December 2019

Fingerprinting Application To Emulate BLE Tags During Calibration Process

Jérôme ELLEOUET
ALE International

Anish VERMA
ALE USA Inc.

Docket Number	**FR82019003**
Title	**“Fingerprinting application to emulate BLE tags during calibration process”**
Contributors	**Jérôme ELLEOUET – Anish VERMA**
Companies	**ALE International – ALE USA Inc.**

Description of the technical solution:

➢ Introduction and background:

Among various technologies available on asset tracking solutions (Wi-Fi, RFID, Ultrasound), **B**luetooth **L**ow **E**nergy (BLE) is one of the most efficient with respect to accuracy and cost of deployment of the solution.

Most of companies delivering asset tracking solutions based on BLE technology are using today some BLE gateways in order to scan BLE tags and push back RSSI (**R**adio **S**ignal **S**trength **I**ndication) measurements to a location engine, in a cloud environment or at customer premises, able to determine the position of the asset on the indoor map.

The calibration/fingerprinting is an important step of the deployment of such solutions. It consists of building a database of sample measurements that are used by the algorithm to enhance its accuracy.

Generally, the algorithm is using a comparison of collected RSSI information of BLE tags via BLE gateways to determine the position of the BLE tag. Trilateration algorithm can lead to errors due to the variable nature of radio signals. To solve this issue, comparing the RSSI values provided by a list of BLE tags as reference samples is often used. The reference samples are collected by a phase called calibration or fingerprinting.

More and more algorithms are using system autocalibration to virtually build this set of reference samples. This consists of a phase where the gateways are scanning beacons around them (with fixed and known positions) as well as the other gateways (also acting as beacons with fixed and known positions) to build a virtual map of RSSI values used later by the algorithm to enhance its accuracy.

Even if some vendors are providing autocalibration, generally to ensure a better accuracy in some specific places, for instance achieving room accuracy in hospital rooms, a manual fingerprinting is still required. An equivalent phase could also be required for troubleshooting purpose.

The phase consists to send somebody on site, generally using a mobile application or a PC, pointing himself at the exact position on the application map (position markers + timestamp of the markers) while gateways are scanning a given list of BLE tags that the person is wearing on him or having in hands.

Four major issues are coming from this process:

1. Timestamp:

A perfect synchronization is requested between the mobile phone where position markers are set (with timestamps) and the measures that occur at gateway level: the mobile phone and the gateways need to be perfectly well synchronized. This can be achieved using NTP (**N**etwork **T**ime **P**rotocol) synchronization but could be complex to manage. In all cases, errors could happen.

2. Doing calibration with dozen types of tags (for instance) is not easy:

Depending on use cases, it is quite common that different form factors of tags are requested. These tags may not come from the same provider. Among time, new types of tags can be needed. It is quite uneasy to wear or keep in hand dozen tags while also using a mobile phone and point positions on the map. All types of tags used in the solution is required for the calibration as each tag has different radio-frequency properties, which vary from each other and can affect the accuracy.

3. Depending on use cases, tag configurations can be different:

Some use cases require tags to emit at low power for usage in proximity mode. Some use cases require high power for wayfinding and positioning for instance. Sometimes, an intermediate transmit power is requested. Ideally during the calibration phase, for a type of tag configured with three different power levels, some RSSI references must be collected for the three tags. Instead, almost all calibration processes are using one type of tag configured at a given transmit power level and then some extrapolation is applied – based on mathematical models – to compute the RSSI value if the tag was set with more or less power.

4. Often, location algorithms do not take into account the difference between tags:

Some tags may come from various providers and have different radio-frequency behaviors. To address specific use cases (like hand hygiene in the healthcare market) it is likely to be required that the tags need to be configured to have specific emission frequency or transmit power. For hand hygiene compliancy, high emission frequency and low transmit power are required to ensure the tags are quickly and precisely detected by a dedicated Bluetooth gateway placed inside the soap dispenser.

Some companies are providing a dedicated mobile application to do fingerprinting of an asset tracking deployment.

The recommendation is to define a NTP server on each BLE gateway of the system. This can be a costly task especially if the system is using third party gateways with specific firmware; each gateway must be configured. A deployment typically in a hospital can easily reached 500 gateways and this single task can represent a significant effort. Providers will full control of the solution can manage using their management platform to push this configuration information to the gateways.

Even with a NTP server defined in the network, if the end user doing the fingerprinting does not connect the phone to the correct Wi-Fi network some desynchronization will occur causing the fingerprinting to be erroneous. The better the synchronization is, the better the fingerprinting will be.

On the other hand, the mobile application generally requires the user of the application to walk in the venue with few tags in the hands. These tags make difficult to properly manage positioning on the map application and handling the tags on the other hand.

Some solutions, from WLAN providers are claiming the usage of a machine learning to auto calibrate the system and applies this only to wayfinding services (application running on mobile phone positioning the end-user and giving him the bluedot/wayfinding to a given point of interest). Such solutions require a SDK (**S**oftware **D**evelopment **K**it) to run on the mobile phone: this SDK receives RSSI information from beacons around and pushes information to the cloud or to a local server where a location engine is calculating positions that are then pushed back to the SDK and so to the mobile application. Along with RSSI information, the SDK provides information about the phone that allows the algorithm in the cloud to adapt the responses (position) based on the type of phone.

- **New solution:**

In this paper, we propose a solution consisting in emulating a set of BLE tags during the calibration/fingerprinting phase.

The mobile application is simultaneously sending signals that correspond to ten different types of tags with different behaviors (transmit power mainly) removing the need to use ten physical tags in hands during calibration. This will allow to calibrate the system with ten different types of tags (from various providers) but also to consider a same tag with multiple different configurations in terms of transmit power.

Currently some beacon emulators exist but they are only able to emulate iBeacon or Eddystone frames that are the most common standard frames sent for BLE signals and that are sending constantly the same BLE frame.

In our case we need a specific frame because the second part of the solution consists in delivering a frame that contains a hash of the timestamp that correspond to the time the user doing the calibration/fingerprinting indicates its position on the application map.

Each BLE frame emitted from the phone contains a hash of the timestamp. If five different BLE gateways are receiving this BLE frame (with five different RSSI), it will be easy and 100% accurate for a calibration algorithm to detect that these five frames provided by five different gateways are corresponding to the same tag emission.

Moreover, with the phone being able to emulate five different types of tag we can ensure the location engine in charge of calculating positions will have the ability to take into account the different behaviors of the tags in the algorithm. The position calculated can be adapted to the type of tag.

During the calibration phase, a user using a mobile application will point his exact position (by clicking on a button) on the application map. This will deliver latitude, longitude and altitude coordinates to the calibration engine. Along with the position, the type of tag (or tag ID) and transmit power level are also sent. The information is sent to the calibration engine using a TCP protocol like HTTP (REST API for instance) or MQTT (**M**essage **Q**ueuing **T**elemetry **T**ransport). Other possibilities exist.

As soon as the button is pressed, N numbers of BLE frames are sent out from the mobile (using the mobile OS SDK).

N depends on the number of tags that needs to be emulated and on the number of different configuration for each tag.

☞ **Note:** Further study needs to be done to check the amount of time needed to send these frames – depending on their number – and check if user needs to stay at fixed position for a given amount of time (<1s). Current expectation is these frames are very small (few bytes) and should take less than few milliseconds.

The length of BLE frames are 31 bytes. Two main protocols are in use today: iBeacon and Eddystone. But it is perfectly possible to define a customized frame format as soon as these proprietary frames stay in the limit of 31 bytes.

The mobile application will send N frames, each frame containing following information:

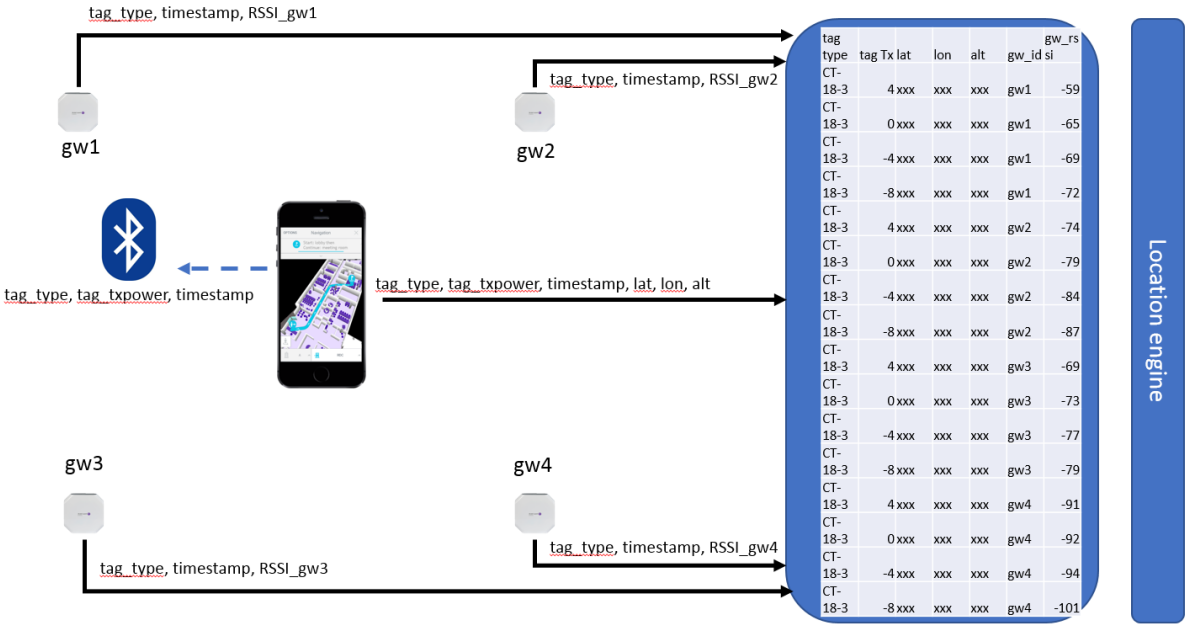
- Tag type
- Tag transmit power level
- Timestamp (or hash or generated ID)

These frames will be caught by the BLE gateway. The RSSI level will be computed at the gateway level and the gateway will then push back information to the calibration engine (e.g. using MQTT or HTTP protocols).

At calibration engine level, the timestamp (or unique ID generated for the BLE frame emission) is used to correlate information coming from the mobile application and from the gateways (four gateways in the figure below).

The calibration engine can then build tables that will allow a location engine to compute positions from a given number of tag types with different transmit power configuration.

☞ **Note:** The calibration engine and the location engine can be hosted on premises or in the cloud. This is not important regarding this proposed solution. The solution also does not describe the way the calibration or location engine are designed.



tag type	tag Tx	lat	lon	alt	gw_id	gw_rssi
CT-18-3	4	xxx	xxx	xxx	gw1	-59
CT-18-3	0	xxx	xxx	xxx	gw1	-65
CT-18-3	-4	xxx	xxx	xxx	gw1	-69
CT-18-3	-8	xxx	xxx	xxx	gw1	-72
CT-18-3	4	xxx	xxx	xxx	gw2	-74
CT-18-3	0	xxx	xxx	xxx	gw2	-79
CT-18-3	-4	xxx	xxx	xxx	gw2	-84
CT-18-3	-8	xxx	xxx	xxx	gw2	-87
CT-18-3	4	xxx	xxx	xxx	gw3	-69
CT-18-3	0	xxx	xxx	xxx	gw3	-73
CT-18-3	-4	xxx	xxx	xxx	gw3	-77
CT-18-3	-8	xxx	xxx	xxx	gw3	-79
CT-18-3	4	xxx	xxx	xxx	gw4	-91
CT-18-3	0	xxx	xxx	xxx	gw4	-92
CT-18-3	-4	xxx	xxx	xxx	gw4	-94
CT-18-3	-8	xxx	xxx	xxx	gw4	-101