%, MAE = 0.0102, RMSE = 0.0105). (c) Comparison of the collision probabilities with 20 advertisers with 50 advertisers (MAPE = 11.6943%, MAE = 0.0079, RMSE = 0.0088)

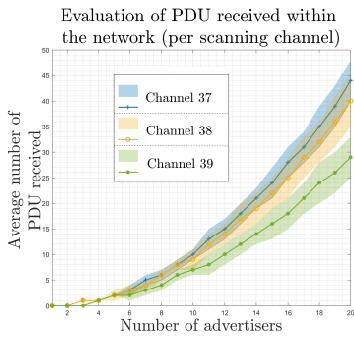


Fig. 4: Evaluation of the number of packets received per channel

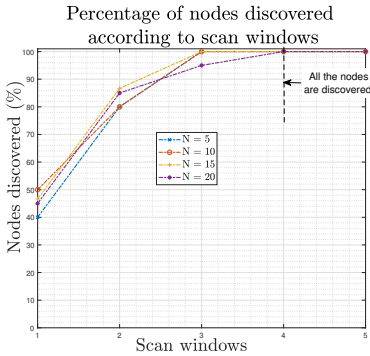


Fig. 5: Percentage of nodes discovered from available  $N$  advertisers w.r.t scanning windows ( $N = \{5, 10, 15, 20\}$ )

### B. Energy consumed from the core architecture's side

We analyzed and compared the energy impact of the scanning process by considering two different cases (Fig. 8): (i) the microcontroller is dual-core based but only one core is dedicated (native process) for the scanning process; (ii) the microcontroller is single-core based.

We observe that the average energy consumption of our

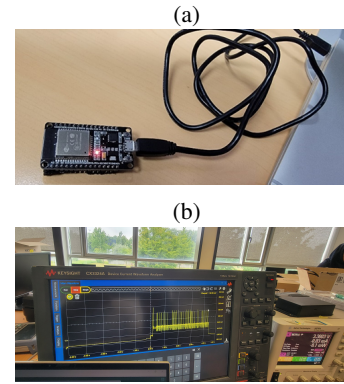


Fig. 6: (a) Board used: esp32-devkit-1 [18]. (b) Device Current Waveform Analyzer KEYSIGHT CX3324A.

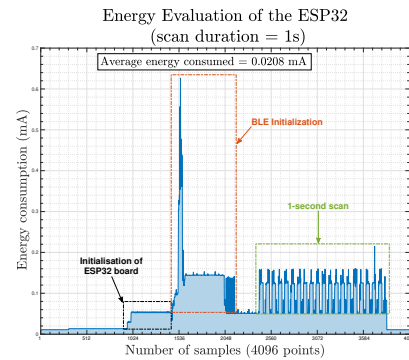


Fig. 7: Evaluation of the energy consumption of the ESP32 after a 1-second scan

proposal is lower than the energy consumed when only one core is utilized in a dual-core module. This can be attributed to the architecture described in Section IV, thus our proposed approach can save in average 0.0056 mA during the scanning process. We observe that as the scanning duration increases,



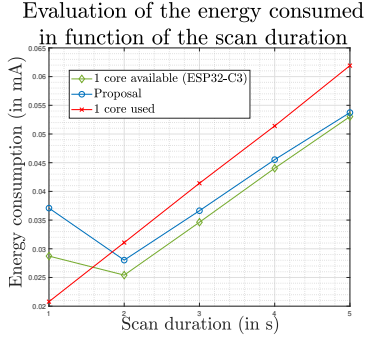


Fig. 8: Comparison of the energy consumed during a fixed scanning duration. In red, only one core is activated to perform the scanning process during a fixed time. In green, the microcontroller is a single core

the energy consumption of the dual-core module with only one core activated rapidly increases. Moreover, Fig. 8 reveals that utilizing the different cores available on a microcontroller can reduce the overall energy consumption during BLE scanning.

Nevertheless, the single-core module consumes less energy than a dual-core microcontroller during the NDP. This outcome is attributed to the simple and efficient architecture of the ESP32-C3 in comparison to the dual-core module. Despite the better performance of ESP32-C3, we observe that as the scanning duration increases, the energy savings become less pronounced compared to our proposed solution. Moreover, the scenarios depicted in Fig. 8 of our proposal represent the most unfavorable cases: in these instances, the requisite  $K$  nodes for discovery among the available  $N$  nodes remain undiscovered within the designated scanning duration.

### C. Impact of the number of nodes to discover

From Fig. 9a, we observe that both cores (*STARTING\_CORE* and *CHECKING\_CORE*) start exactly at the same time (11:26:44.404). Core 1 executes the setup function of the main program. While the starting core performs the scan process, the checking core (Core 0) counts the number of discovered nodes and compares it to the fixed threshold. When the number of discovered reaches the needed number, the checking core, which has a higher priority, stops this starting core. According to Fig. 9b, the fixed 5 nodes to be discovered are found at 11:26:44.782 (0.372 s after the beginning) and the ESP32 call *Go\_to\_deep\_sleep()* function at 11:26:44.825 (0.421 s after the beginning). Since the 5 different nodes are discovered before the scanning duration, the scanning process is stopped by Core 0 instead of the *STARTING\_CORE* that initializes the scan process (Fig. 9). Thus, by setting up the number of nodes to be discovered ( $K = 5$ ), the overall energy consumed by the scanner is reduced by 57.36% compared to the 1-second scan given in Fig. 7. This energy gain is due to the shorter scanning duration achieved by our multicore approach, which is able

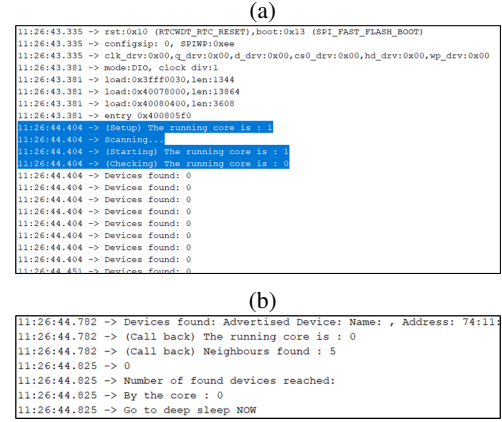


Fig. 9: Evaluation of the scanning duration when the number of nodes to discover is fixed ( $K = 5$ ). (a) Timestamp at the beginning. (b) Timestamp at the end

to dynamically adapt the scanning duration according to the number of nodes (advertisers) to be discovered.

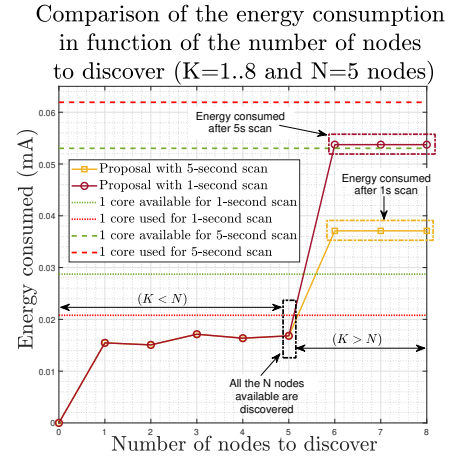


Fig. 10: Comparison of the energy consumed in terms of the number of nodes to discover (5 nodes are available)

The worst case occurs when the number of needed nodes  $K$  is not reached before the fixed scanning duration or if it is higher than the number of available nodes ( $K > N$ ). In this case, the overall energy consumed by the scanner will be equal to the energy drained during a normal scanning process.

By evaluating the number  $K$  of nodes to discover from 1 to 5, the related average energy consumed by the scanner is given in Fig. 10. When the  $K$  is smaller or equal to the number of available nodes ( $K \leq N$ ), the energy needed by the scanner to discover these nodes is slightly identical. Nevertheless, when  $K$  is higher than the number of available advertisers  $N$  the total energy consumed increases to the worst cases. As we can expect, the longer the scan time, the more energy the scanner

consumes ( $\approx 0.0208$  mA and  $\approx 0.0619$  mA for 1-second and 5-second respectively).

In contrast to related works (Section III) that attempt to determine the optimal parameters for energy-efficient NDP in BLE, we have proposed a new solution that takes advantage of the SMP architecture of recent microcontroller hardware to enable dynamic NDP based on the required number of nodes to be discovered. In this way, the proposed solution can be applied in addition to all those aimed at optimizing BLE parameters during NDP, regardless of the Bluetooth version.

## VII. CONCLUSION

In this paper, we proposed a new approach based on symmetric multiprocessing of recent microcontrollers to adapt the scanning duration dynamically during the NDP. The main objective is to efficiently use the energy to find the needed number of BLE devices. To achieve that, we parallelized the scanning process thanks to the widely used SMP architecture of microcontrollers. Two identical cores have been considered for this purpose: the *STARTING\_CORE* initiates the NDP, and the *CHECKING\_CORE* checks the input requirement. This novel approach reduces the energy consumption of a scanner regardless of the NDP parameters of the BLE. The evaluation process, which involved intensive simulations and real experiments, showed that our approach can save more than 50% energy. As future works, we plan to improve the energy consumption from the advertiser side during the NDP. The use of synchronization communication modules such as Wake-up Radio could be a promising direction to follow.

## REFERENCES

- [1] A. S. H. Abdul-Qawy, N. M. Almurisi, and S. Tadisetty, "Classification of energy saving techniques for iot-based heterogeneous wireless nodes," *Procedia Computer Science*, vol. 171, pp. 2590–2599, 2020.
- [2] W. S. Jeon, M. H. Dwijaksara, and D. G. Jeong, "Performance analysis of neighbor discovery process in bluetooth low-energy networks," *IEEE Transactions on Vehicular Technology*, vol. 66, pp. 1865–1871, 2017.
- [3] G. Shan, G. Choi, B.-h. Roh, and J. Kang, "An improved neighbor discovery process in ble 5.0," in *IEEE Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*, 2019.
- [4] B. Luo, F. Xiang, Z. Sun, and Y. Yao, "Ble neighbor discovery parameter configuration for iot applications," *IEEE Access*, vol. 7, pp. 54097–54105, 2019.
- [5] G. Shan and B. H. Roh, "Performance model for advanced neighbor discovery process in bluetooth low energy 5.0-enabled internet of things networks," *IEEE Transactions on Industrial Electronics*, vol. 67, pp. 10965–10974, 2020.
- [6] P. H. Kindt, S. Narayanaswamy, M. Saur, and S. Chakraborty, "Optimizing ble-like neighbor discovery," *IEEE Transactions on Mobile Computing*, vol. 21, no. 5, pp. 1779–1797, 2022.
- [7] W. Bai, J. Chen, Y. Xu, F. Song, H. Wang, G. Li, and Y. Jiao, "Euclidean-division-based low-complexity precise analytical approach of ble-like neighbor discovery latency," *IEEE Internet of Things Journal*, vol. 10, no. 5, pp. 4184–4201, 2023.
- [8] G. Shan and B. H. Roh, "Advertisement interval to minimize discovery time of whole ble advertisers," *IEEE Access*, vol. 6, pp. 17817–17825, 2018.
- [9] L. Wei, B. Zhou, X. Ma, D. Chen, J. Zhang, J. Peng, Q. Luo, L. Sun, D. Li, and L. Chen, "Lightning: A high-efficient neighbor discovery protocol for low duty cycle wsns," *IEEE Communications Letters*, vol. 20, no. 5, pp. 966–969, 2016.
- [10] T.-T. Yang and H.-W. Tseng, "Two-way communication with wait-slot scheme for neighbor discovery process in dense bluetooth low energy networks," in *13th International Conference on Network and Service Management (CNSM)*, pp. 1–7, 2017.
- [11] B. S. I. G. (SIG), "Core specification 5.0 – bluetooth® technology website," 2016.
- [12] G. Shan and B. hee Roh, "Maximized effective transmission rate model for advanced neighbor discovery process in bluetooth low energy 5.0," *IEEE Internet of Things Journal*, vol. 4662, pp. 1–13, 2022.
- [13] Z. Shen, Q. Yang, and H. Jiang, "Multichannel neighbor discovery in bluetooth low energy networks: Modeling and performance analysis," *IEEE Transactions on Mobile Computing*, vol. 22, no. 4, pp. 2262–2280, 2023.
- [14] K. Mikhaylov and H. Karvonen, "Wake-up radio enabled ble wearables: empirical and analytical evaluation of energy efficiency," in *IEEE Medical Information Communication Technology (ISMICT)*, 2020.
- [15] E. H. Djidi Nour, D. Wohwe Sambo, M. Gautier, O. Berder, and N. Mitton, "Wubble: Energy efficient ble neighborhood discovery leveraging wake-up radio," in *19th International Symposium on Algorithmics of Wireless Networks (ALGOWIN)*, pp. 74–86, 2023.
- [16] B. S. I. G. (SIG), "Core specification 4.0 – bluetooth® technology website," 2010.
- [17] M. Ghamari and et al., "Detailed examination of a packet collision model for bluetooth low energy advertising mode," *IEEE Access*, vol. 6, pp. 46066–46073, 2018.
- [18] ESPRESSIF, "Esp-idf programming guide - esp32-devkitm-1."
- [19] ESPRESSIF, "Esp-idf programming guide - esp32-c3."