



HAL
open science

Snowball, an asynchronous probabilistic protocol for neighbour discovery in mobile BLE network.

Clément Rup, Eddy Bajic, Kais Mekki

► **To cite this version:**

Clément Rup, Eddy Bajic, Kais Mekki. Snowball, an asynchronous probabilistic protocol for neighbour discovery in mobile BLE network.. 8th IEEE World Forum on Internet of Things, WFIoT2022, Oct 2022, Yokohama, Japan. pp.1-8, 10.1109/WF-IoT54382.2022.10152274 . hal-03823749

HAL Id: hal-03823749

<https://hal.science/hal-03823749>

Submitted on 4 May 2023

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.

Snowball: An Asynchronous Probabilistic Protocol for Neighbour Discovery in Mobile BLE Network

Clément Rup
CRAN CNRS UMR 7039
University of Lorraine
Nancy, France
clement.rup@univ-lorraine.fr

Eddy Bajic
CRAN CNRS UMR 7039
University of Lorraine
Nancy, France
eddy.bajic@univ-lorraine.fr

Kais Mekki
Research and Development
OKKO SAS
Rémering-les-Puttelange, France
kais.mekki@okko-france.com

Abstract— Neighbour discovery is crucial for mobile ad-hoc networks and especially in Internet of Things (IoT) networks which are distributed by nature and built on mobile, communicating, and mostly battery-powered autonomous IoT devices. The challenge of efficient neighbour discovery in such architecture is to achieve both low latency and high energy efficiency discovery process, which have been shown to be major issues in the scientific community. In this paper, we contribute to the resolution of this problem applied on Bluetooth Low Energy (BLE) personal area network. A new neighbour discovery protocol called Snowball is proposed, which leverages the advertising and scanning features of BLE beaconing mode. Snowball provides new delay parameters upon the BLE stack to achieve a stable duty cycle schedule in the neighbourhood, which guarantee an efficient and low energy discovery process. The performance of Snowball is discussed and evaluated by simulation using Netlogo, and also by a practical experimentation using ESP32-based IoT devices.

Keywords— Internet of Things; Mobile Ad-hoc Network; Bluetooth Low Energy; Neighbour Discovery

I. INTRODUCTION

Nowadays, mobile ad-hoc networks are widely deployed as sensing and monitoring tools for asset tracking, smart environments, and industrial risk supervision. They are designed using wireless mobile sensor devices which can either be autonomous or attached to mobile entities such as container in warehouses [1]. Its topology is always not known as mobile devices can continuously join or exit the network area. Several mobile devices come together during a period of time and then it disaggregates. As example in warehouse monitoring application, containers are periodically moved and there is always new entering or exiting containers [1].

Therefore, to ensures its functionalities, the mobile ad-hoc networks employ a main mechanism called Neighbours Discovery [2]. In this mechanism, mobile devices need to discover each other and continuously detect new neighbours, maintain group community, and be aware of neighbour departure, typically under latency and energy constraints (i.e. all the devices are typically battery powered).

To affront the energy consumption constraint, mobile devices work usually in duty cycle mode where they continuously switch between active and sleep mode. During the active mode, devices communicate with its surrounding devices to discover the identity of its neighbours. Each devices needs a long duty cycle to save its battery life since it may spend long periods without any existing neighbour. However, when they have nearby neighbours, the device needs low duty cycle to quickly discover its surrounding devices. To achieve this, devices must carefully and relevantly trade off duty cycling and latency. As a consequence, the neighbour discovery mechanism could be energy efficient, reactive (i.e. has a low discovery latency), and adaptive to network scales [3].

In the literature, neighbour discovery is mostly proposed under a wireless technology such as UHF, WIFI, and Zigbee [2]. In this paper, we propose a new protocol designed for Bluetooth Low Energy (BLE).

BLE offers two communication mode: connected mode and advertising mode (called also beaconing mode). In the first one, devices work in master-slave mode. It allows a central master device (for example, a smartphone) to connect and interact with services implemented in the slave device. The master device scan for nearby slave devices. Once a slave device is found, its capabilities are discovered by connecting to its GATT (Generic ATtribute Profile) server [4].

In the second one, the devices work in a fully broadcasting mode. It simply exchanges information that is stored internally in each device. It does not active any connection capabilities, and it achieves the lowest possible power consumption by simply waking up, broadcast an advertise, and then going back to sleep.

In the literature, several solutions were attempted to propose a neighbours discovery solution under the BLE connected mode as discussed in [5]. However, to our knowledge, there is no solution was proposed for the BLE advertising mode. In this paper, we work in this unexploited research area, and we propose a new neighbour discovery protocol for a mobile BLE network. We called

this protocol “Snowball”. Snowball operates using two phases: neighbour discovery phase where each device builds its neighbourhood table, and neighbour group maintenance where duty cycle schedule is managed for aggregating neighbourhood tables into a unique scheduled group. The performance of Snowball is discussed and evaluated by simulation using Netlogo, and also by a practical experimentation using ESP32-based IoT devices.

The rest of the paper is organized as follows. First, the state of the art is presented. Then, Snowball protocol is detailed. Finally, simulation and experimental results are discussed.

II. STATE OF THE ART

In the beginning of mobile ad-hoc networks, a basic solution was proposed for neighbour discovery problem: all devices are always listening, and each device periodically broadcasts a “Hello” message to announce its presence. Then, when a device receives a “Hello” message, it considers the sender device as a neighbour, whose distance can be estimated roughly with the received signal strength indicator (RSSI) [3]. Such solution works well but are not energy efficient since all devices must always be listening for incoming “Hello” message. Energy effectiveness is a primordial and relevant factor in mobile ad-hoc networks where all devices are mostly battery powered [2].

To handle the energy factor, the duty-cycle mechanism is used in the literature which make the neighbours discovery more challenging. In compliance with the duty-cycle principle, each device switches between three states to preserve its energy and not pollute the bandwidth: listen, transmit and sleep. Thus, the challenge is that all devices must be awake at the same time for listening and broadcasting messages in order to accomplish neighbour discovery. In the literature two approaches are proposed: synchronized and asynchronous.

In the synchronized approach, the discovery process is performed using an external synchronization tool. So that the duty cycle rhythm and the wake-up time of all devices is known. In [6], the devices clock is synchronized using GPS information, and in [7] the wake-up time of devices is synchronized using NTP (Network Time Protocol) requests.

In the asynchronous approach, the neighbour discovery problem is more complex and challenging. Obviously, a device cannot predict when the neighbours are awake, and a device cannot even know whether any neighbour is present. In the literature, various algorithms are proposed for asynchronous neighbour discovery [8][9][10]. Research works can be categorized according to two approaches: probabilistic and deterministic [2].

The probabilistic approach aims to randomly switch the device between the sleep, listen, and transmit states. It was shown to provide a higher average neighbour discovery rate, but it suffers from an unbounded discovery latency [11]. In deterministic approach, the protocols select active states based on prime numbers or “quorum” techniques [12], so that it is able to guarantee an upper bound on discovery latency, while the choice of parameters (e.g. prime numbers) is often limited.

In this paper, we propose an asynchronous probabilistic neighbour discovery protocol for mobile ad-hoc network. Our proposal is an asynchronous probabilistic solution which is different from existing protocols as it combines initial probabilistic encounters and continuous statistics predictions to establish an efficient connection-less communication network between a swarm of broadcasting devices. The characteristic of our proposal is similar to the natural process that makes fireflies flashing in unison by observing each other’s [13]. The nature-inspired approach of our proposal improves the discovery rate performance as well as latency at each random encounter, thus creating a snowball effect. This effect generates a neighbour discovery synchronisation without any connection or external clock. The snowball protocol is developed for BLE technology and is based on low energy communication using the BLE beaconing advertising mode. To our knowledge, this research work is the first contribution to the development of an asynchronous neighbour discovery algorithm using advertising BLE mode.

In the next sections, the proposed protocol called *Snowball* is detailed.

III. BLUETOOTH LOW ENERGY COMMUNICATION

BLE is an energy-efficient wireless technology standard developed by the Bluetooth Special Interest group (SIG). BLE operates in the 2.4 GHz frequency band with 40 channels, 2 MHz bandwidth, and 1 Mbps output. BLE can work either in connected mode (i.e. master-slave connection between one master and up to seven slaves, on 37 channels) or advertising mode on three channels (i.e. channels number 37, 38, and 39).

In this study, we will focus on the advertising mode. It offers the following advantage *i)* there is no connections required between devices *ii)* there is no interference with others wireless protocols like Wi-Fi *iii)* it offers RSSI capabilities that can be used as a localisation tool [14].

The structure of the BLE frame is shown in Fig. 1. Our Snowball protocol exploits all the 31 bytes of the available *Broadcast Data*.

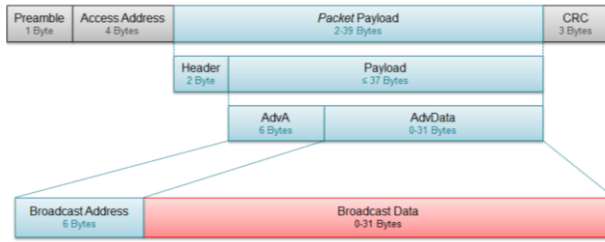


Fig. 1. Structure of the BLE frame [15].

A. BLE Advertising

The BLE advertising mechanism is presented in Fig. 2, it works as follow. First, the device broadcasts an *Advertising Event* (*advEvent*) in each BLE channel (i.e. BLE uses three channel for advertising mode as presented above). Regarding to the structure of BLE frame, the length of the *Broadcast Data* (31 bytes), and the BLE output of 1 Mbps, we can conclude that the duration of *advEvent* is 1.428 ms, as shown in Then, the device waits during an *Advertising Interval* (*advInterval*). The *advInterval* is set between 20 ms and 10.24 s (in units of 625 μ s). Next, the device adds a random *Advertisement Delay* (*advDelay*) between 0 and 10 ms to avoid wireless communication collisions as shown in Fig. 2. Finally, this process is repeated until the expiration of an *Advertisement Duration* (*advDuration*).

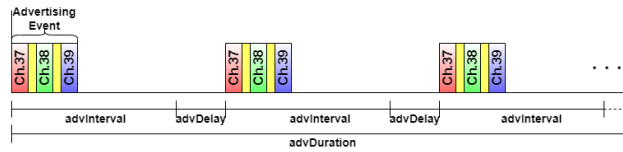


Fig. 2. BLE advertising process.

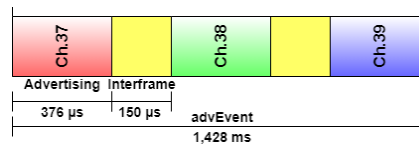


Fig. 3. Advertising event process.

B. BLE Scanning

The BLE device is able to scan the three advertising channels of its neighbours using the following mechanism.

First, the device starts a *Scan Window* (*scanWindow*) during less than 10.24 s (in units of 625 μ s). Then, it waits during a *Scan Interval* (*scanInterval*). This process is repeated for each advertising channel until the expiration of the *Scan Duration* (*scanDuration*) as shown in Fig. 4. Obviously, the duration of *scanWindow* must be longer than the duration of *advEvent* to achieve BLE scanning.

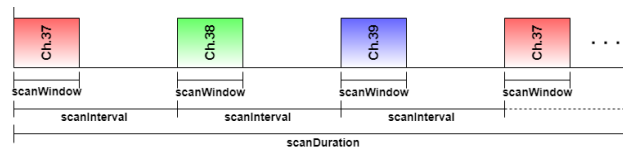


Fig. 4. BLE scanning process.

In our Snowball protocol, the BLE device switches quickly between Advertising and Scanning. To prevent overlap of these operations on the device, a set of constraint equations and mechanisms has been defined which will be described in the next section.

IV. SNOWBALL PROTOCOL

The aim of our Snowball protocol is to propose a neighbour discovery solution for a battery powered BLE devices in mobile ad-hoc network. Each device is autonomous and works independently from other devices without establishing any BLE connection as discussed above (i.e. fully advertising mode). Thus, the energy consumption is the major factor for long lifetime BLE devices. The reduction of the energy consumption of the BLE device is mainly done by the duty cycle mechanism as discussed in section II.

Devices in a BLE network generally operate at a low duty-cycle (<1%) [16]. Another constraint is the data needed by Snowball protocol, only 31 bytes of data being available. Snowball integrates two data fields in the *Broadcast Data*:

- *-ID* field: it contains the identifier of the device that sent the advertisement.
- *-Sync* field: it contains the last sleep duration (*sleepDuration* in seconds) of the device that sent the advertisement. It is used to share duty cycle schedules between devices.

Moreover, Snowball has to cope with mobile network. No device knows in advance the size of the network. Moreover, each device must be able to join or leave the network at any time. Therefore, Snowball keeps track of the scanned neighbours in each device. This neighbour's community is stored in the device as a table of *ID* and scan timestamps. Each *ID* has also a Time-To-Live (*TTL*) in the table.

Snowball works in three phases. The first one is the Commissioning phase. The second one is the Neighbours Aggregation phase, that will roughly give the same activity schedule to all neighbours. The third phase is the Group Cohesion, in which the individual schedules are brought and kept closer.

A. Commissioning

We define the commissioning as the first entering of the device in the BLE network. We assume that devices are commissioned at random times. Their initial behaviour is described in Fig. 5.

Devices operate at a low duty-cycle. It switches between two modes: an active mode where it communicates and process data, and a sleep mode where they save battery. The duration of the active mode and sleep mode are defined as *bleDuration* (> 1 second) and *sleepDuration* respectively.

When a node is commissioned, it starts BLE advertising and scanning to discover its neighbourhood. When the device does not receive any BLE advertisement, it is considered as alone as shown in Fig. 5. Thus, the device sleeps during a random duration named *sleepDuration* which respects two requirements:

- *sleepDuration* is obtained from a random Uniform Distribution that choses a prime number in seconds. This way, the probability of a successful communication between devices is increased as discussed in [16].
- *sleepDuration* is bounded by a *maxSleepDuration* parameter.

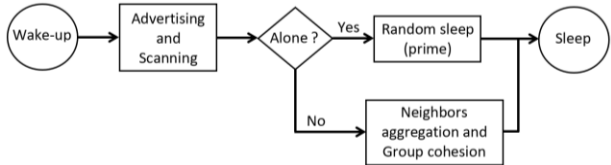


Fig. 5. State diagram of device behaviour.

However, if the device receives an advertisement from another device, it is considered not alone, and the Neighbours Aggregation phase is executed as shown in Fig. 5.

B. Neighbours Aggregation

Before switching to sleep mode, the device processes all the received advertisement *Broadcast Data*. It calculates the mean ($\mu_{sleepDuration}$) and the standard-deviation ($\sigma_{sleepDuration}$) of the neighbours *sleepDuration*. Using these results, the device decides if neighbour communications are too scattered.

If $\sigma_{sleepDuration} > \frac{bleDuration}{4}$, the device needs to aggregate with its neighbours. To do so, a new *sleepDuration* is generated by a Poisson Distribution Law using $\mu_{sleepDuration}$ as a parameter. The new *sleepDuration* is a statistical prediction of the most plausible duration before other devices switch to active mode.

As a result of this aggregation phase, groups are formed among the neighbours. Each group communicates in the same timeframe and share the same *sleepDuration*. The bigger a group, the bigger its influence on smaller ones. A snowball effect takes place, eventually aggregating devices in increasingly bigger groups, until all the neighbours are in the same group (i.e same communication schedule).

However, if $\sigma_{sleepDuration} < \frac{bleDuration}{4}$, the device is already in a group, the Poisson Distribution is useless and lead quickly to the group destruction. In response to that phenomenon, when the device is in a group, the Group Cohesion phase begin.

C. Group Cohesion

To maintain the group, a cohesion method is required. To estimate togetherness, all received advertisements are stored by the device. The temporal repartition of the stored advertisements is used to maintain cohesion.

As shown in Fig. 6, when a group is communicating, the time distribution of advertisements forms a communication spike. The devices affecting the group cohesion are either early or late on the spike. An early device starts to communicate before the majority of its group, and thus, the frequency of reception will increase toward the end of its activity period and vice versa.

A device can know its situation by averaging the reception time of the advertisements. It then uses the standard deviation and an Exponential Distribution to create a correction factor. That factor is added to $\mu_{sleepDuration}$ to compensate the earliness or lateness of the device.

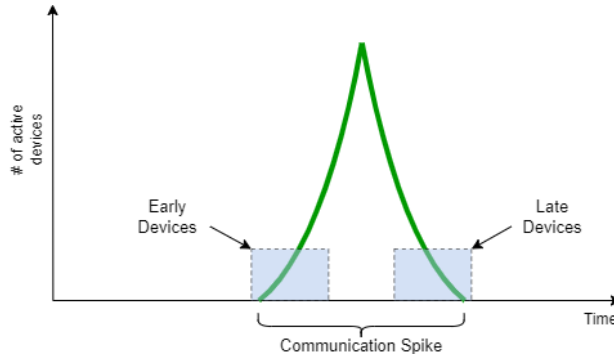


Fig. 6. Time distribution of a group's advertisements.

The communication spike created by those devices can create collisions problems. That requires a dynamic tuning of the BLE parameters.

D. BLE parameters tuning

Snowball uses constraint equations to allow a device to alternatively advertise and scan during its active mode.

a) Constraint equations

In addition to the durations described in section III, two new durations are introduced:

- *switchDuration*: it describes the time taken by the BLE controller to switch between advertising and scanning.
- *scanOffset*: it describes the time between the start of *advDuration* and *scanDuration*. Fig. 7 presents the overall communication chronogram in each device during its active mode.

The constraint equations can be found in TABLE 1 and TABLE 2. Three variables are used:

- The duration *U*: it is equals to the duration of *advEvent* (rounded to 1.5 ms).
- The Coefficient *X*: it is the ratio between *scanWindow* and *U*.
- *N*: it is the number of advertisements events (or intervals) in a *bleDuration*.

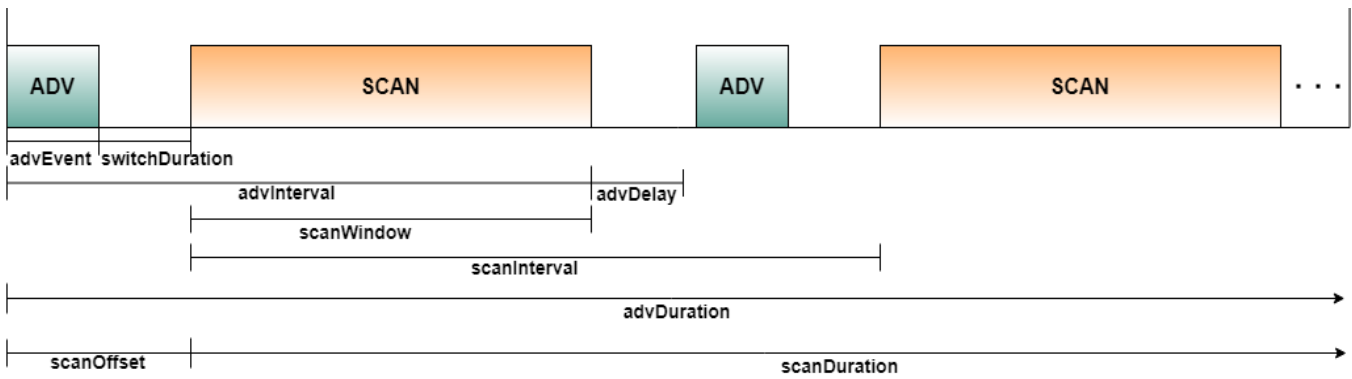


Fig. 7. Interleaved advertising and scanning phases of BLE beaconing.

TABLE 1. INTERLEAVED BLE CONSTRAINTS EQUATIONS.

Duration	Equation
$advEvent$	$advEvent = U$
$switchDuration$	$switchDuration = U$
$advDelay_{Max}$	$advDelay_{Max} = 10\text{ ms}$
$scanWindow$	$X \times U \leq advInterval - 2U - 10$
$advInterval$	$X \times U + 2U + 10 \leq advInterval$
$scanInterval$	$X \times U + 2U + 10 \leq scanInterval$
$advDuration$	$advDuration = N \times advInterval$
$scanDuration$	$scanDuration = N \times scanInterval$
$scanOffset$	$2U \leq scanOffset$

TABLE 2. INTERLEAVED BLE TUNING PARAMETERS.

Tuning parameter	Description
$bleDuration$	Communication duration
N	Number of advertisement events

Using the two parameters in TABLE 2 and keeping in mind (1), the BLE communication can be tuned. A demonstration of the validity of all these equations will be discussed in section V.

$$N_{max} = \frac{bleDuration}{3 \times U + 10} \quad (1)$$

b) Listening and Yelling devices

At commissioning step, the $bleDuration$ is fixed and common to all devices. To tune the BLE parameters, only N is left.

When a device switches to active mode and was in a group in its previous cycle, it varies N . It uses a Bathtub-shaped Distribution as shown in Fig. 8. This distribution creates *Listening Devices* that will advertise less and scan more and *Yelling Devices* that will scan less and advertise more. The population of *Listening Devices* and *Yelling Devices* is randomly distributed at each cycle.

Devices are randomly limiting collisions by choosing to either *Listen* or *Yell* each cycle. They can still change their behaviour in the next cycle. Devices have a memory of their previous communities, with a tunable retention time. That mechanism is used to account for the possible lack of communication during a cycle. Therefore, *Listen* or *Yell* will improve the reliability of Snowball.

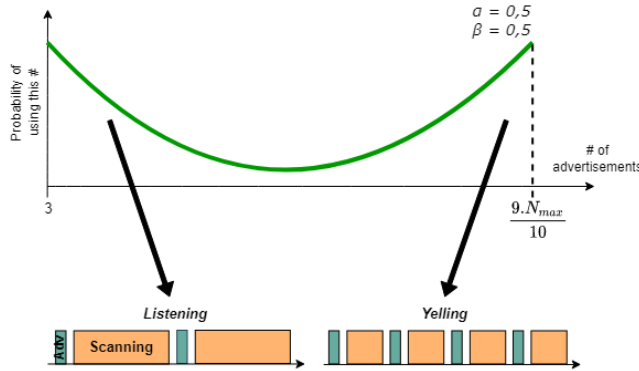


Fig. 8. Probability distribution of listening and yelling.

V. SIMULATION OF SNOWBALL PROTOCOL

Netlogo is used to simulate Snowball protocol. Netlogo [17] is a multi-agent programmable modelling and simulation environment. It permits a bottom-up approach for modelling, where rules are followed individually by agents and the general behaviour of the system arises from the sum of the local behaviours of the involved entities as presented in [18]. This kind of behaviour is exactly what we expect to achieve for a relevant simulation of Snowball neighbor discovery protocol.

A. Protocol implementation with NetLogo

In our model, each BLE device is implemented as an agent (or turtle). It possesses a set of characteristics and variables analogous to the ones of a real device. These agents and their characteristics are used as parameters to procedures that implements BLE communication (using the information given in III) and our protocol. A dashboard is created to monitor several aspects of the simulation, with various levels of insight. The simulation is made with the following conditions:

- Ten devices are placed in communication range of each other's.
- All devices are turned on randomly between $t = 0$ second and $t = 120$ seconds (corresponding to the *maxSleepDuration* they are allowed).
- The activity of all devices is monitored with a time step of 0.5 millisecond.
- All devices share the same parameters.

B. Simulation results

Using the above defined conditions, a neighbourhood is simulated for 15 minutes. The results obtained are shown in APPENDIX A.

Fig. 9.1 shows the devices activity (by monitoring their scan) through time. Also, it shows the number of devices that are alone and in a group. At the start, all devices are alone but as they encounter each other randomly, the Neighbour Aggregation takes place. After that, the first groups are formed and almost all devices are in groups. These groups apply their Group Cohesion. When all groups aggregate into one, the group indicator falls, as $\sigma_{sleepDuration}$ suddenly increase. A short Neighbour Aggregation phase takes place, forming the last and biggest group. After that, the Group Cohesion maintains the community.

Fig. 9.2 shows the devices discovery rate. Each increase in the plot correspond to neighbours aggregating. Increases in community sizes become larger as the groups grow, showing the snowball effect giving its name to the protocol. The community is fully aggregated at $t=492$ s. Expressed as a multiple of final functioning cycles, it means that the community can be aggregated in 10 cycles.

Fig. 9.3 shows the parameters used and the final metrics. As all devices share the same communication schedule, their duty cycle are equals. This simulation reaches a final duty-cycle of 4.2%. That is higher than our objective of 1%, but this proves that the protocol can work to attain useful duty-cycles. As the duty-cycle is homogeneous, the battery-life estimations of a stable Snowball

All those results shows that Snowball can be used to create an emerging synchronization pattern in a BLE ad-hoc network.

system are made easier.

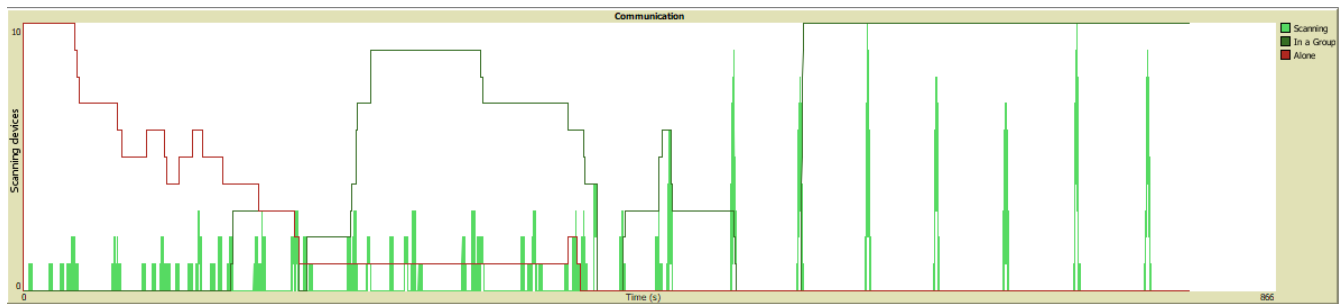


Fig. 9.1. Devices activity.

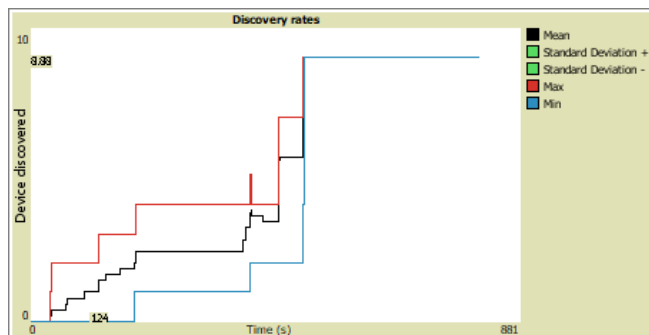


Fig. 9.2. Devices discovery rates

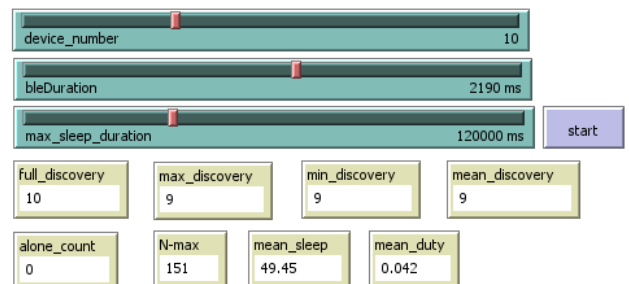


Fig. 9.3. Devices parameters and final indicators

VI. EXPERIMENTATION OF SNOWBALL PROTOCOL

The proposed Snowball protocol is implemented on a testbed of ten ESP32-based devices. All experience details are presented in this section.

A. Architecture and material

Ten FireBeetle ESP32 IoT devices (as the test community) and a custom ESP32 board (as a monitoring and scan-only board) are used for the real-life test. Moreover, a specific JAVA application was developed. The monitoring board is connected through serial connection to this application. It records each received advertisement event, and shows various debug data such as size of the community, number of received advertisements, number of different devices detected, Snowball state, and RSSI measurements (see Fig. 10). In addition, each device has a LED that flash during its activity to provide visual feedback on the firefly-like behaviour.

B. Use case scenarios

The same simulation scenario is used in this experimentation study. After start listening of the monitoring board, each device is powered on separately at random. By observation of the plotted data in the JAVA application and by visual interpretation of displayed advertisements, it is possible to assess the formation of groups and evaluate the real performance of Snowball.

C. Experimental results

a) Snowball protocol results

The Snowball experimentation provides similar results to the simulation. The results are shown in Fig. 10, which is divided in three parts. In part 1, the raw information coming from the monitoring board by a serial port is displayed. They are then organized and assigned a timestamp to be displayed in part 2. Each row and colour correspond to a device. Even if Snowball only needs the ID and Sync fields, other fields are used as monitoring and debug to study the protocol.

A plot is shown in part 3. It is the main monitoring tool. On it, we can see bursts of communication, where groups of devices emit at the same time. Although the groups are less stable and have difficulty coalescing into a unique group, there is clearly an aggregation taking place. The aggregated groups communicate with a period of approximately $\frac{\text{maxSleepDuration}}{2}$ (here ≈ 60 seconds), which is better than the 50 seconds found by the simulation.

That difficulty probably come from less-than-ideal parameters. That means that the algorithm used in Snowball must be improved in a better simulation. That improved algorithm will then transpose with better performances in a real-life setting.

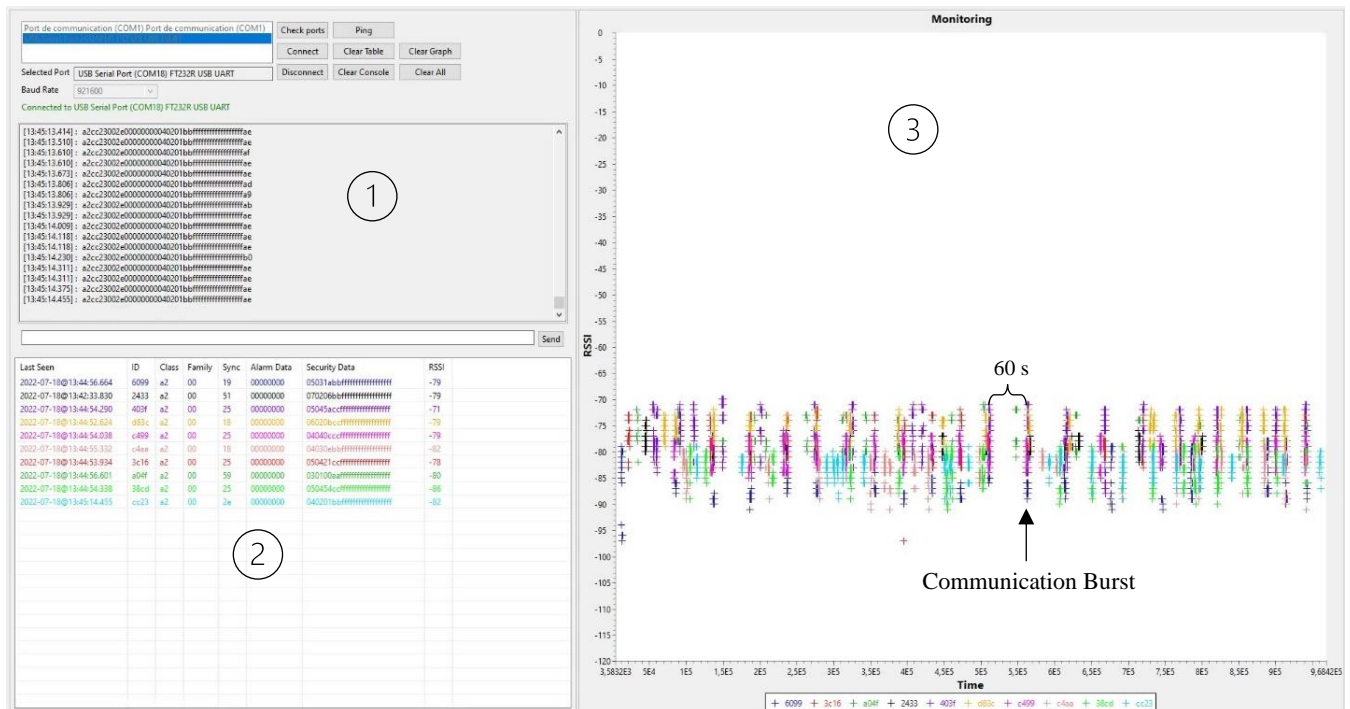


Fig. 10. Activity results of a N=10 community formation

b) BLE parameters tuning

The validity of the constraints equations is demonstrated by monitoring the energy consumption of a FireBeetle ESP32 device with the OTII tool [19]. Not complying with the determined constraints equations (Fig. 11) results in advertisements during the scan window. Such collisions can produce undefined behaviour in the BLE controller and errors at the application layer. By complying with the constraints equations, we obtain a communication operating at the edge of constraints but not producing collisions. (Fig. 12).

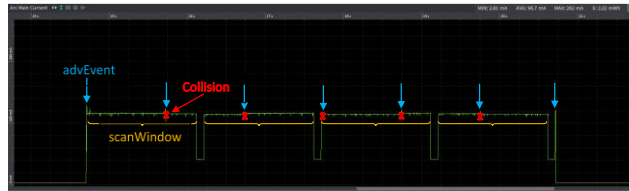


Fig. 11. Analysis of a non-compliant communication

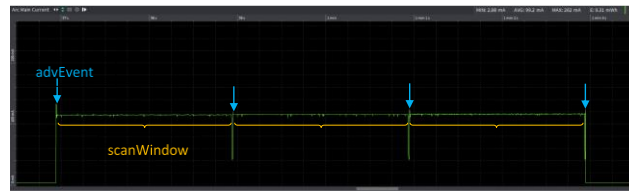


Fig.12. Analysis of a compliant communication

VI. CONCLUSION

This paper presents a new neighbour discovery protocol for ad-hoc BLE networks. It consists of a statistical aggregation after random encounters. A simulation and a real-life implementation on a 10-devices testbed have demonstrated its feasibility. Our future works will consist of testing the scalability of the Snowball protocol (up to a 100 devices) and finding how to improve its reliability. It opens the way to more computationally heavy low-layer protocols, thanks to powerful communication microprocessors.

REFERENCES

- [1] S. Trab, E. Bajic, A. Zouinkhi, A. Thomas, N. Abdelkrim, "A communicating object's approach for smart logistics and safety issues in warehouses", *International journal of Concurrent Engineering: Research and Applications*, Vol. 25, No. 1, 2017, pp. 53-67.
- [2] W. Sun, Z. Yang, X. Zhang, Y. Liu, "Energy-Efficient Neighbor Discovery in Mobile Ad Hoc and Wireless Sensor Networks: A Survey", *IEEE Communications Surveys & Tutorials*, 2014, Vol. 16, No. 3, pp. 1448-1459.
- [3] W. Sun, Z. Yang, K. Wang, Y. Liu, "Hello: A generic flexible protocol for neighbor discovery", *IEEE Conference on Computer Communications*, 27 April 2014 - 02 May 2014, Toronto, ON, Canada.
- [4] K.E. Jeon, J. She, P. Soonsawad, P.C. Ng, "BLE Beacons for Internet of Things Applications: Survey, Challenges, and Opportunities", *IEEE Internet of Things Journal*, Vol. 5, No. 2, April 2018, pp. 811-828.
- [5] B. Luo, Y. Yao, Z. Sun, "Performance Analysis Models of BLE Neighbor Discovery: A Survey", *IEEE Internet of Things Journal*, Vol. 8, No. 11, June 2021, pp. 8734-8746.
- [6] T. Liu, C. M. Sadler, P. Zhang, M. Martonosi, "Implementing software on resource-constrained mobile sensors: experiences with impala and zebrant", *2nd International Conference on Mobile systems, applications, and services*, 6-9 June 2004, Boston MA USA.
- [7] D. Li, P. Sinha, "RBTP: Low power mobile discovery protocol through recursive binary time partitioning", *IEEE Transactions on Mobile Computing*, Vol. 13, No. 2, February 2014, pp. 263-273.
- [8] V.K. Amalladinne, K.R. Narayanan, J.-F. Chamberland, D. Guo, "Asynchronous Neighbor Discovery Using Coupled Compressive Sensing", *IEEE International Conference on Acoustics, Speech and Signal Processing*, 17 April 2019, Brighton, UK.
- [9] H. Chen, W. Lou, Z. Wang, F. Xia, "On Achieving Asynchronous Energy-Efficient Neighbor Discovery for Mobile Sensor Networks", *IEEE Transactions on Emerging Topics in Computing*, Vol. 6, No. 4, October-December 2018, pp. 553-565.
- [10] N. Karowski, A.C. Viana, A. Wolisz, "Optimized asynchronous multi-channel neighbor discovery", *30th IEEE International Conference on Computer Communications*, 10-15 April 2011, Shanghai, China.
- [11] L. Wei, Y. Chen, L. Chen, L. Zhao, L. Chen, "A Neighbor Discovery Method Based on Probabilistic Neighborhood Model for IoT", *IEEE Internet of Things Journal*, Vol. 6, No. 6, December 2019, pp. 9350-9359.
- [12] W. Li, J. Zhang, F. Jiang, Z. Li, C. Xu, "Asynchronous neighbor discovery with unreliable link in wireless mobile networks", *Peer-to-Peer Networking and Applications Journal*, Vol. 12, 2019, pp. 635-646.
- [13] R. Sarfati, K. Joshi, O. Martin, J.C. Hayes, S. Iyer-Biswas, O. Peleg, "Emergent periodicity in the collective synchronous flashing of fireflies", *bioRxiv*, 11 March 2022.
- [14] K. Mekki, E. Bajic, F. Meyer, "Indoor positioning system for IoT device based on BLE technology and MQTT protocol", *5th IEEE World*

Forum on Internet of Things, 15-18 April 2019, Limerick, Ireland.

- [15] Joakim Lindh, "Bluetooth low energy Beacons", Application Report Texas Instruments, October 2016.
- [16] T. Herman, S. Pemmaraju, L. Pilard, M. Mjelde, "Temporal Partition in Sensor Networks", 9th International Symposium on Self-Stabilizing Systems, November 14-16, 2007, Paris France.
- [17] E. Sklar, "NetLogo, a Multi-agent Simulation Environment." Artificial Life 13, 2007, pp. 303-311
- [18] F. Chiacchio, M. Pennisi, G. Russo, S. Motta, F. Pappalardo, "Agent-Based Modeling of the Immune System: NetLogo, a Promising Framework", BioMed Research International, 2014.
- [19] Otii by QOITECH Available at: www.qoitech.com