applied sciences

Article

Energy-Saving Method for Nearby Wireless Battery-Powered Trackers Based on Their Cooperation

Nerijus Morkevičius [1], Agnius Liutkevičius [1,*], Laura Kižauskienė [1], Audronė Janavičiūtė [1] and Roman Banakh [2]

[1] Department of Computer Sciences, Kaunas University of Technology, 44249 Kaunas, Lithuania; nerijus.morkevicius@ktu.lt (N.M.); laura.kizauskiene@ktu.lt (L.K.); audrone.janaviciute@ktu.lt (A.J.)
[2] Department of Information Technology Security, Lviv Polytechnic National University, 79013 Lviv, Ukraine; roman.i.banakh@lpnu.ua
* Correspondence: agnius.liutkevicius@ktu.lt

Featured Application

This article describes the energy-saving method, which can be implemented in various wireless cargo or other asset tracking devices that operate using battery power. The idea came from situations where multiple cargo items with attached trackers are transported in the same truck, ship, plane, etc. Instead of sending their locations to the central server individually and constantly draining their batteries, trackers cooperate and periodically select one of the healthiest nearby tracking devices to report their positions to the central server. This can significantly improve the lifetime of such trackers, since communication with the central server requires much more energy than local data processing and local communication between nearby trackers. This method can be applied by various companies that produce GPS (Global Positioning System) tracking devices, used by logistics companies around the world.

Abstract

The tracking of assets or cargo is one of the main objectives of global logistics and transportation systems, ensuring operational efficiency, security, and timeliness. Currently, battery-operated GPS (Global Positioning System)-based tracking devices are used for this purpose. The main shortcoming of these devices is the lifetime of the batteries because they cannot be replaced or recharged, or because this is simply not economically feasible. Therefore, efficient methods are needed to prolong battery life as much as possible. Various existing energy-saving techniques can be applied to solve this problem. However, none of these consider situations in which multiple tracking devices are transported together and can cooperate to further increase their energy efficiency. In this study, we propose and evaluate the novel lightweight peer-to-peer energy-saving method for nearby wireless battery-powered trackers based on their cooperation. The proposed method is based on the short-range BLE (Bluetooth Low Energy) device discovery mechanism and the dynamic election of the leader tracker (with the highest battery capacity) to report the location of its own and other neighboring trackers to the central server. The experimental evaluation of the proposed method shows that, compared to the traditional approach, where each tracker sends its location individually, the proposed method allows a reduction in the average battery charge required for one position report from 19% to 240% per each cooperating tracker. The average energy consumption for one location report per node decreased from 4.68 mWh using the traditional approach to 3.93 mWh for 2 cooperating devices and 1.92 mWh for 15 cooperating devices.

Academic Editors: Miguel Jimeno and Alfredo J. Pérez

Received: 27 October 2025
Revised: 2 December 2025
Accepted: 3 December 2025
Published: 5 December 2025

Citation: Morkevičius, N.; Liutkevičius, A.; Kižauskienė, L.; Janavičiūtė, A.; Banakh, R. Energy-Saving Method for Nearby Wireless Battery-Powered Trackers Based on Their Cooperation. *Appl. Sci.* **2025**, *15*, 12886. https://doi.org/10.3390/app152412886

Keywords: tracking; cooperation; wireless; energy efficiency; positioning; GPS; Bluetooth; battery-powered

1. Introduction

1.1. Background

Cutting-edge technology has revolutionized the efficiency, transparency, and sustainability of supply chains, making modern logistics systems a highly advanced and well-organized ecosystem. Different technologies affect separate stages and processes within the supply chain. Route optimization based on artificial intelligence, demand forecasting algorithms, and inventory management solutions reduce the time and cost of logistics operations. The introduction of automation and robotics reduces the need for manual labor and increases the speed and accuracy of sorting, packaging, and delivery tasks. Computational vision, virtual and augmented reality technologies enhance the interaction and cooperation of innovative technologies, human resources, and organizational processes, leading this sector to become a smart industry with its own "intelligence" [1].

One of the primary goals of global logistics and transportation systems, responsible for cargo or asset tracking, is to ensure operational efficiency, security, and punctuality. The integration of GPS (Global Positioning System) and IoT (Internet of Things) technologies is essential for global cargo tracking [2], as modern supply chains must rely on accurate real-time data on the location and condition of cargo to avoid delays, optimize routes, and respond to unexpected events. IoT technology enables the construction of intelligent traceability network systems for products [3], which in turn increase visibility and proactive problem solving. Furthermore, the IoT is not just a technological upgrade, but a strategic asset that can define the competitive edge of supply chain operations in the contemporary business environment [4].

Despite academic interest and the number of published articles related to smart logistics, comprehensive surveys of IoT applications in this domain highlight both opportunities and challenges [5,6]. Current research indicates that key technological limitations related to IoT, RFID (Radio Frequency Identification) and WSN (wireless sensor network) infrastructures must still be resolved to achieve fully intelligent and scalable logistics systems [7]. In addition, smart sensing devices participating in data exchange are deficient in power and require energy friendly communication protocols [8]. Although research activity in smart logistics is increasing, the domain requires deeper investigation into critical technological aspects before reaching maturity [9]. It is still in its early stages and has significant potential for growth [10].

1.2. Problem and Purpose of the Study

One of the main challenges associated with long-term cargo tracking is high energy consumption because continuous data collection and transmission require constant energy sources. However, the lifespan of most IoT devices is dependent on the capacity of their battery. Therefore, the foundational technologies for cargo tracking solutions must be carefully designed and thoroughly evaluated to ensure continuity and reliability. Ensuring data transmission and synchronization in global locations with limited connectivity is another important issue. Most traditional cargo tracking solutions often rely on centralized servers and mobile communications networks, which may not be available in remote or low-coverage areas, leading to data loss or tracking interruptions. Moreover, constant server access requires high energy consumption, leading to the problem defined above. Typical tracking systems are based on independent tracking devices that do not share information. Cooperation between IoT devices could lead to more energy-efficient solutions, enhancing efficiency, battery savings, and adaptability to environmental changes. The main problem with battery-operated tracking devices is that, in most cases, their batteries cannot be replaced or recharged, or this is simply not economically feasible. Therefore, efficient methods are needed to prolong their battery life as much as possible.

The purpose of this study is to develop an efficient energy saving method for wireless tracking devices, which allows them to reduce their energy consumption and prolong battery life. During the investigation of the energy consumption aspects of IoT devices, four main subsystems are usually distinguished [11]: the power supply subsystem, the sensor subsystem, data processing modules and wireless data transfer modules. In order to minimize overall energy consumption and prolong battery life, the decrease in the energy requirements in each of these modules should be investigated. However, most researchers agree that sensors and data processing modules require significantly less energy compared to communication modules. The research results in [12] show that up to 93–94% of the total energy consumption of devices is used by communication modules.

Therefore, the proposed approach is based on the hypothesis that cooperation between nearby trackers and selection of only one device to report the position of all nearby devices will reduce such long-range communications to a minimum, decreasing the average energy usage of all cooperating trackers. Such a method would be useful for commercial GPS trackers, since it is quite common that multiple assets with attached trackers are transported together, making their cooperation possible. This would allow the creation of cargo tracking solutions that ensure energy efficiency and uninterrupted global tracking, allowing tracking devices to cooperate by sharing data and roles.

1.3. Related Work

The use of low-power IoT technologies for cargo tracking is one of the primary aspects proposed by other authors to address the problem of the extension of battery life. For example, the systems presented in [13–15] are developed on embedded hardware platforms that operate with relatively low energy consumption. The research in [13] incorporates Raspberry Pi and GPS modules, which can operate in low-power states, reducing overall energy consumption. In [14], the authors introduce a fleet management system using ESP8266 microcontrollers integrated with load cells for weight tracking and GPS modules for location tracking that ensure cost-effective real-time data collection and transmission. Another approach includes a combination of GPS and GSM (Global System for Mobile Communications) modules with microcontrollers suitable for long-term tracking, is proposed in [15]. A proposed battery-powered GPS tracker sends location updates via SMS (Short Message Service) and hosts data on a MongoDB/NodeJS server, besides including an alert system to notify nearby devices during emergencies. All articles highlight another important aspect: periodic data transmission. Transmitting data at intervals rather than continuously reduces energy drainage. Data transmission intervals can be adjusted based on the device's battery level and cargo status. This allows devices to remain in "sleep mode" more frequently, transmitting data only as needed. Furthermore, GSM modules can send data in compact formats, minimizing communication overhead and energy usage [15]. However, the energy-saving strategies in these articles are described at a general level and lack specific algorithms or protocols detailing how devices should transition to sleep mode under various conditions (e.g., cargo movement or environmental changes).

The integration of GPS and IoT technologies is essential for global cargo tracking [2]. More insights into the use of GPS and IoT for global tracking, providing scalable and user-friendly solutions, are presented in [16,17]. The systems described utilize cloud databases (e.g., Firebase or custom IoT server setups) to store and manage tracking data, ensuring scalability and accessibility, which is critical for a global tracking system. Reference [16] focuses on maritime vessel tracking, which includes features such as iceberg detection and emergency response systems. However, these elements are not so relevant to our problem, because the environmental monitoring aspect is not a part of our focus. Similarly, the techniques presented in [17] are narrowly tailored to cargo logistics and client-facing solutions, not considering the goal of achieving energy efficiency and cooperation of autonomous devices.

Another relevant aspect identified in the research literature is collaboration between devices and data synchronization. This can bring valuable benefits to achieving energy efficiency in IoT networks. As mentioned above, [15] discusses how IoT devices can communicate via GSM modules, relaying location data to a central server. While data synchronization in this setup is centralized, which could be a single point of failure, it demonstrates foundational principles for data synchronization that can be adapted to a decentralized system. The principles of device collaboration are presented in the research by [18]. Said research describes how multiple UAVs (unmanned aerial vehicles) can leverage 5G PRS (5G Positioning Reference Signal) to maintain real-time positional accuracy, even in dense urban environments, assuring high-precision collaboration between nodes and achieving sub-meter accuracy and low-latency data exchange. This capability is crucial to ensuring smooth collaboration between devices, especially in dynamic or remote environments. Although the authors of [18] solve different problems related to location accuracy and latency rather than energy efficiency, this example of collaborating nodes can inspire a similar model for coordinating tracking devices in our network.

A research domain where energy efficiency is widely recognized as a fundamental challenge is wireless sensor networks. Consequently, researchers propose various energy-saving strategies that focus on the different layers and operational mechanisms of WSNs. Some studies apply different sensor activation parameters to avoid continuous monitoring overhead [19,20], others introduce cooperative node scheduling to activate only a minimal subset of nodes required for tracking tasks [21], and other works explore role-based trilateration coordination to limit redundant sensing and communication [22]. Alternative approaches include clustering, which is used to reduce GPS usage and energy consumption [23] or energy-aware cooperative routing in heterogeneous 5G-enabled WSNs [24]. Finally, the study in [25] demonstrates how communication overhead and energy usage in mobile and static wireless-powered IoT networks can be reduced by reinforcement-learning-based resource allocation, thereby extending the spectrum of energy-efficient coordination strategies relevant to multi-node tracking systems. These findings, supported by our comparative analysis Table 1, demonstrate that the reviewed literature covers different technological, architectural, and functional aspects of tracking systems. The research presented in [19] seems to be mostly conceptually aligned with the research presented in this article. However, agent-based reasoning and cluster-formation mechanisms would introduce unnecessary architectural complexity and communication overhead for co-located mobile tracking devices in our case, making them less energy efficient compared to lightweight local cooperation based on the decentralized approach. Finally, none of the analyzed works combine all the strengths of mobile multi-node cooperation, local short-range wireless discovery, leader-based task delegation, and decentralized ad hoc network architecture to improve energy efficiency. Together with real hardware energy measurements, these are the main contributions of our proposed solution, making it the closest to real-world multi-device cargo tracking scenarios that are practical and measurable.

Table 1. The comparison and summary of related work.

Article	Principal Goal and Problem Domain	Technologies and Methods Used	Localization Method	Energy Saving Strategy	Sensor Mobility	HW Testbed
[13]	Provide real-time GPS-based tracking and monitoring, ensuring safety in transportation	Raspberry Pi-based embedded system, web interface, email notification, cloud data storage	GPS	None	Mobile	Yes
[14]	Enhance fleet management using IoT-based monitoring	ESP8266 IoT sensing; Wi-Fi data transmission; cloud dashboard; alerting	GPS	None	Mobile	Yes
[15]	Provide portable GPS tracking with SOS alert capability for Personal safety, emergency notification, location sharing	Microcontroller, GPS, GSM/SMS, cloud server (NodeJS/MongoDB), web dashboard	GPS	None	Mobile	Yes
[16]	Create an IoT-based global ship tracking, obstacle detection, and automated rescue system for maritime navigation safety and accident prevention	Ultrasonic sensing, IR sensing, image processing (MATLAB), Arduino control, Wi-Fi data reporting	GPS	None	Mobile	Yes
[17]	Develop a cross-platform cargo tracking and information management system	Mobile app (Flutter), Firebase cloud storage, Node.js backend, Google Maps API	GPS	None	Mobile	No
[18]	Achieve high-accuracy real-time UAV tracking using 5G PRS-based positioning instead of GPS	5G PRS signals, TDOA multilateration, Kalman filtering, multi-object tracking, MATLAB simulation	Hybrid 5G multilateration	None	Mobile	No
[19]	Reduce redundant communication in large-scale container logistics	Agent-based decision-making, clustering, utility evaluation	GPS	Context-based activation	Mobile/Static	No
[20]	Reduce energy via dispatcher-based scheduling in WSN-based moving target tracking	Cooperative dispatchers, dynamic wake-up scheduling, trilateration, probabilistic tracker selection	Trilateration (distance-based)	Selective activation, sleep mode, minimal sensor set, limited communication	Static	No
[21]	Track a moving target using optimized node scheduling	Kalman-consensus filtering, optimization	Mathematical prediction; no GPS	Dynamic scheduling of active nodes	Static	No
[22]	Track objects in WSNs more energy-efficiently without reducing accuracy	Node optimization, mobile node trajectory optimization, node clustering, data reporting optimization and detection optimization	Non-GPS sensing (object detection)	Optimized node use and minimized transmissions	Static	Yes
[23]	Tracking of resource-constrained mobile nodes by a cluster-based cooperative tracking algorithm	Cooperative Kalman Filter, Boid Flocking Algorithm, Cluster-based cooperative tracking architecture	GPS	Clustering; Kalman Filter	Mobile	No
[24]	Maintain fairness, prevent overload, Improve energy efficiency in heterogeneous overlapping 5G WSNs	Cooperative routing, energy pools, fairness algorithms	Assumes known node locations; no GPS	Energy pooling and fair distribution	Static	No
[25]	Minimize the Age of Information (AoI) in mobile wireless-powered IoT networks by optimizing resource allocation	Deep deterministic policy gradient (DDPG)-based distributed multinode resource allocation (DDMRA) algorithm	None	Optimizing transmit power, transmission duration, and node/channel selection through the distributed DDMRA algorithm	Mobile/Static	No

1.4. Aim and Principal Conclusions

The research presented in this paper contributes to the field of efficient energy usage and energy savings of battery-powered IoT tracking devices (e.g., GPS asset trackers). The aim of this study is to create an energy-saving method for multiple wireless tracking devices based on the cooperation of nearby trackers. We propose a discovery and coordination model and algorithm that allow these trackers to elect the healthiest device, i.e., with the highest battery capacity, to send the location data of all neighboring trackers to the central server in a periodic manner. This approach reduces communication energy usage since only one tracker obtains a position using GPS satellites and sends it to the central server, while others can be put into sleep mode.

The proposed method was implemented as a prototype system, where several nearby trackers wake up periodically, checking the state of their neighbors (or the absence of them) and selecting the "leader" tracker, who is responsible for communication with the central server and sending the position of all trackers in its proximity, including itself. Unlike previous studies, we propose an additional improvement in terms of reduced communication between the central server and tracking devices, as well as minimized scanning to the GPS signals, replacing it with less energy demanding local communication and synchronization between neighboring trackers. This approach allows us to save energy and prolong the average battery life of the tracking devices.

This article evaluates the efficiency of the proposed method in terms of energy consumption. We compare the energy efficiency of the prototype system, based on the proposed cooperation method, with the traditional one, where each tracker sends its location to the central server individually. The results show a high potential for the proposed approach that allows us to decrease the average battery charge consumption by up to 2.4 times when more than 15 devices cooperate.

The main contributions of this paper are listed as follows:

- An energy-saving method for multiple co-located wireless tracking devices based on the cooperation of these trackers. In contrast to existing energy-saving techniques and methods, the proposed method reduces energy usage by employing the Bluetooth Low Energy (BLE) protocol-based short-range wireless discovery mechanism combined with a lightweight cooperation and leader node election algorithm, enabling fully decentralized, coordinator-less operation, which is resistant to neighbor changes. This approach reduces the average energy consumption of all cooperating nodes, by minimizing long-range communications and distributing energy consumption among the cooperating nodes.
- An experimental evaluation of the proposed method, by measuring the energy consumption of the different phases of operation of tracking devices. The experiments were carried out using ESP32 microcontrollers with an integrated BLE radio module, the NEO-M8N GNSS (global navigation satellite system) module and the SIM7000E communication module.
- The energy consumption measurements and analysis of each hardware module, including GNSS, mobile communication, CPU, and BLE radio. These findings enable future researchers to assess the real-world energy efficiency of their proposed methods.

This article is organized as follows. Section 2 covers the cooperation model, algorithms, and evaluation methodology. Section 3 presents the results obtained. Section 4 discusses the results, and Section 5 concludes the article.

2. Materials and Methods

2.1. Background and the Proposed Cooperation Model

The main goal of the proposed method is to improve the lifetime of wireless battery-operated asset tracking devices, reducing their energy consumption and battery drain. Existing trackers operate on their own, periodically sending their locations to the central server for further processing. Existing trackers are using sleep states and periodic data transmissions in order to reduce energy consumption to a minimum. However, in some cases, energy consumption can be reduced even further.

The idea of the proposed method comes from the real-life scenario, when multiple GPS trackers are transported together in the same truck, ship, plane or by other means. Instead of sending their locations to the central server individually and independently, tracking devices can cooperate and periodically select one of the healthiest neighbors to report their positions to the central server. Therefore, only one tracker at a time is used to obtain the position and communicate it to the central server, while others can be put in a sleep state, saving energy. This can improve the lifetime of these trackers, as synchronization with GPS satellites and communication with the central server require more energy than local data processing and local communication between neighboring trackers.

As shown in Figure 1, cooperation between asset trackers is possible, when multiple trackers are transported together by plane, boat, truck, etc. The tracking devices wake up periodically and check the surroundings for other tracking devices, using the proposed discovery and data synchronization algorithm (presented in Section 2.2) via Bluetooth communication channels. If there are no adjacent trackers, the tracking device operates as usual, sending its position to the central server on its own. However, if there are at least two neighboring trackers, they discover each other and synchronize their states ("Discovery and synchronization" phase in Figure 1). Then the cooperation phase takes place, where the tracking devices elect a "Leader" node that will be responsible for sending their locations to the central server until the next discovery and synchronization cycle. The leader node is selected considering the state of the batteries of all cooperating trackers. The tracker with the healthiest battery becomes a leader and communicates the IDs and locations of all trackers, including itself, to the central server. The remaining trackers are put to sleep mode for energy conservation. This cycle is constantly repeated and resistant to changes in the number of cooperating trackers. Even if the leader node is removed from the cooperating tracker group, the next time the nodes wake up and synchronize, they will elect a new leader. Moreover, the new leader is reelected depending on the current state of the batteries of all cooperating trackers. This approach allows to balance energy consumption between all cooperating trackers, prolonging their average battery life.

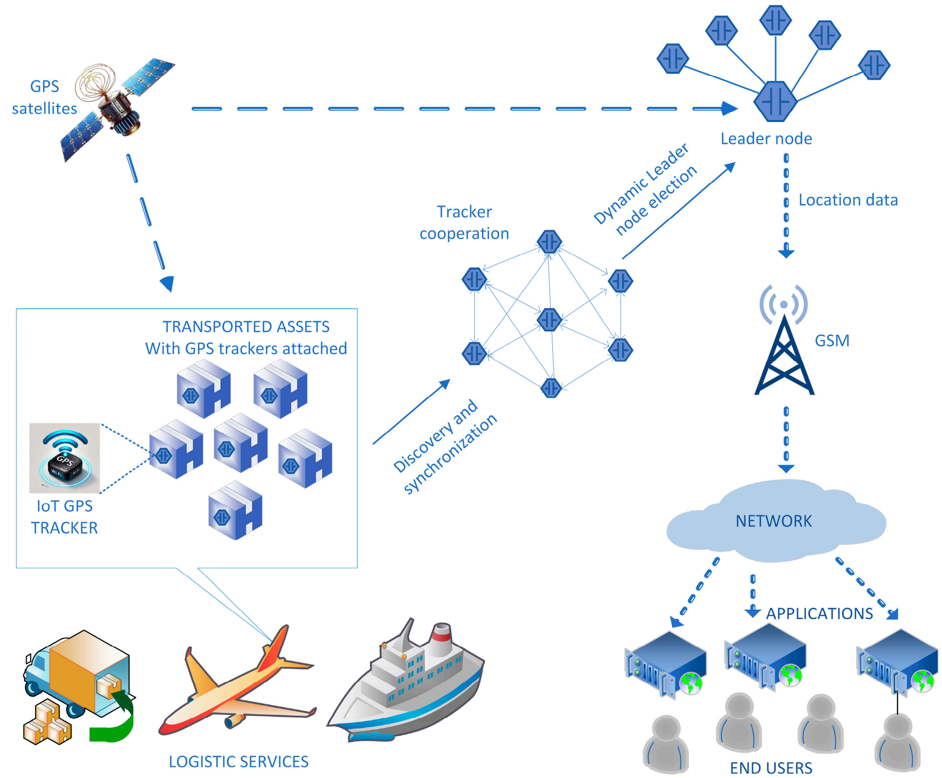

Figure 1. The proposed model for the cooperation of asset trackers.

2.2. Neigbor Discovery, Data Synchronization, and Leader Node Election Algorithms

The proposed energy saving method is based on the cooperation model, which includes three main phases: discovery, synchronization, and leader node election. The algorithms of each phase are presented below.

2.2.1. Neighbor Discovery Algorithm

The discovery phase includes the wake-up of the tracking devices and the scanning of the radio media for advertisements from other adjacent devices.

The main challenge for this discovery process is a specific t_{disc} neighbor discovery time window, where all nodes should wake up and scan radio media to detect if there are any neighbors.

The flowchart of the discovery algorithm is presented in Figure 2. The proposed algorithm utilizes several mechanisms for this purpose, including internal clock, timers, and CPU (central processing unit) interrupts. Each tracking device sets its internal timer in such a way that the device is put to sleep state until its timer generates an interrupt and wakes up the tracker at a specific predefined time moment. We used configurations, where the tracker is woken up at the beginning of each hour (1:00, 2:00, …, 24:00), though this time and period could be adjusted according to the specific application and user needs. Finally, when the tracker wakes up, it can start advertising its state and, at the same time, scan for neighbors using its Bluetooth radio. The proposed algorithm uses a predefined time window t_{disc} for this purpose, which can be adjusted. A bigger t_{disc} window means more chances to detect neighbors, but, at the same time, increases energy usage. If the tracking device does not detect any neighbors in its vicinity during the t_{disc} window, it obtains its location using the GPS receiver and sends this location via the mobile communication network to the central server by itself. In contrast, if the tracker detects at least one neighbor, then the data synchronization phase takes place.

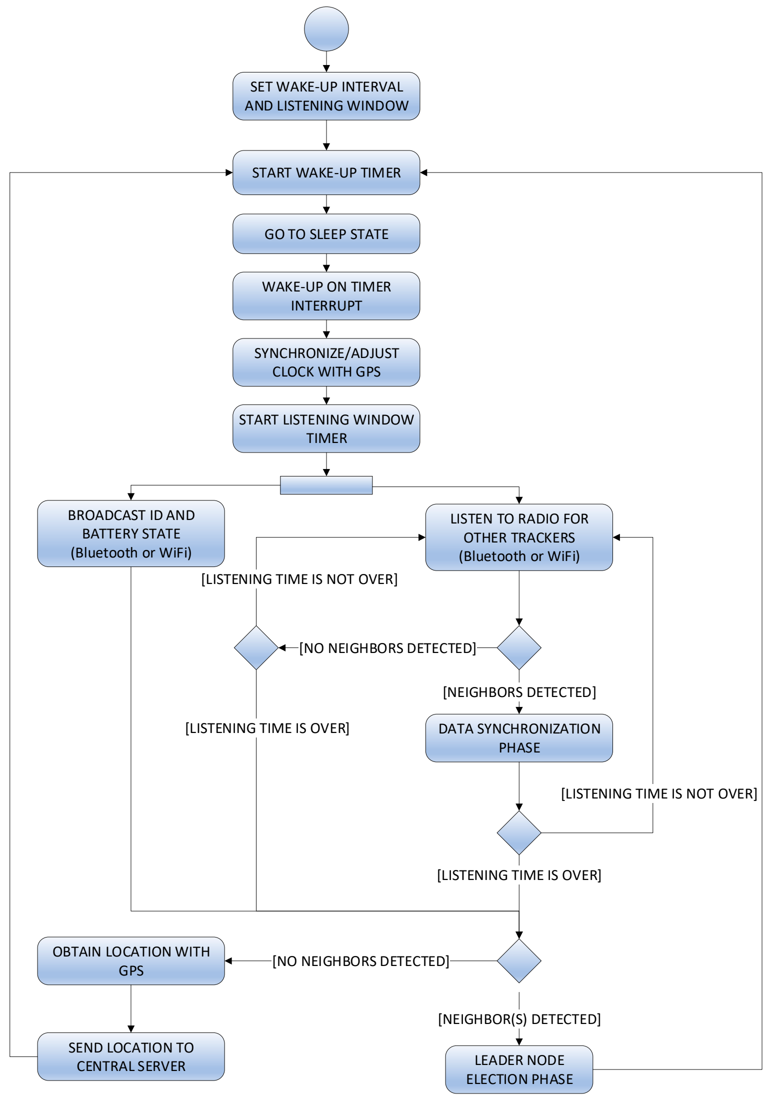

Figure 2. The flowchart of the general cooperation algorithm, including neighbor discovery, data synchronization and leader node election phases.

2.2.2. Data Synchronization Algorithm

The data synchronization algorithm (Figure 3) is used by the trackers to obtain the battery states of all other nearby trackers.

At the beginning of the t_{disc} window, discussed in Section 2.2.1, each tracking device starts to advertise its unique ID and battery state via Bluetooth. In other words, the trackers advertise their state and scan radio media at the same time (see Figure 2). This allows us to achieve two goals: trackers can detect each other by scanning radio media and obtain the exact level of the battery of each neighbor. Each tracker makes a list of all its neighbors with corresponding battery levels. When the data synchronization process ends, each tracker contains a table with all neighbors' IDs and current battery levels, including itself. Then the leader node elections process starts.

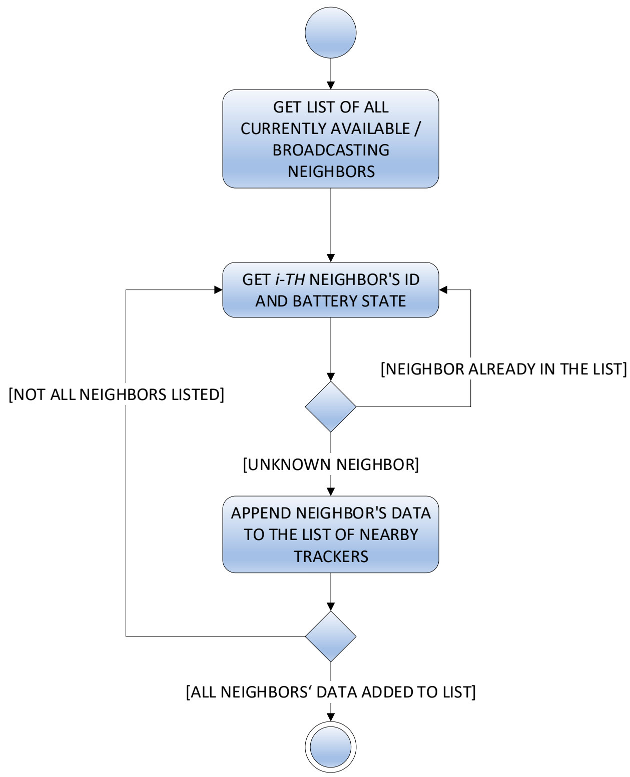

Figure 3. The flowchart of the data synchronization algorithm.

2.2.3. Leader Node Election Algorithm

The leader node election algorithm is used to choose the tracking device with the highest battery level among all neighboring trackers. The flowchart of the leader node election algorithm is presented in Figure 4:

This device is responsible for sending its and all its neighbors' IDs to the central server. To reduce the overall processing time and avoid complex calculations and data exchange between trackers, we chose a very straightforward approach. Since all trackers are synchronized and know the battery level of each other, they can independently decide who should be the leader. If the tracker detects that it has the highest battery level among its neighbors, it understands that it should become a leader and sends the location together with all tracker IDs to the central server. If there is a situation where several trackers have the same battery level, then the tracker with the smallest ID becomes the leader, avoiding excessive communication and additional voting. Finally, the leader tracker obtains the location via a GPS receiver, sends this location together with all tracker IDs to the central server, and is put to sleep state. At the same time, the other trackers are also put to sleep state until the next neighbor discovery time window.

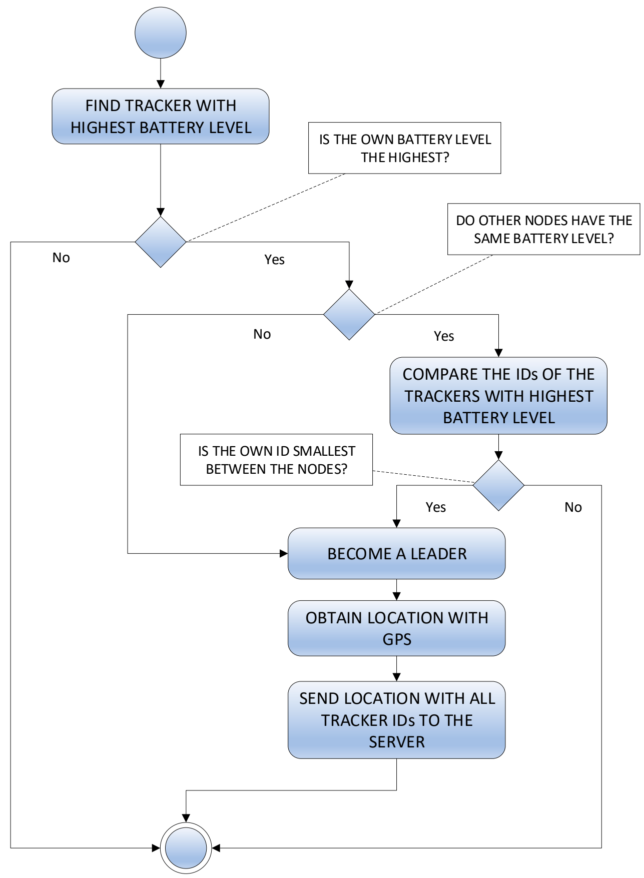

Figure 4. The flowchart of the leader node election algorithm.

2.3. Performance Evaluation Methodology

The following approach was used to evaluate the potential energy savings of the proposed method: the energy requirements of the four main components of the location tracking devices were experimentally measured under conditions similar to those expected in the proposed method. The main components of the cooperating devices are:

- CPU running the code and managing all the communications;
- BLE radio module, used for the detection of other cooperating devices;
- GNSS module used for location services;
- Mobile communication module that is used to report location data to the central server.

The conditions for evaluating individual devices were based on the expected application scenario. The BLE scan interval and scan window were chosen to ensure reduced energy requirements while potentially compromising detection latency, which is not very important when all devices are looking for each other for a significantly longer period of time imposed by the neighbor discovery time window t_{disc}. The energy and time requirements of the GNSS module were measured from the "cold" state, because the potential application scenario implies only several location operations per day.

Subsequently, a spreadsheet-based analysis was conducted to estimate the potential energy requirements of the proposed method when applied to varying numbers of cooperating devices. The following assumptions were made for this evaluation: all devices are stable, they do not change relative locations, and no new devices arrive or leave. All devices have exactly the same battery, and all batteries are used uniformly. The leader node changes after each wake-up and cooperation cycle. Although these assumptions potentially represent the "best case scenario", they also provide a good sense of situations when the proposed method is preferable to the "traditional" approach.

2.4. Experimental Setup

To evaluate the performance and energy requirements using BLE communications, four ESP32-C6-DevKitC-1 development boards [26] were used. They are based on Espressif Systems's (Shanghai, China) ESP32-C6-WROOM-1U general-purpose modules that integrate Wi-Fi 6 in the 2.4 GHz band, Bluetooth LE (Bluetooth 5.3 certified) and IEEE 802.15.4 (Zigbee 3.0 and Thread 1.3) functionalities. ESP32-C6-WROOM-1U has a 32-bit RISC-V single-core microprocessor, 320 KB of ROM (read-only memory), 512 + 16 KB of SRAM (static random-access memory), 8 MB quad SPI (serial peripheral interface) flash memory, GPIO (general-purpose input/output), SPI, parallel IO, UART (universal asynchronous receiver-transmitter), I2C (inter-integrated circuit), etc., connectivity capabilities. It was also used as the main microcontroller of the system managing and communicating with all other modules. Hardware UART and GPIO pins were used for communication with GNSS and mobile communications modules using AT (attention) commands.

The GY-GPSV3-NEO-M8N module [27] based on the u-blox (Thalwil, canton of Zürich, Switzerland) NEO-M8N chip [28] was used for location services. NEO-M8N supports GPS/QZSS L1 C/A, Galileo E1B/C, BeiDou B1, and GLONASS L10F GNSS systems, and is able to communicate with host system using UART, USB (universal serial bus), SPI and I2C-compliant DDC (display data channel) communication protocols. NEO-M8N has an RTC (real-time clock) based on TCXO (temperature compensated crystal oscillator) for accelerated weak signal acquisition and faster start times. In our hardware setup, ESP32 communicated with NEO-M8N using the hardware UART port and the NMEA 0183 standard commands.

Waveshare's (Shenzhen, China) SIM7000E NB-IOT HAT [29] based on Simcom's (Shanghai, China) SIM7000E Quad-Band (B3/B8/B20/B28) LTE-FDD and Dual-Band (900/1800Mhz) GPRS/EDGE (general packet radio service/enhanced data rates for GSM evolution) communications module [30], which supports LTE CAT-M1 (eMTC) and NB-IoT was used for mobile data communications. This module natively supports the TCP/IP, UDP/IP, EMAIL, FTP, HTTP, and HTTPS protocols. SIM7000E provides UART, USB 2.0, I2C, and GPIO interfaces. In our experimental setup, ESP32 communicated with the SIM7000E module using hardware UART port and AT commands.

The Nordic Semiconductor (Trondheim, Norway) Power Profiler Kit II v. 1.0.1 (PPK2) [31] was used for power consumption measurements. PPK2 is capable of measuring current from 500 nA to 1 A at up to 100 kS/s by acting as an ampere meter. The 8-pin digital port could be used for synchronization with other systems. A desktop application is designed to acquire real-time current measurements through a USB interface, offering capabilities for data visualization and export to support further analytical processing.

All devices were powered with 5 V external power supplies. The external voltage was reduced to a stable 3.3 V using linear voltage regulators that are an integral part of all devices used in the experiments. All measurements presented in this paper are expressed in Coulombs and Milliamperes, because these are the values which were physically measured. If needed, one could convert these values into energy units (i.e., Joules) by simply multiplying by 3.3 V.

Figure 5 shows the hardware setup used to measure the energy requirements during the scanning and advertising phases of the BLE protocol. In this scenario, the PPK2 was used to measure the current consumed by the ESP32 device while it exchanged information with other neighboring BLE devices. Three additional ESP32 microcontrollers were used as cooperating devices in this process.

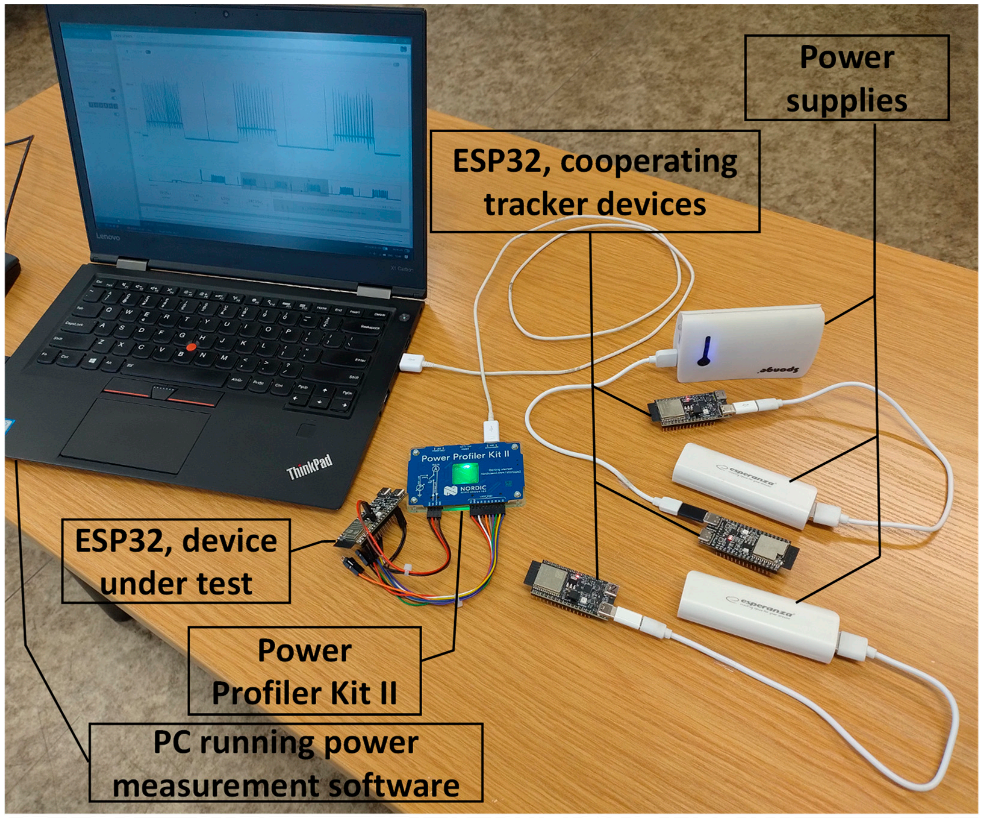

Figure 5. Hardware setup used to measure energy requirements during BLE scanning and data exchange.

Arduino IDE v. 2.3.4 was used as the main software development platform for programming ESP32 microcontrollers. An Arduino Core for Espressif Devices software suite v. 3.2.0 [32] developed by Espressif Systems was installed for support of the ESP32 functionality. BotleticsSIM7000 library v. 1.0.6 by Botletics LLC [33] was used for communications with the SIM7000E module. The TinyGPSPlus library v. 1.0.3 was used to assist in parsing NMEA (National Marine Electronics Association) sentences during operation with the NEO-M8N GNSS module.

3. Results

3.1. Energy Requirements for the Discovery of Neighbors (BLE Module)

Four ESP32-C6-DevKitC-1 development boards running advertising/scanning process were used. The same software was deployed on all devices, which simultaneously executed scanning and advertising procedures in order to detect and collect information from other similar devices within their neighborhood. Only one ESP32 device was connected to the Power Profiler Kit to investigate power requirements during this process. Figure 6 shows the screenshot of the PPK2 software which presents current consumption waveform for the ESP32 microcontroller while performing advertising/scanning process using BLE protocol. The time corresponding to a single BLE scan interval is shown, because all other scan intervals are almost identical to the first one. Moreover, the detection of all similar devices usually occurs during the first scan interval, with rare exceptions where one device is identified during the second interval. Several different phases of the process are highlighted in Figure 6. Phase “a” represents current consumption while BLE radio is turned off and ESP32 is running code. In this mode an average current consumption of the ESP32 is 22.1 mA which remains quite stable, with peaks reaching only 23.6

mA. Phase “b” is the transitional process when BLE radio is turned on, but advertising/scanning process is not started yet. Phase “c” and “d” represent one scanning interval of BLE protocol. In our experiments, a 1500 ms scanning interval was used, as the analysis presented in [34] indicates that this interval length ensures good energy efficiency while maintaining adequate device detection latency. Phase “c” is a 620 ms long scan window, i.e., the active part of the scanning interval when the detection of other devices is performed. Phase “d” is the rest of the scanning interval when only advertising is performed. 300 ms advertising interval was used in our experiments. Waveform part “f” shows the three peaks of current during advertising process which correspond to the three advertising packets sent by the BLE radio on three different channels. Waveform “g” shows the increased current consumption corresponding to the reception of an incoming advertising packet, which is subsequently processed by the ESP32. Part “e” shows the status of two GPIO pins, which were used to provide a visual representation of device detection events. The results shown in Figure 6e indicate that four BLE devices were detected (top line, pin 0), and three of them (bottom line, pin 1) were recognized as cooperating trackers. At the same time, they were added to the known devices list and exchange of the battery status information (total of 20 bytes) was performed.

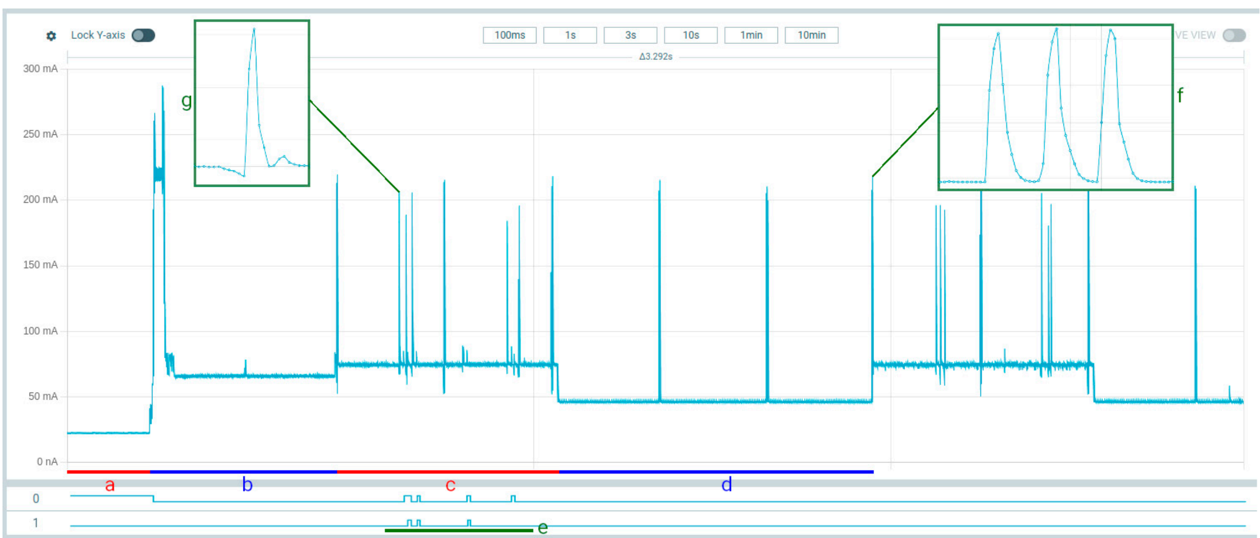

Figure 6. The current consumption waveform for the ESP32 microcontroller while advertising and scanning for similar devices using BLE.

The provided results clearly indicate that there is no visible difference in energy requirements to receive advertising packet from cooperating trackers (when useful information is received) and other random BLE devices. This is because all information is already embedded into the advertising packets and no additional packet exchange is required. Moreover, it is possible to separate distinct parts of the energy requirements and compare them. The comparison chart of the areas where the energy is actually consumed is presented in Figure 7. The numbers presented in Figure 7 are extracted from the current waveform shown in Figure 6 using the analysis capabilities of the PPK2 software.

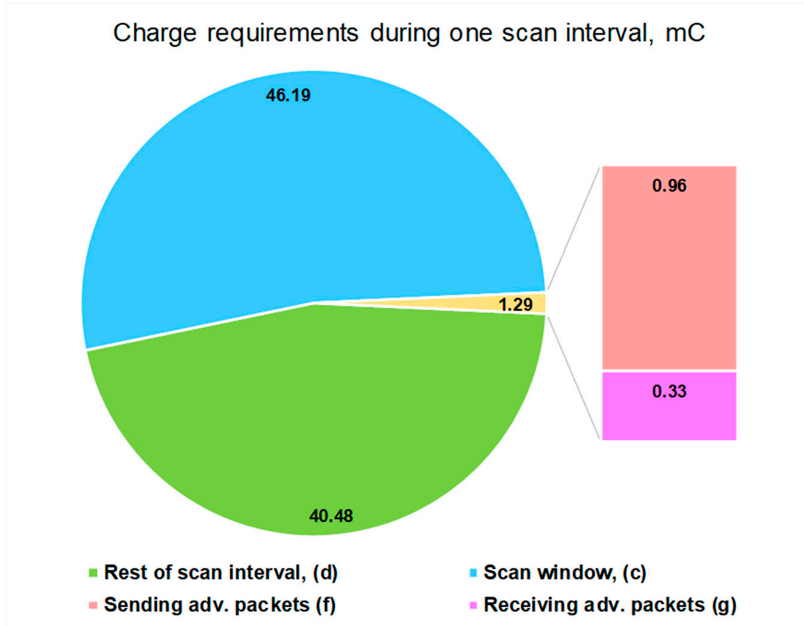

Figure 7. Distribution of the charge requirements during the one full scanning interval between various parts of the BLE scanning/advertising process.

The analysis of Figures 6 and 7 shows that only approximately 1.5% of the total energy requirements depend on the frequency of advertising and scanning packets, as well as on the number of nearby BLE devices. Moreover, no significant differences are observed regardless of whether these devices are cooperating trackers or totally random, unrelated BLE devices. This means that energy requirements may slightly increase in environments which have lots of random BLE devices. But the percentage of energy consumed by communicating with other BLE devices is not very high.

The comprehensive analysis of the current waveforms measured using different numbers of cooperating devices (from 2 to 5 devices) allows one to generalize the charge requirements during one scanning interval for one cooperating device using the following equation:

$$Q_{int} = 5 \cdot Q_{ap} + 2 \cdot n \cdot Q_{sp} + I_s \cdot t_s + I_{rest} \cdot t_{rest}, \tag{1}$$

where Q_{int} is the total charge required during full scanning interval t_{int}, which was fixed to 1500 ms in all experiments ($t_{int} = 1500\ ms$). Q_{ap} is the charge required to send three advertising packets on different BLE channels. The interval of the advertising packets was fixed to 300 ms, therefore 5 of such packets are sent during each scanning interval. Q_{sp} is the total charge required to receive advertising packets from the other devices. Only two of such packets could be received during each 620 ms long scan window because the interval for advertising packets is 300 ms. n is the number of cooperating devices, excluding the device which energy consumption is evaluated. t_s is the length of scan window which was fixed to 620 ms. t_{rest} is the length of the rest of the scanning interval, which was fixed to 880 ms ($t_{rest} = t_{int} - t_s = 880\ ms$). I_s and I_{rest} are the average currents consumed by the device (excluding the peaks when packets are received or transmitted) during scan window and the rest of the scanning interval, respectively.

If Equation (1) is applied to the situation corresponding to the experiment depicted in Figures 6 and 7, then the following results may be obtained:

$$\begin{gathered} Q_{int} = 5 \cdot 0.00019 + 2 \cdot 4 \cdot 0.00004 + 0.0745 \cdot t_s + 0.0460 \cdot t_{rest} = \\ 0.00095 + 0.00032 + 0.04619 + 0.04048 = 0.08794 \end{gathered} \tag{2}$$

where the values for $Q_{ap} = 0.19$ mC, $Q_{sp} = 0.04$ mC, $I_s = 74.5$ mA, $I_{rest} = 46$ mA were acquired from the analysis of current consumption graphs corresponding to experimental evaluation of situations where 2, 3, and 4 devices are cooperating. One can observe that these estimates quite closely repeat the experimental values provided in Figure 7.

Figure 8 shows the impact of varying numbers of active (i.e., advertising or cooperating) BLE devices present in the neighborhood. The first bar in this graph was measured experimentally using 4 devices. The other bars are the estimates using Equation (1) and providing $n = 10, 15, 20, 30$ accordingly.

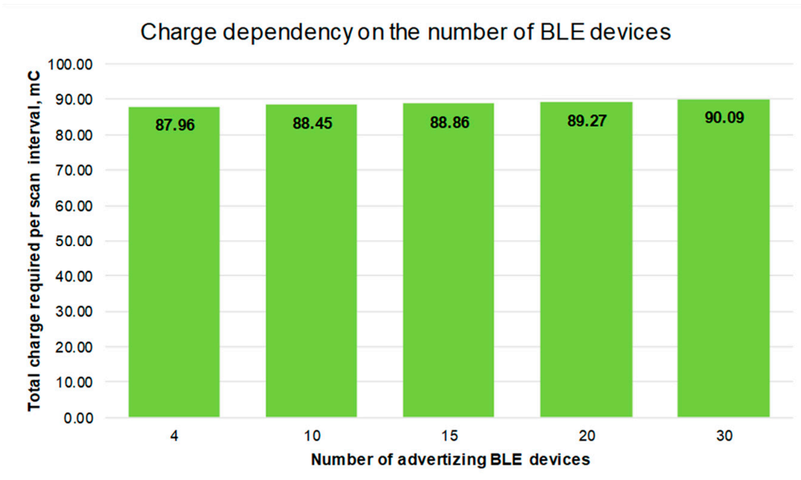

Figure 8. Estimated dependency of the total charge required for one cooperating BLE device during one full scanning interval of 1500 ms on the number of advertising BLE devices.

As shown in Figure 8, even when the number of active and advertising BLE devices increases several times, the energy used for their communication increases only a few percent.

3.2. Energy Requirements for Location Acquisition (GNSS Module)

The GY-GPSV3-NEO-M8N GNSS module was connected to the ESP32 microcontroller using the UART interface to evaluate the energy requirements for acquiring the current location (GPS L1 C/A was used in all experiments). The current was measured at the GNSS module and therefore does not account for the current consumed by the ESP32 itself. The test scenario was as follows: both modules were powered on simultaneously, and the ESP32 microcontroller continuously interrogated the GNSS module until the first location fix was received. The location fix was based on the response provided by the GNSS module libraries, which in turn relied on the availability of signals from at least four satellites. The "power-off" time between measurements ranged from 15 min to several days. We tried to avoid instances of the so-called "hot start" of the GNSS module, because in our application domain the period between consecutive location fix attempts is typically measured in hours. Figure 9 shows the time and charge requirements to acquire fix on current location from the cold start of the GNSS module.

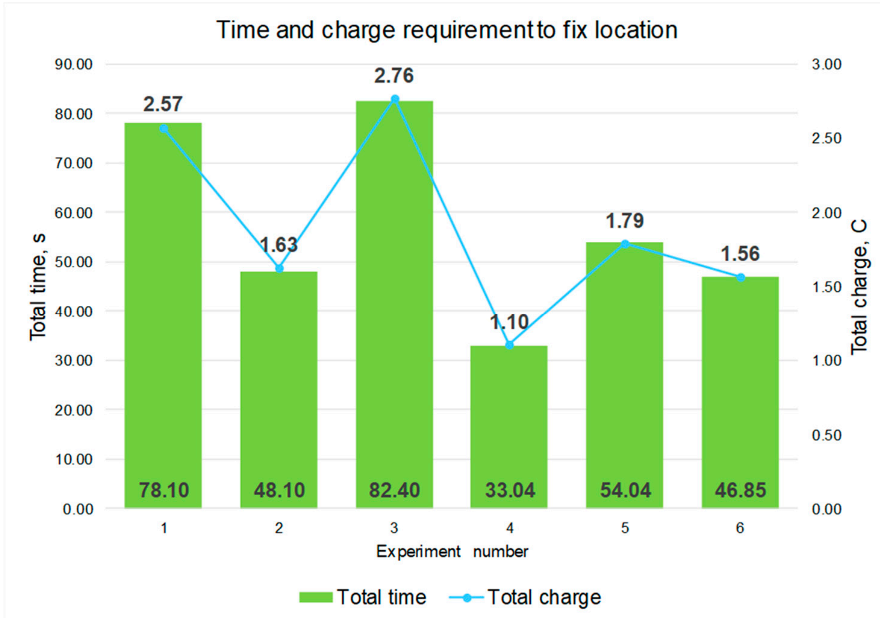

Figure 9. The time and charge requirements to acquire one fix on current location.

Upon analyzing the charts presented in Figure 9 one can draw two conclusions. The time required for a location fix varies significantly, starting from 30 s and going up to several minutes. On the other hand, the total energy consumption required by the GNSS module is proportional to the time taken to acquire the location. Our experiments show that the average current required by the GNSS module is 33.37 mA. The average time to acquire a location fix is 57 s, and the average charge required for this task is 1.90 C. Standard error for the average total position acquisition time and the average total charge requirements for one position acquisition is 7.86 and 0.26, respectively.

3.3. Energy Requirements for Data Transmission (Mobile Communications Module)

To evaluate the energy requirements for data transmission to the central server, we used the SIM7000E communication module connected to the ESP32 microcontroller via the UART interface. The current was measured at the SIM7000E module, so it does not account for the current consumed by the ESP32 itself. The following test scenario was carried out: the SIM7000E module was powered on from the “power off” state, allowed to connect to the communication service provider network, used to transmit 100 bytes of useful data to a web server, and then shut down again into the “power off” state. The energy consumption was measured for the LTE CAT-M1 and 2G (GSM) communication modes. We also evaluated the differences between the use of the plain HTTP and the secure HTTPS protocol. The results are summarized in Figure 10 and Table 2.

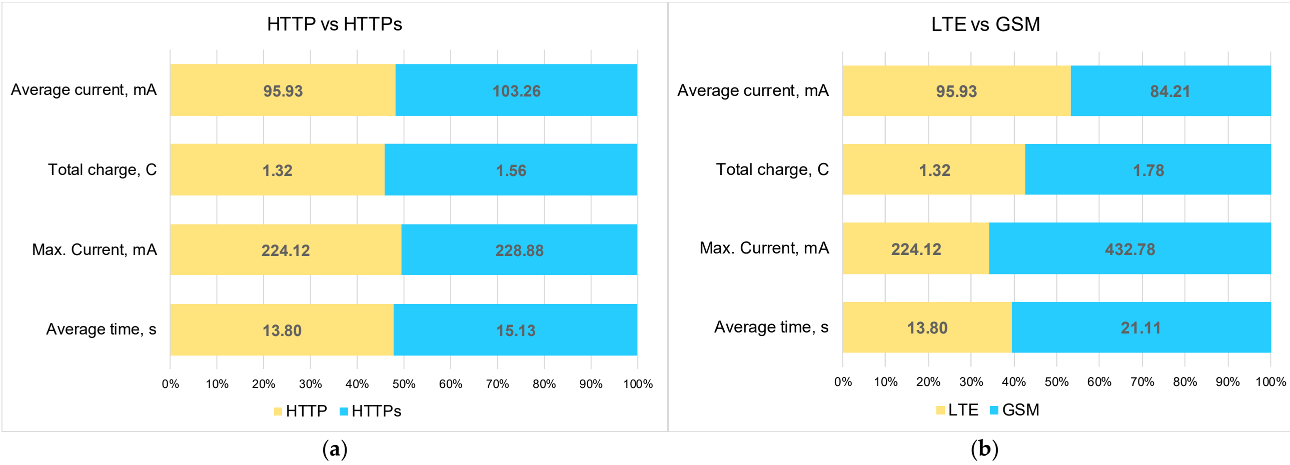

Figure 10. Comparison of time and energy requirements to transfer 100 B of useful data using different protocols: (**a**) Comparison of the energy requirements of the SIM7000E communications module for HTTP and HTTPS data transfer using the LTE CAT-M1 mode; (**b**) Comparison of the energy requirements of the SIM7000E communications module for HTTP data transfer using the LTE CAT-M1 and 2G (GSM) modes.

Table 2. Standard errors of mean values presented in Figure 10.

Protocol	Avg. Current	Total Charge	Max. Current	Avg. Time
HTTP using LTE	1.495	0.033	1.789	0.200
HTTPs using LTE	0.136	0.017	0.595	0.150
HTTP using GSM	2.175	0.046	15.385	0.346

Figure 10a shows that there is no advantage in avoiding the use of the secure HTTPS protocol for network communications. Switching to plain HTTP saves only a small amount of time (about 1.3 s, or 10%) and charge (about 0.24 C, or 18%). On the other hand, these numbers may be significant if maximum battery time is required and data security is ensured using other means. Figure 10b illustrates the drawbacks of the legacy GSM-based protocol. Compared to LTE, it consumes more charge, requires a longer transmission time, and draws nearly twice the peak current, which may negatively affect the battery lifespan. Consequently, the legacy GSM protocol should be used only in scenarios where LTE is not available.

Figure 11 presents a screenshot of the PPK2 software displaying the current waveform of the SIM7000E module during a single communication session. Part "a" corresponds to the power on phase, during which the module transitions from the "power off" state to a state where it can communicate with the host device (i.e., the ESP32) using AT commands. Part "b" stands for the phase in which SIM7000E is trying to register in the service provider data transfer network. The actual data transfer is carried out during part "c". Part "d" stands for the module shutdown phase, in which the module transitions back to the "power off" state, which takes some time after the command to shutdown is issued.

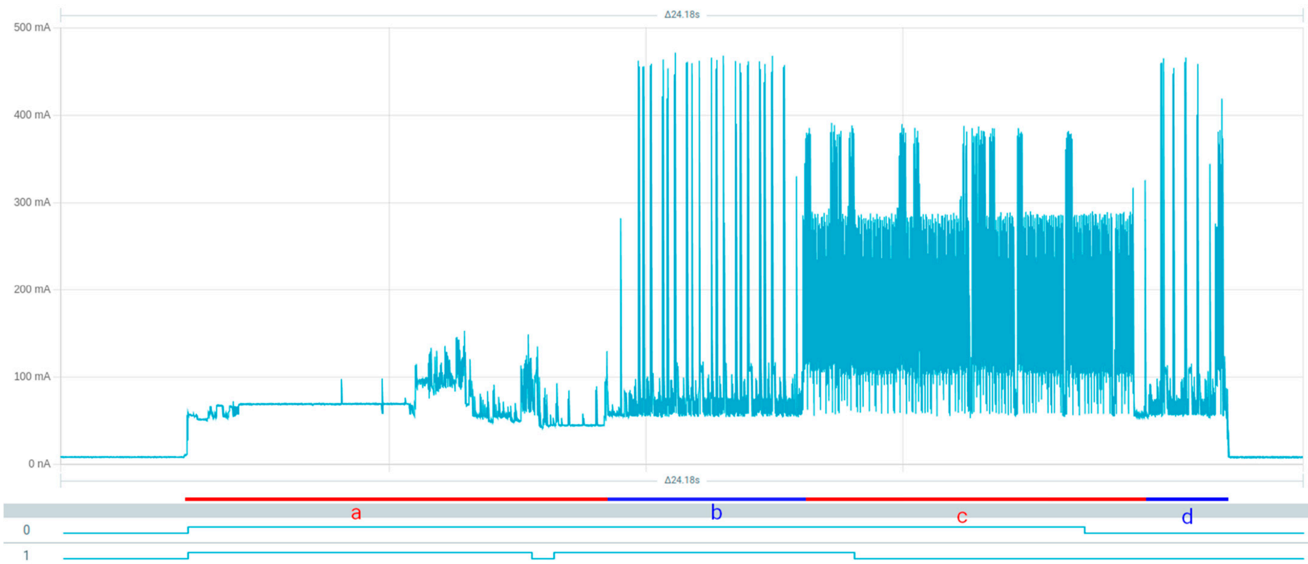

Figure 11. The current consumption waveform for the SIM7000E module while sending 100 B data to the web server.

A summary of energy requirements during the data transfer process is presented in Figure 12. The numbers presented in Figure 12 are extracted from the current waveform using the PPK2 software. One can observe that the power-on process takes the longest—8.46 s, or about 42% of the total time, but requires only 0.56 C, or about 28% of total charge. However, the data transfer phase is the most energy-intensive, accounting for only approximately 32% of the total session duration, but consuming 0.97 C, which represents 49% of the total charge.

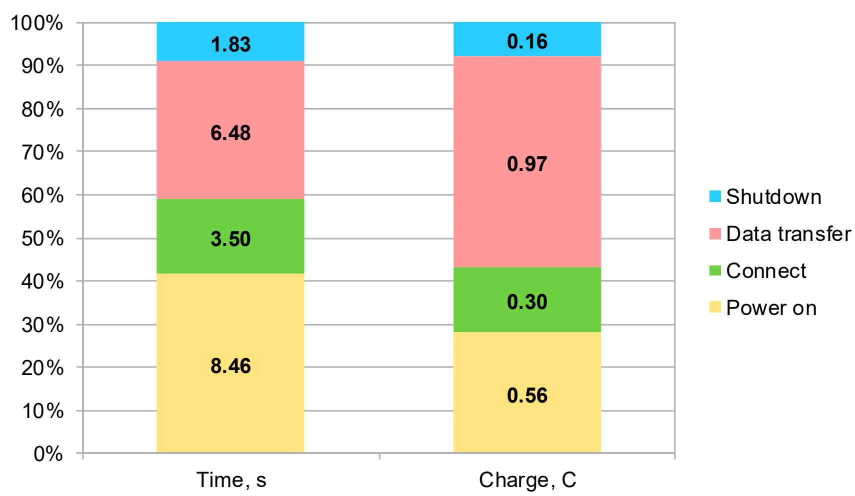

Figure 12. Summary of the energy requirements during the 100 B of location data transfer to the web server.

3.4. Dependence of Energy Requirements on the Number of Cooperating Devices

The first question to evaluate the energy efficiency of the proposed method is how many devices it needs to be more efficient compared to the "traditional" tracking solution when all devices act independently and personally report their locations to the central server. The results of this comparison are provided in Figure 13.

Figure 13. Summary of the average charge requirements for one location report for one device using the proposed energy-saving method.

Figure 13 shows the average charge requirements to send one location report to the central server per device. The red and green lines show the charge required to produce a location report using the "traditional" approach. In this case, each device acts independently. The tracking device wakes up at the predefined time, powers on the GNSS module, waits for location lock, powers down the GNSS module, powers on the mobile communications module, sends location data to the central server using the LTE CAT-M1 mode and the HTTP (or HTTPS) protocol, shuts down the communications module, and goes to sleep until the next communications period. In this case the total charge requirements for one location data report to the HTTP server may be evaluated by the following equations:

$$Q_{gnss} = Q_{fix} + t_{fix} \cdot I_{esp}, \tag{3}$$

$$Q_{lr} = Q_{snd} + t_{snd} \cdot I_{esp}, \tag{4}$$

$$Q_{http} = Q_{gnss} + Q_{lr}, \tag{5}$$

where Q_{fix}—is the total charge required by the GNSS device to fix location, t_{fix}—is the time needed to acquire the location, I_{esp}—is the average current consumed by the ESP32 microcontroller while waiting for other devices to perform their tasks. Q_{gnss}—is the total charge consumed by all devices during the phase of location acquisition. Q_{lr}—is the total charge required by all devices during the phase of location reporting to the server using HTTP protocol. Q_{snd} and t_{snd} are the charge and time required by the communications device to make one location report to the central server. Q_{http}—is the total charge requirement including all phases and all devices in the system to perform one location fix and report cycle. According to the experimental results presented earlier in the corresponding chapters the following values were used: $Q_{fix} = 1.9$ C, $t_{fix} = 57.1$ s, $I_{esp} = 22.15$ mA,

Q_{snd} = 1.324 C, t_{snd} = 13.8 s. To evaluate the total charge requirements in the case of HTTPs protocol (Q_{shttp}), one can replace Q_{snd} and t_{snd} with corresponding values of HTTPs protocol, i. e. Q_{ssnd} = 1.56 C and t_{ssnd} = 15.1 s.

The average charge consumption in this mode is 4.8 C when using the plain HTTP protocol, and 5.1 C when using the HTTPS protocol, and these numbers are not dependent on the total number of devices.

The bars on the graph in Figure 13 represent the average charge requirements for one report on location per one device when the devices are cooperating. In this case, the devices act according to the method described in Section 2.2. Estimation of the average charge requirement in the case of the cooperating devices may be expressed by the following equations:

$$Q_{ble} = \frac{t_{disc} \cdot Q_{int}}{1.5} \tag{6}$$

$$Q_{coop} = Q_{ble} + \frac{Q_{gnss} + Q_{lr}}{n} \tag{7}$$

where Q_{ble} is the total charge required by the whole system for one cycle of neighboring device discovery and advertising using BLE protocol, t_{disc} is the discovery time window during which devices are looking for each other, Q_{int} is the total charge required during one BLE scanning interval of 1.5 s calculated using Equation (1), and n is the total number of cooperating devices. Discovery time window $t_{disc} = 30$ s was used, which we believe is sufficient to overcome potential drift in the internal clocks of the cooperating devices. In this case, only the leader node device uses GNSS and LTE services to report the locations of all cooperating devices, and the leader changes after each cycle.

The analysis of Figure 13 shows that some energy savings could be achieved when at least two devices are in range. However, starting from 10 and more devices, further energy savings are not significantly increasing. The optimal number of devices to save energy is 5 and more. A single device will always lose energy while trying to find nonexistent cooperating devices. The use of the plain HTTP protocol does not give any substantial improvement, especially when the number of cooperating devices is 5 and more.

Figure 14 highlights the reason why energy savings start to increase insignificantly when the number of cooperating devices reaches 10. The values in this figure were calculated using Equations (3)–(7) and separating energy demands of different devices.

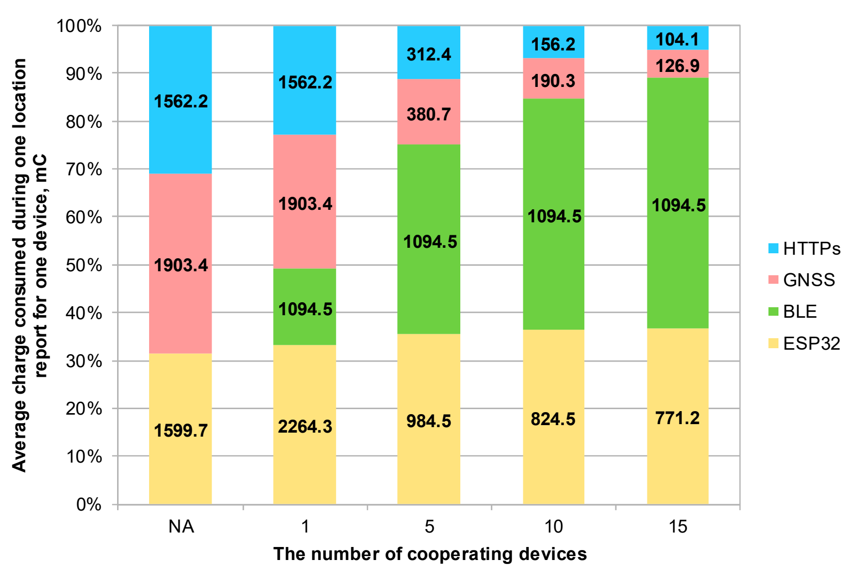

Figure 14. Relative amount of charge required for the different activities of the tracking devices.

This figure shows the relative amount of charge required by the different tasks of the tracking devices. The column labeled "NA" shows the case of the "traditional" approach when each device reports its location to the central server individually and independently. In this case, BLE communications are not used at all. Other columns represent the average charge distributions per device while making one location report. In cases where several devices are cooperating, the relative charge requirements for HTTPS and GNSS services decrease, but, on the other hand, the relative charge requirements for BLE communication increases. These tendencies balance each other when the total number of devices increases and the total amount of charge required for one location report stabilizes (see Figure 13).

The neighbor discovery time window used in the cooperation process was t_{disc} = 30 s and directly influences the amount of energy required for the BLE part of the activities. Figure 15 shows what happens when the different durations of the t_{disc} are used in Equations (6) and (7).

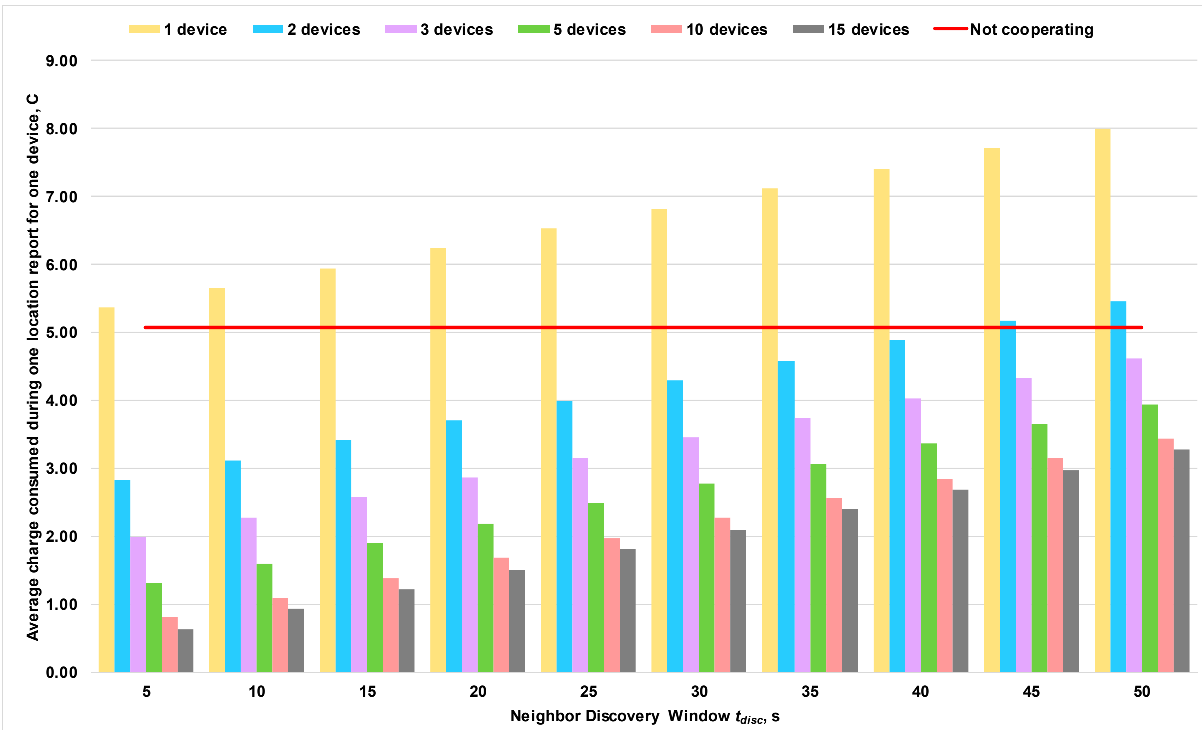

Figure 15. Dependency of the average charge required for one location report on the duration of the t_{disc} window.

One can see that shortening the duration of the neighbor discovery window t_{disc} can slightly increase the energy efficiency of the cooperating devices. However, a short t_{disc} requires very good synchronization of the clocks of all cooperating devices, which is not a trivial task when devices spend most of their time in a deep sleep state.

4. Discussion

In this work, we have evaluated the proposed energy saving method for cooperating nearby battery-powered tracking devices. The main idea of the proposed approach is to save energy by combining the position information of several devices and sending it to the central server using a single session. It is based on previous research results, which show that sensors and data processing modules require significantly less energy compared to communication modules, as concluded in [12]. The proposed method is based on the hypothesis that cooperation between nearby tracking devices and the selection of only one device to report the position of all nearby devices will reduce the average usage of long-range communications modules (mobile communication and GNSS) to a minimum, decreasing the average energy usage of all cooperating trackers. To achieve this a light-weight protocol for the discovery of the cooperating devices and their information exchange was used, based on the BLE scanning/advertising mechanism.

The main challenge of the experimental evaluation was to find the conditions in which the proposed method is more energy efficient than the straightforward approach when each tracker works independently. Evaluation of the energy efficiency of all main hardware modules of the tracking devices revealed the main characteristics and tendencies that lead to reasonable recommendations on how to implement and use the proposed method in practice.

Based on the results of the evaluation of the proposed method, several important findings were revealed:

1. The microcontroller with an active BLE radio subsystem uses significantly more current than while running code and performing calculations (58.7 mA vs. 22.1 mA). To reduce total energy requirements, the BLE radio could be used less. But this leads to the potential risk that tracking devices will not be able to discover each other when their clocks are not synchronized. A 30 s discovery time window starting at the predefined time (i.e., at the beginning of each hour) should be sufficient to compensate for the clock drifts in the cooperating devices. If further energy savings are required, one can try to reduce the time window duration even further and try to compensate for the time drift by synchronizing the time of all cooperating devices. In this case, a slight modification of the BLE-based communication protocol is required, which could be an interesting topic for further research.
2. Experimental evaluation shows that there is only a slight dependency of the energy requirements on the number of nearby BLE devices (including all discoverable devices, not necessarily trackers). Energy requirements may increase by up to 3% when the number of BLE devices increases from 1 to 30 or more. This dependency does not differ according to the type of device, as all nearby BLE devices add some additional energy losses. This means that even in environments with many active random BLE devices, the proposed method should be sufficiently efficient.
3. Experimental evaluation of the GNSS module shows that its energy requirements are not as high as was expected, and they are comparable with those of the active BLE radio, which performs scanning/advertising (33.4 mA average current for the GNSS module vs. 36.6 mA for the BLE radio subsystem alone). The only problem is the stability of the time interval required to acquire the current position. This time interval depends on many external factors, such as physical location of the device, orientation of the antenna, radio interference, etc. The measurements presented in this study were performed under good conditions and represent the "best case scenario" with respect to GNNS energy requirements. If conditions are worse and location fix time is increased, then the efficiency of the proposed method should also be more beneficial compared to the traditional tracking approach. Research revealing dependencies on the efficiency of the proposed algorithm when tracking devices are under more ungrateful conditions (i.e., inside the metal shipping container) could be performed in the future.
4. Evaluation of the mobile communications module shows that there is no significant difference if TLS (Transport Layer Security) based HTTPS or plain HTTP protocols are used, which means that there is no reason to not use the secure protocol. Selecting the secure protocol is best practice. Additionally, the LTE-based communication protocol ensures better energy efficiency compared to legacy GSM, so this protocol should only be used in situations where LTE is not available.

Compared to the traditional tracking method (where each tracking device is working independently), the proposed method allows energy savings of 19% to be achieved when at least two cooperating devices are in range. The average energy requirements to make one report on device location drops from 4.68 mWh to 3.93 mWh. Energy savings increase with the number of devices and reach 240% when 15 devices cooperate. In this case, the average energy for one location report per device drops from 4.68 mWh to 1.92 mWh. However, starting from 10 devices, additional energy savings do not increase significantly (225 percent for 10 cooperating devices vs. 240 percent for 15 devices). The optimal number of devices is 5 or more (providing energy savings starting from 180 percent and more). On the other hand, a single device will always lose some energy while trying to find nonexistent cooperating devices.

Therefore, the results of this study are in line with previous research, proving that communication of wireless devices requires much more energy than local processing, and the minimization of this communication allows one to achieve significant energy savings, as in our proposed method.

5. Conclusions

In this paper, we have presented a cooperative location tracking method for nearby battery-powered tracking devices, which allows them to save battery energy by sharing location reporting tasks. The results presented here confirm that communication tasks require much more energy than the local processing and calculations performed by the CPU alone. The energy requirements of the GNSS module are comparable with those of the active BLE radio: the average current for the GNSS module is 33.4 mA, while the average current for the BLE radio subsystem is 36.6 mA. On the other hand, an average current for the CPU is 22.1 mA, and for most of the communication tasks, the CPU should be active as well. Therefore, it is beneficial to minimize the usage of communication modules (BLE, GNSS, and mobile networks) by sharing location reporting tasks between adjacent tracking devices, as proposed in this study. Our results show that when the number of neighboring tracking devices is two or more, the proposed approach allows the average energy usage of tracking devices to be reduced, compared to the traditional approach, when each tracker reports location independently. The experimental evaluation of the proposed method shows that, compared to the traditional approach, where each tracker sends its location individually, the proposed method allows a reduction in the average battery charge required for one position report from 19% to 240% for each cooperating tracker. The average energy consumption for one location report per node decreased from 4.68 mWh using the traditional approach to 3.93 mWh for 2 cooperating devices and 1.92 mWh for 15 cooperating devices. The proposed method could be applied to mobile asset tracking devices, used by logistics companies around the world, allowing them to prolong the life of their battery-powered trackers. The beneficial effect of the proposed method can be achieved in the cases when multiple assets with attached trackers are transported together, which would be a quite frequent case in parcel delivery in urban areas, for example.

To achieve greater energy savings, future studies could further investigate reducing the duration of the neighbor discovery time window. To address any resulting time drift, it would be necessary to synchronize the clocks of all cooperating devices. This may require minor adjustments to the current BLE-based communication protocol, offering an opportunity to expand the scope of future studies. Another direction for future research is to evaluate the performance of the method under less favorable conditions. When location fix times are longer, the proposed approach is expected to outperform traditional tracking methods even more. Therefore, future studies could examine its effectiveness in environments such as metal shipping containers or areas with weak and unstable GPS signals to better understand the impact of environmental factors.

Author Contributions: Conceptualization, A.L. and L.K.; methodology, N.M. and A.L.; software, N.M., L.K. and R.B.; validation, N.M., A.L., L.K., A.J. and R.B.; formal analysis, A.L., A.J. and R.B.; investigation, N.M. and R.B.; resources, A.L. and A.J.; writing—original draft preparation, N.M., A.L., L.K., A.J. and R.B.; writing—review and editing, N.M., A.L., L.K., A.J. and R.B.; visualization, N.M., A.L., L.K. and A.J.; supervision, A.L.; project administration, A.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Capua, M.D.; Ciaramella, A.; De Prisco, A. Machine Learning and Computer Vision for the Automation of Processes in Advanced Logistics: The Integrated Logistic Platform (ILP) 4.0. *Procedia Comput. Sci.* **2023**, *217*, 326–338. https://doi.org/10.1016/j.procs.2022.12.228.
2. Charles, P.; Tamgole, P.; Dugane, A.; Sagar, V. A Literature Survey on Automated Cargo Tracking System. *Int. J. Res. Appl. Sci. Eng. Technol.* **2022**, *10*, 1300–1302. https://doi.org/10.22214/ijraset.2022.42450.
3. Mu, X.; Antwi-Afari, M.F. The Applications of Internet of Things (IoT) in Industrial Management: A Science Mapping Review. *Int. J. Prod. Res.* **2024**, *62*, 1928–1952. https://doi.org/10.1080/00207543.2023.2290229.
4. Udeh, E.O.; Amajuoyi, P.; Adeusi, K.B.; Scott, A.O. The Role of IoT in Boosting Supply Chain Transparency and Efficiency. *Magna Sci. Adv. Res. Rev.* **2024**, *12*, 178–197. https://doi.org/10.30574/msarr.2024.11.1.0081.
5. Ding, Y.; Jin, M.; Li, S.; Feng, D. Smart Logistics Based on the Internet of Things Technology: An Overview. *Int. J. Logist. Res. Appl.* **2021**, *24*, 323–345. https://doi.org/10.1080/13675567.2020.1757053.
6. Song, Y.; Yu, F.R.; Zhou, L.; Yang, X.; He, Z. Applications of the Internet of Things (IoT) in Smart Logistics: A Comprehensive Survey. *IEEE Internet Things J.* **2021**, *8*, 4250–4274. https://doi.org/10.1109/JIOT.2020.3034385.
7. Ravi Chandra, B.; Kumar, K.; Roy, A.; Chandra, I.S. Overview of Internet of Things-Based Smart Logistics Systems. In *Communication Technologies and Security Challenges in IoT*; Prasad, A., Singh, T.P., Dwivedi Sharma, S., Eds.; Internet of Things; Springer Nature Singapore: Singapore, 2024; pp. 241–259, ISBN 978-981-97-0051-6.
8. Alzahrani, B.A.; Irshad, A. An Improved IoT/RFID-Enabled Object Tracking and Authentication Scheme for Smart Logistics. *Wirel. Pers. Commun* **2023**, *129*, 399–422. https://doi.org/10.1007/s11277-022-10103-7.
9. Kalkha, H.; Khiat, A.; Bahnasse, A.; Ouajji, H. The Rising Trends of Smart E-Commerce Logistics. *IEEE Access* **2023**, *11*, 33839–33857. https://doi.org/10.1109/ACCESS.2023.3252566.
10. Lagorio, A.; Cimini, C.; Pinto, R.; Cavalieri, S. 5G in Logistics 4.0: Potential Applications and Challenges. *Procedia Comput. Sci.* **2023**, *217*, 650–659. https://doi.org/10.1016/j.procs.2022.12.261.
11. Singh, J.; Kaur, R.; Singh, D. A Survey and Taxonomy on Energy Management Schemes in Wireless Sensor Networks. *J. Syst. Archit.* **2020**, *111*, 101782. https://doi.org/10.1016/j.sysarc.2020.101782.
12. Hassan, M.F.; Al-Musawi, B.; Al-Janabi, A.K. Innovative Fitness Functions for Robust Energy Management in WSNs. *J. Netw. Syst. Manag.* **2023**, *31*, 76. https://doi.org/10.1007/s10922-023-09768-7.
13. Poornima, G.; Parthasarathy, C.; Umapathy, K.; Dinesh Kumar, T.; Archana, M.A.; Kumar, N.A. GPS Based Smart Vehicle Tracking and Monitoring System. In Proceedings of the 2024 Second International Conference on Intelligent Cyber Physical Systems and Internet of Things (ICoICI), Coimbatore, India, 28–30 August 2024; IEEE: Piscataway, NJ, USA, 2024; pp. 506–511.
14. Degadwala, S.; Upadhyay, R.; Upadhyay, S.; Dave, S.S.; Mahida, D.; Vyas, D. Enhancing Fleet Management with ESP8266-Based IoT Sensors for Weight and Location Tracking. In Proceedings of the 2023 3rd International Conference on Innovative Mechanisms for Industry Applications (ICIMIA), Bengaluru, India, 21–23 December 2023; IEEE: Piscataway, NJ, USA, 2023; pp. 13–17.

15. Jacob, R.O.; Alyaan, S.M.; Nikitha, K.; Niranjana Murthy, H.S. IoT Based GPS Tracking System with SOS Capabilities. In Proceedings of the 2022 International Mobile and Embedded Technology Conference (MECON), Noida, India, 10–11 March 2022; IEEE: Piscataway, NJ, USA, 2022; pp. 72–75.
16. Hagargund, A.G.; Shreya, P.; Spandana, N.; Varsha, D.; Vishrutha, V.S. Implementation of Global Ship Tracking and Monitor System. In Proceedings of the 2022 IEEE North Karnataka Subsection Flagship International Conference (NKCon), Vijaypur, India, 20–21 November 2022; IEEE: Piscataway, NJ, USA, 2022; pp. 1–6.
17. Qadir, A.M.; Cooper, P. GPS-Based Mobile Cross-Platform Cargo Tracking System with Web-Based Application. In Proceedings of the 2020 8th International Symposium on Digital Forensics and Security (ISDFS), Beirut, Lebanon, 1–2 June 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 1–7.
18. Sahana, B.; Jayanthi, P.N.; Muktha, V.N.; Natasha, D.A. UAV Tracking Using 5G. In Proceedings of the 2024 8th International Conference on Computational System and Information Technology for Sustainable Solutions (CSITSS); IEEE: Bengaluru, India, 7–9 November 2024; IEEE: Piscataway, NJ, USA, 2024; pp. 1–3.
19. Kaiser, J.; Hernández, M.P.; Kaupe, V.; Kurrek, P.; McFarlane, D. An Agent-Based Approach for Energy-Efficient Sensor Networks in Logistics. *Eng. Appl. Artif. Intell.* **2024**, *127*, 107198. https://doi.org/10.1016/j.engappai.2023.107198.
20. Zhang, C.; Fei, S. Energy Efficient Target Tracking Algorithm Using Cooperative Sensors. *J. Syst. Eng. Electron.* **2012**, *23*, 640–648. https://doi.org/10.1109/JSEE.2012.00080.
21. Liu, W.; He, Y.; Zhang, X.; Jiang, F.; Gao, K.; Xiao, J. Energy-Efficient Node Scheduling Method for Cooperative Target Tracking in Wireless Sensor Networks. *Math. Probl. Eng.* **2015**, *2015*, 627479.
22. Dev, J.; Mishra, J. Energy-Efficient Object Detection and Tracking Framework for Wireless Sensor Network. *Sensors* **2023**, *23*, 746. https://doi.org/10.3390/s23020746.
23. Kumar, V.; Bergmann, N.W.; Ahmad, I.; Jurdak, R.; Kusy, B. Cluster-Based Position Tracking of Mobile Sensors. In Proceedings of the 2016 IEEE Conference on Wireless Sensors (ICWiSE), Langkawi, Malaysia, 10–12 October 2016; pp. 7–14.
24. Anitha, C.; Karthik, G.M.; Uma Maheswari, M.; Kumar, L.V.; Sasikala, G.; Jayanthi, S. A Robust Design of Cooperative Model to Enhance Energy Efficiency over Heterogeneous 5G Wireless Sensor Network Paradigm. In Proceedings of the 2024 Asia Pacific Conference on Innovation in Technology (APCIT), Mysore, India. 26–27 July 2024; pp. 1–6.
25. Zheng, K.; Luo, R.; Liu, X.; Qiu, J.; Liu, J. Distributed DDPG-Based Resource Allocation for Age of Information Minimization in Mobile Wireless-Powered Internet of Things. *IEEE Internet Things J.* **2024**, *11*, 29102–29115. https://doi.org/10.1109/JIOT.2024.3406044.
26. Espressif Systems. ESP32-C6 Esp-Dev-Kits Documentation. Available online: https://docs.espressif.com/projects/esp-dev-kits/en/latest/esp32c6/esp-dev-kits-en-master-esp32c6.pdf (accessed on 15 November 2025).
27. Ublox NEO-8M GPS Modulis. Available online: https://www.duino.lt/gps/16339-arduino-ublox-neo-8m-gps-modulis.html (accessed on 15 November 2025).
28. u-Blox AG. NEO-M8 u-Blox M8 Concurrent GNSS Modules Data Sheet. Available online: https://www.u-blox.com/sites/default/files/NEO-M8-FW3_DataSheet_UBX-15031086.pdf (accessed on 15 November 2025).
29. Waveshare. SIM7000E NB-IOT HAT. Available online: https://files.waveshare.com/upload/7/76/SIM7000E-NB-IoT-HAT-Manual-EN.pdf (accessed on 15 November 2025).
30. SIMCom Wireless Solutions Co., Ltd. SIM7000E eMTC/NB-IoT/EDGE Module. Available online: https://simcom.ee/documents/SIM7000E/SIM7000E_SPEC_2017-9-21.pdf (accessed on 15 November 2025).
31. Nordic Semiconductor ASA. Power Profiler Kit II. Available online: https://nsscprodmedia.blob.core.windows.net/prod/software-and-other-downloads/product-briefs/power-profiler-kit-ii-pbv10.pdf (accessed on 15 November 2025).
32. Espressif Systems. ESP-IDF Programming Guide. Available online: https://docs.espressif.com/projects/esp-idf/en/v3.3.2/index.html (accessed on 15 November 2025).
33. Botletics, LLC. BotleticsSIM7000 Library. Available online: https://github.com/botletics/SIM7000-LTE-Shield/wiki/Library-Functions (accessed on 15 November 2025).
34. Tosi, J.; Taffoni, F.; Santacatterina, M.; Sannino, R.; Formica, D. Performance Evaluation of Bluetooth Low Energy: A Systematic Review. *Sensors* **2017**, *17*, 2898. https://doi.org/10.3390/s17122898.