BLE Beacon Based Patient Tracking in Smart Care Facilities

Maeve Kennedy, Graham W. Taylor, Petros Spachos

School of Engineering, University of Guelph, Guelph, ON, N1G 2W1, Canada
E-mail: {maevek, gwtaylor, petros}@uoguelph.ca

Abstract—**Patient tracking is an important component toward creating smart cities. In this demo we use Bluetooth Low Energy beacons and single board computers to track patients in the emerging field of smart care facilities. Our model utilizes a fixed scanner moving transmitter method for wireless tracking of patients through the facility. The data collected by all scanners is stored within a central database that is designed to be efficiently queried. We show how inexpensive components in conjunction with free open source software can be used to implement a patient tracking system. We focus on the pipeline between acquisition and display of the location data. Additionally, we also discuss the manipulation of the data required for usability, and optional filtering operations that improve accuracy.**

I. Introduction

A central challenge in health care centres is patient tracking. Many applications of patient tracking exist within care facilities; one prominent application is ensuring the safety of patients [1], [2], [3]. Tracking wandering patients with Alzheimer's disease or dementia, or preventing infant abduction are commonplace applications in health care facilities. Currently available solutions are costly and intrusive, often requiring virtual fencing such as placing RFID scanners in entrances and exits of the tracked area [4]. RFID scanners also do not provide fine grained location information. Other available systems are proprietary or use energy costly solutions such as WiFi with mobile battery powered units requiring frequent refreshing of batteries.

In this demo, we propose a Bluetooth Low Energy (BLE) solution using consumer available beacons, edge modules, and free open source software. BLE beacons are non intrusive, low cost, and freely available to consumers. These beacons broadcast an identifier to all listening edge units that can be used to give coarse locations within a facility. Unlike RFID solutions, our system does not require alterations to a facility to allow for the installation of virtual fences. Our system also adds value by providing support staff with non-critical location information for use when administering care in a large-scale facility. Freely available consumer hardware is used for the edge modules in our system. The edge modules use the Raspberry Pi 3 which are a low cost, low power, WiFi and Bluetooth enabled, single board computer.

II. System Overview

In this section, we describe the components of our solution for monitoring patients in smart care facilities. We focus on the aggregation pipeline starting with the low-energy beacons. The client server architecture is implemented for communication between the edge modules and beacon server in our system. This system has five main components that are described below.

Fig. 1. Gimbal Series 10 Beacon next to an American quarter dollar for scale.

A. BLE Beacons

BLE is a wireless technology that markets toward the Internet of Things (IoT). BLE has two channels: the advertising channel and data channel. iBeacon is a simple protocol by Apple for BLE beacons. iBeacon identifies beacons with a 3 tuple of UUID, Major, Minor which are 20 bytes of identifying information [6]. The iBeacon protocol only makes use of only the advertising channel of BLE. It is important to note, whereas we utilize the iBeacon protocol in our solution due to simplicity of implementation, our system is not restricted to beacons using the iBeacon protocol. For this demo we use the Gimbal Series 10 beacon [7] in iBeacon configuration with 10Hz advertisement shown in Fig. 1.

We estimate distance from the edge to the beacon using the RSSI (Received Signal Strength Indicator). We utilize the path-loss model suggested in [8] given as:

$$RSSI = -(10 \times n) \log_{10}(d) - A \tag{1}$$

where $n = 2$ is our signal propagation constant, d is our distance in metres, and A is our transmit power.

The beacons should be affixed to the patients in a non invasive place: one such area is around the patient's wrist in combination with a wrist band. The beacons are roughly the size of a watch face and typically lighter. Being affixed to the patients wrist is likely the best compromise between comfort and signal efficiency as it is not obstructed by most clothing.

B. Edge Modules

As edge modules our Raspberry Pi 3s run the Raspbian Stretch Lite along with other free and open source software.

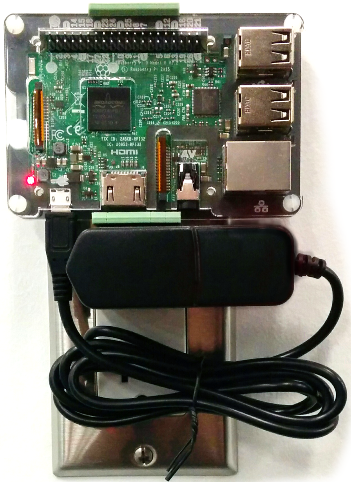


Fig. 2. Raspberry Pi 3 with custom bracket and DC power supply mounted to AC power outlet.

The complete module for the edges includes a bracket, AC to DC power supply, and the Raspberry Pi 3 unit mounted to a wall power outlet as shown in Fig. 2. The edge modules actively scan for BLE advertisements, checking them against a permissive list of registered beacons obtained from the beacon server. Using a permissive list allows the Raspberry Pis to ignore other BLE beacons that are not part of our system, saving on bandwidth and processing time. The edge devices periodically check the central database to see if there has been updates to the registered list of beacons. The edges are hot swappable, simple to set up, and remotely controllable.

When the edges receive a beacon advertisement they append a time stamp and aggregate the advertisement into a packet. The edge modules bundle up to a protocol specified but arbitrary limit of beacon sightings, forwarding them to the beacon server over a Transmission Control Protocol (TCP) connection secured with Transport Layer Security (TLS). TLS is a common cryptographically secure method of communicating over untrusted networks. In order to prevent data from becoming stuck, the edge device will wait a predetermined timeout period and then send whatever data it has collected to the central database. This prevents certain sightings from waiting too long in the queue. The timeout can be adjusted to balance efficiency and response time. WiFi attached to the backbone Internet Protocol (IP) network connects the edge modules to the beacon server.

C. Beacon Server

The beacon server indexes and collects sightings into a responsive database capable of storing and querying millions of rows. Our implementation of the beacon server uses a Go based service that is backed with a Postgres database. Go is a programming language that is designed to be memory safe, powerful, and has parallel programming as a core feature. The beacon server service ensures confidentiality, integrity, and availability to the edge nodes, processing multiple incoming connections in parallel. The Postgres database uses the Structured Query Language (SQL) to ensure that the database adherers to the ACID properties (Atomicity, Consistency, Isolation, Durability). Postgres is a freely available standards-driven database.

The beacon server has a control feature where upon receiving a packet from an edge, the server will send back control commands to the edge allowing the edge to be remotely updated, rebooted, or any other necessary operations. The beacon server can be installed on premise or in the cloud depending on the implementer's security and latency needs. It is based on free software solutions and therefore has no costly or restrictive licensing.

D. Metrics Server

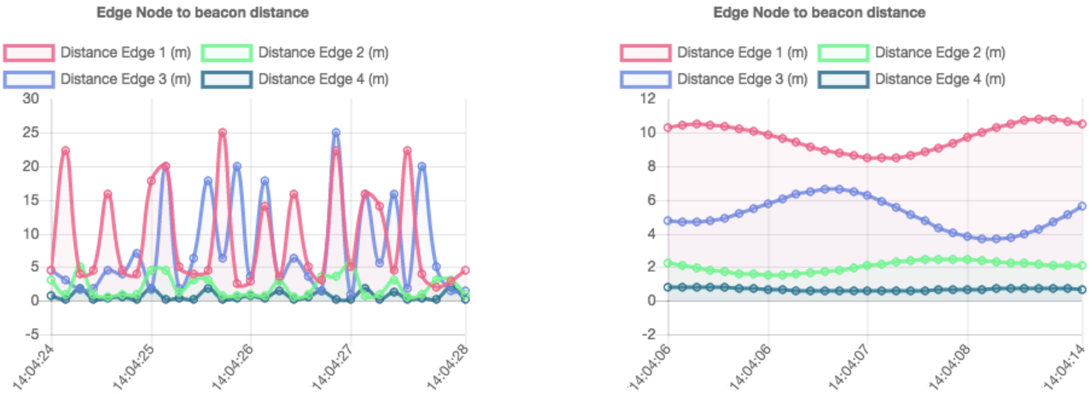
The server provides a JavaScript Object Notation (JSON) service for requesting data from the database. JSON is a popular data interchange format that has parsing libraries available for all popular programming languages. By using JSON we separate the concern of collecting data, serving data, and displaying data. One such use of this service is requesting all sightings for a given beacon since a certain time. This allows us to build location-based services decoupled from the main server software. The metrics server is just a simple example of a service that can be implemented based on the database. Any popular data interchange format could be implemented on top of the database.

E. Client Interface

The client interface is a proof of concept to locate the beacon in a physical space. The current prototype allows one to select and lay out the edge locations in a virtual canvas and then select which beacon is to be displayed. The virtual canvas is shown in Fig. 4. The client is responsible for data filtering and distance calculations. In our demo the beacon is assigned to the closest beacon given the filtered distances. Filtering the distances is done by applying a low-pass Butterworth filter with an order of 4 and cutoff frequency of $\frac{\pi}{10}$ radians/sample. The nearest edge measurement should approximate the location of the wearer in a smart facility. The client could implement trilateration for fine grained location of the beacon in the facility. Signal filtering is very important in this system as the propagation of the BLE advertisements is often not in line of sight and very noisy. We show the distance calculations with a fixed target and without filtering in Fig. 3 as taken from the client interface. The client interface is written in JavaScript and HTML. Therefore, it should be usable on any devices that have compatible web browsers such as smart phones or tablets.

III. Prototype Implementation

We implemented our prototype in two laboratories in our home department. Raspberry Pi 3s configured as edges were placed at outlet height (approximately 30cm from the ground). The edges were placed with overlapping zones, such that the test area was in the visible zone of all edge units, with a minimum distance of 2 meters. We then carried the beacon



(a) (b)

Fig. 3. (a) Raw distance from edge to beacon in metres. (b) Filtered distance.

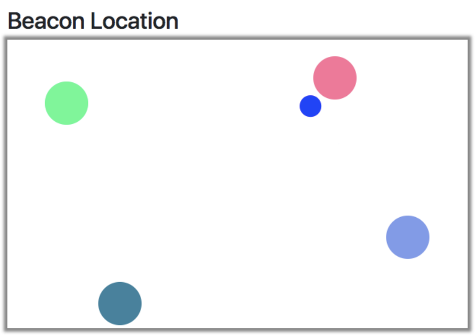


Fig. 4. Excerpt from web based client interface with edges (big circles) and beacon (small circle).

between the relative zones of the edges and observed the beacon's assigned edge. This imitates the movements that a patient would make in a care facility. The assignment was determined by the closest edge with a filtered distance. The distance metric for each edge was filtered with a simple low pass filter described in II-E. In practice, one would want to have smaller overlaps between edges to be more cost efficient and cover more area; this experiment was performed simply to test the sensitivity of the edges.

We then observed the number of invalid assignments of the edge. With the filtering disabled the assignment was very noisy. With the filtering enabled the assignment was much more stable, but it prevents quick transitions between zones. The difference between the filtered and unfiltered data is shown in Fig. 3. Depending on the overlap between zones the implementer would need to tune this parameter. Future work will employ more complex filtering schemes.

IV. Demo Outline

Our demo will cover the assignment of the beacons to the proximity of the edges. The setup will utilize four edge modules with the minimum edge distance being greater than 2 metres in a rectangular shape. The demonstrator will move between the relative proximity of all the edges with a beacon. The edges will send the data to a remote server with the beacon server and metric server running. The demonstrator will enable and disable filtering to compare. The client interface will be displayed, and the beacon will move from relative proximity of one edge to the next. Enabling and disabling the filtering will show how much of an improvement is made with the filtering when assigning the beacon to a given location.

V. Conclusion

We presented a system for aggregating and querying beacon location information for monitoring patient locations. The system we have designed will be used in the future to develop new filtering methods to improve the resolution of the patients' location. The aim of this demo was to create a system utilizing freely available hardware and software at a low cost per patient. The system will be expanded in the future to improve usability, make the system more accurate, and feature rich. A free open source implementation of this project is available at [5].

References

[1] A. Puiatti, A. Rivero, A. Frster, A. Bernaschina, G. C. Dozio, N. Rizzo, N. D. Bitonti, and A. Cavicchioli, "Proxy-care: A novel patient care tracking system with wireless sensor networks," in *2014 IEEE Healthcare Innovation Conference (HIC)*, Oct. 2014, pp. 38–41.

[2] L. Wan, G. Han, L. Shu, S. Chan, and T. Zhu, "The application of doa estimation approach in patient tracking systems with high patient density," *IEEE Transactions on Industrial Informatics*, vol. 12, no. 6, pp. 2353–2364, Dec. 2016.

[3] H. Q. Omar, A. Khoshnaw, and W. Monnet, "Smart patient management, monitoring and tracking system using radio-frequency identification (rfid) technology," in *2016 IEEE EMBS Conference on Biomedical Engineering and Sciences (IECBES)*, Dec. 2016, pp. 40–45.

[4] Advance Technology, Inc, "Preventing infant and pediatric abductions, patient elopements, and wandering," accessed 2017-11-26. [Online]. Available: http://www.advancetechnology.com/patient-safety-solutions-infant-and-patient-wandering

[5] B. Kennedy, "beaconpi," accessed 2017-11-29. [Online]. Available: https://github.com/co60ca/beaconpi

[6] Apple Inc, "Getting started with ibeacon," accessed 2017-11-26. [Online]. Available: https://developer.apple.com/ibeacon/Getting-Started-with-iBeacon.pdf

[7] Gimbal Inc, "Gimbal proximity beacon series 10," accessed 2017-11-26. [Online]. Available: https://store.gimbal.com/collections/beacons/products/s10

[8] Q. Dong and W. Dargie, "Evaluation of the reliability of rssi for indoor localization," in *2012 International Conference on Wireless Communications in Underground and Confined Areas*, Aug. 2012, pp. 1–6.