Dispatchable Indoor Location System for Mobile Phones based on a BlueTooth Low Energy Array

Carol Davids*, Cary Davids† Bharat Ramaswamy Nandakumar‡ Neilabh Okhandiar§
Francisco Rois¶ Chakib Ljazouli‖ Adrian Calle Murillo**
Illinois Institute of Technology
3201 S. State St. Chicago, IL 60616
Email: { *davids, †cdavids2 }@iit.edu
{ ‡bnandaku, §nokhandi, ¶froissiso, ‖cljazouli, **acallemurillo }@hawk.iit.edu

Abstract—**This paper describes a system that provides the indoor location of an emergency 9-1-1 caller who uses a smartphone while inside a large, multi-story building. The system consists of an Array of BlueTooth Low Energy (BLE) beacons and a Database that together provide the indoor location, and a Smartphone Application that obtains the location information and makes an IP-based call, including the indoor location information, to a Next Generation 9-1-1 Public Safety Answering Point (PSAP.) The Array and the Database together constitute a platform that can be used by developers to create other applications that use its data. Both the platform and the smartphone application will be described. Information about the architecture of the service platform and the operations and maintenance of the array are included. Tests to determine the accuracy of the caller's location are described and their results presented. Potential additional uses for the platform are proposed.**

I. Overview and Background

When a person calls an emergency number, such as 9-1-1 in the US, the location of the caller must be provided to the network. It is used for two purposes: First, to route the call to the appropriate answering point; Second, to display the caller's location to the call-taker who will dispatch first-responders. When the call is made from a mobile device inside a building, the call-taker and the first responders need more information than the latitude and longitude and/or the street address to find the caller. They need accurate indoor location — a floor and room number for example. This paper describes a proof of concept system that provides the floor and room number as well as the street address of a caller. A method for providing the x,y coordinates of the caller on the correct floor has also been created and will be briefly described.

The proof of concept system was developed in response to the 'Roadmap for Improving E911 Location Accuracy' [1] developed by an alliance of the Association of Public-Safety Communications Officials-International (APCO) [2], the National Emergency Numbers Association (NENA) [3], and four national wireless carriers, AT&T, Sprint, T-Mobile and Verizon. The system described in this paper is integrated with a replica of an Emergency Services IP Backbone Network (ESInet)] on a test-bed in the IIT Real-Time Communications Lab [4]. The ESInet is specified in the NENA i3 Standard [5]. The roadmap called for the use of WiFi Access Points, BlueTooth Beacons or both to be the source of the indoor location. A proof of concept system that used the WiFi Access Points on the University campus was developed and is described in [6]. The current work makes use of BlueTooth Low Energy (BLE) iBeacons, leveraging the experience gained in that earlier work. The iBeacon is a communications protocol developed by Apple and based on the BlueTooth Low Energy portion of the BlueTooth Specification [7]. Development of a new configuration for the array, that uses a hierarchical rather than a flat architecture is in progress. The iBeacons in the new array are divided into groups, each of which is managed by a gateway device. The new architecture is described briefly in Section IV.

The system consists of the following four components: (1) A smartphone application that detects the emergency call, obtains the location of the caller, and forwards the call, including location information, to the answering point; (2) An array of BlueTooth Low Energy (BLE) devices that are deployed on the inside walls of a building at selected locations; (3) A database that contains the location of the BLE devices; (4) A Location Server capable of querying the database to discover the location of particular BLE devices, and of using the location information thus obtained to calculate the location of the caller, format this location, and send the formatted information to the smartphone. Figure 1 illustrates these components and the flow of information between them. In the Proof of Concept System, the location server and database are local, whereas in the Roadmap [1] it is centrally located and available to callers in all compliant buildings.

When the user places an emergency call using the smartphone application, the application first scans for beacons then sends the identifications of all the beacons that it sees, together with the strength of the signals it receives from each - their Received Signal Strength Indicators (RSSI), to the database server as an HTTP Request. The server interacts with the database to learn the location of each BLE device in the set and applies an algorithm to the locations and RSSIs to determine either the location of the closest beacon or the actual x,y coordinates of the caller. The two options are explained in more detail in Section VIII.

Once the address and the indoor location of the caller are identified by the location server module and returned to the smartphone application, the phone initiates a call to a test-bed

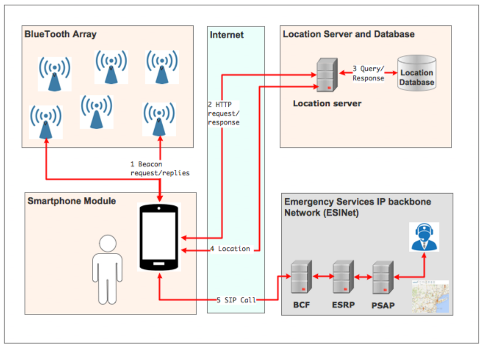


Fig. 1. Indoor Location Service Modules

in the IIT Real-Time Communications Lab (RTC Lab) that replicates the functions of an ESINet. The call is IP-based and uses the Session Initiation Protocol (SIP) [8] for its signaling component and the Real-time Transport Protocol (RTP) for its audio and video component. When the PSAP operator answers the call, the civic address as well as the room, floor and room number of the caller are displayed on the PSAP operator's computer screen.

The mobile application is built on the Android OS and includes the functions of a SIP User Agent, which are provided by the open source SIPdroid distribution. The iBeacons in the BlueTooth array operate in "advertise" mode only, broadcasting their Major and Minor identifiers every half second. The database is built using MySQL and the location server is a Linux system that includes a node.js module.

The rest of this paper is organized as follows: Section II contains the system requirements; Section III explains the flow of information from end to end; Section IV describes the BLE array; Section V contains a description of the operations and maintenance systems that were developed to support the service; Section VI describes the Android Application; Section VII describes the database and its associated server; Section VIII contains a description of four different algorithms that have been used to calculate the caller's location, the experimental method used to test the accuracy of each algorithm, and the test results. Section IX describes related work and Section X contains some conclusions and next steps. Section XI contains the references.

II. Requirements

The system was designed to meet a set of high level requirements derived from the Roadmap [1] and from the NENA i3 standards [5]. To adhere to the roadmap requirements the system must (1) provide the caller's building address and floor number; (2) provide a way to identify the location of the caller on that floor, whether that identification be a room number, a room descriptor, or an x,y coordinate given in feet or meters from an origin on that floor; (3) provide the location of the BLE device or sensor that is closest to the caller; (4) use WiFi Access Points, BlueTooth beacons, or a combination of these to locate the caller. (5) Additionally, the time elapsed between when the caller initiates the call and when the call is received at the PSAP should not exceed 10 seconds. (6) The location accuracy should be within 50 meters. The system was also designed to follow the routing methods and the location formats specified in the NENA i3 standard for the Emergency Services IP-backbone Network (ESINet) and the PSAP.

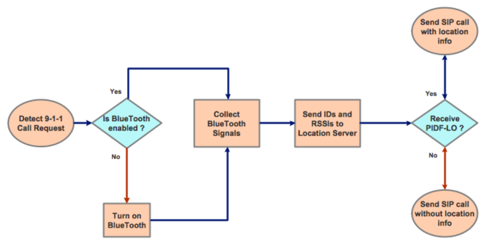

Detect 9-1-1 Call Request
Is BlueTooth enabled ?
Yes
No
Turn on BlueTooth
Collect BlueTooth Signals
Send IDs and RSSIs to Location Server
Receive PIDF-LO ?
Yes
No
Send SIP call with location info
Send SIP call without location info

Fig. 2. Smartphone Flow Diagram

```
[{"major":100,"minor":6,"rssi":-28},
 {"major":100,"minor":6,"rssi":-29},
 {"major":100,"minor":6,"rssi":-29},
 ... ]
```

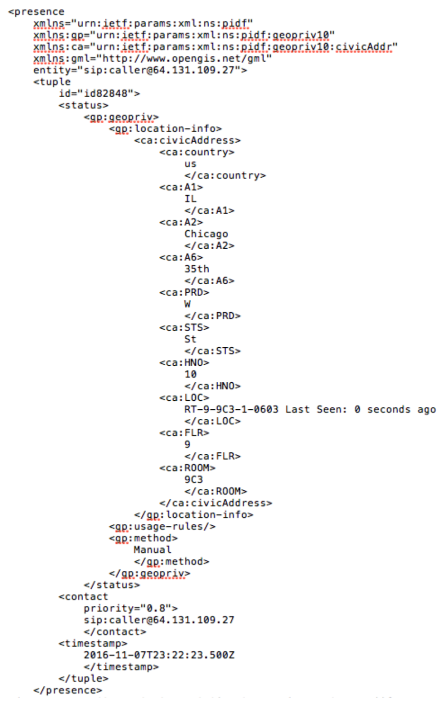
Fig. 3. Data in JSON Array format sent by the smartphone application to the database

III. Indoor Location Service Information Flow

The steps taken by the smartphone application and described in this section are illustrated in Figure 2. (1) The smartphone application detects a request for an emergency call and checks to ensure that BlueTooth is turned on, turning it on if necessary. (2) Then it initiates a 10-second scan for BlueTooth signals, recording the (major, minor) pair and RSSI associated with each signal it receives. There may be 5 -7 devices within range and each will send about 20 signals during this time frame. (3) This set of data is sent by the smartphone application to the location server. The location server averages the power in the signals sent by each unique sender and thus creates a set of 5-7 rows of data, each identified by a unique (major,minor) pair and an average RSSI. (See Figure 3) (4) The location server searches the database for the location of the iBeacon associated with each (major,minor) pair (See Figure 4). The result is a set of 5-7 iBeacon locations and their associated average RSSIs. (5) The location server uses these data to calculate the location of the caller, using algorithms that are described in Section IV. (5) The location server now formats the location information using a standard that is specified by the IETF and NENA and sends the formatted information to the phone application (See Figure 5). (6) The phone application inserts this formatted location information into the MIME body of a SIP INVITE and sends the INVITE to the ESInet.

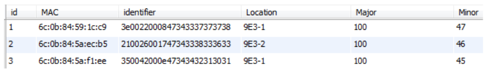
IV. The BlueTooth Array

The devices containing the BlueTooth LE beacons are deployed in a single three-story building on the IIT Mies Campus. One hundred devices were manufactured and sixty



```
<presence
    xmlns="urn:ietf:params:xml:ns:pidf"
    xmlns:gp="urn:ietf:params:xml:ns:pidf:geopriv10"
    xmlns:ca="urn:ietf:params:xml:ns:pidf:geopriv10:civicAddr"
    xmlns:gml="http://www.opengis.net/gml"
    entity="sip:caller@64.131.109.27">
    <tuple
        id="id82848">
        <status>
            <gp:geopriv>
                <gp:location-info>
                    <ca:civicAddress>
                        <ca:country>
                            us
                            </ca:country>
                        <ca:A1>
                            IL
                            </ca:A1>
                        <ca:A2>
                            Chicago
                            </ca:A2>
                        <ca:A6>
                            35th
                            </ca:A6>
                        <ca:PRD>
                            W
                            </ca:PRD>
                        <ca:STS>
                            St
                            </ca:STS>
                        <ca:HNO>
                            10
                            </ca:HNO>
                        <ca:LOC>
                            RT-9-9C3-1-0603 Last Seen: 0 seconds ago
                            </ca:LOC>
                        <ca:FLR>
                            9
                            </ca:FLR>
                        <ca:ROOM>
                            9C3
                            </ca:ROOM>
                        </ca:civicAddress>
                    </gp:location-info>
                <gp:usage-rules/>
                <gp:method>
                    Manual
                    </gp:method>
                </gp:geopriv>
            </status>
        <contact
            priority="0.8">
            sip:caller@64.131.109.27
            </contact>
        <timestamp>
            2016-11-07T23:22:23.500Z
            </timestamp>
        </tuple>
    </presence>
```

Fig. 4. Formatted location information returned to the smartphone application by the Location Server



id	MAC	identifier	Location	Major	Minor
1	6c:0b:84:59:1c:c9	3e0022000847343337373738	9E3-1	100	47
2	6c:0b:84:5a:ec:b5	210026001747343338333633	9E3-2	100	46
3	6c:0b:84:5a:f1:ee	350042000e47343432313031	9E3-1	100	45

Fig. 5. Table in the Location Database associating the Device's Major/Minor identifier with its indoor location

of them are deployed, twenty on each of the building's three floors. The devices are evenly spaced along corridors and in classrooms. Devices are also placed in each bathroom and in the stairwells. The remaining forty devices are kept on charge in the lab and can be substituted for the deployed devices when necessary.

A. Device Design

The devices containing the beacons include a battery, a WiFi chip and a BLE chip as shown in the photograph in Figure 6. The BLE chip is configured to be an iBeacon and as such to operate in "Advertise Mode", broadcasting its Major and Minor identifiers every half second but not permitting other devices to connect to it. The device thus acts as a "lighthouse" that the caller's smartphone will use to gather data that the Indoor Location Server will use to identify the caller's location. The WiFi chip is used for administrative and monitoring purposes only. It is specifically not used by the caller's phone. There is no need for the phone to connect to the device. The lack of connectability adds to the security of the Service. The WiFi chip does not directly support the caller, rather it helps the service's administrators maintain system integrity. Our device code "wakes up" the WiFi chip once every 24 hours and causes it to send its identification, code version and battery level to a secure cloud provided

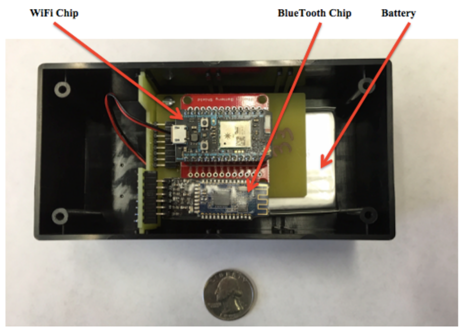


Fig. 6. BlueTooth Device

by the chip manufacturer. The identification, code version and battery level data flows through to the indoor location system's Location Database that we have installed on a server in an AWS cloud. The data thus collected enables system administrators to monitor the battery level and code version of the array of deployed devices. This process also allows the administrator to flag devices that fail to report.

The device code is protected from hackers. It must be compiled on the manufacturer's servers and the only people who have access to those servers are those that are authenticated during the purchase process. Once compiled, the binaries can be installed on the devices in two different ways. Either through a wireless connection between the manufacturer's servers and the devices, secured by the presence of keys written into the devices at the time of manufacture, or on a wire through a serial connection between the developer's computer and the device. This latter method might seem to provide an opportunity for hackers to install malicious code on the devices. However, the binaries compiled by the manufacturer's server include an application hash that will not be verifiable if the code is changed.

1) Device Deployment Architectures — Homogeneous vs. Hierarchical: In the first deployment configuration, all the devices play the same role. They act like lighthouses, beaming their IDs every half second. Each device sends its status data to the database once every 24 hours. An alternative configuration is being developed now. It involves two types of device, one a gateway containing both a WiFi chip and a BLE chip, and the other an iBeacon without WiFi capability. In this hierarchical configuration each gateway is responsible for managing 5 or 6 iBeacons. The iBeacons were purchased off the shelf and include sensors that report the temperature and humidity in their location. The code for the gateways is under development by the research team. The gateway connects to each iBeacon in its group and retrieves temperature and humidity data from each. These data are passed to the Location server at regular intervals. A prototype of this system is currently being tested in one university building. The data collected will be valuable to building engineers who can use it to improve their understanding of the building's environment. Eventually the data could be used to control the heating, cooling and other building systems.

V. Operations and Maintanence

The indoor location service is designed to save lives. Its ability to do so is dependent on the accuracy of the information in the database and on the availability of all the beacons at all times. Systems to ensure the accuracy and availability of data were developed. The first of these was a deployment application that allows the person who is putting the beacons onto the walls to do so in a way that minimizes the possibility of human error. An android application was developed for this purpose and is described in Section VI. The application provides drop-down menus with the identification of the beacons available to be deployed and the positions at which they are to be placed. More recently this application has been replaced by a web-based system that does server-side error checking and requires authentication of its user.

If a beacon's battery level is too low, it will stop participating in the array and thus skew the data used to determine the caller's location. The system includes an administrative portion of the database that enables a system administrator to monitor the battery levels of all reporting devices and to remove and replace devices that have low battery or that have failed to report. Most of the devices already deployed have been in continuous operation for eight months.

Periodic changes will need to be made to the device code itself. The administrative portion of the database also supports automatic code changes to all or a portion of the devices. A field in the database contains the value of the code version that the administrator intends the device to run. In case the version reported by the device does not match the version indicated in the database, the system updates the device code accordingly.

VI. The Smartphone Application

Three mobile applications were developed, each for different scenarios and actors in those scenarios: The caller's application, the deployer's application, and the researcher's application. All are built on the Android operating System (OS). The user's application supports the caller seeking emergency assistance. The deployer's application is used by operations and support personnel who are placing the BLE beacons initially or who are replacing defective or low-battery beacons. The experimenter's application is used by researchers who are developing and testing algorithms that will produce accurate indoor location information. The user's application includes the functions of a SIP User Agent, a BlueTooth agent and an HTTP client. The experimenter's application uses the BlueTooth and the HTTP component. The deployer's only needs the HTTP functionality.

A. Caller App

The Main Screen and the Calling Screen of the User's application are shown in Figure 7. Three sections on the Main Screen allow the user to place a call, to see the status of the call and to add more information via a text interface if desired. When the user selects the“911 Call” box, the smartphone's operating system scans for BLE beacons. The set of beacon-identifiers that results, together with their respective RSSIs, is sent to the Location server in an HTTP message and the location that the server calculates is returned to the smartphone application in an HTTP response. The smartphone application sends the civic address together with the indoor location to the ESInet in the body of the SIP message that is used to place the call. To create the SIP signaling and the RTC Media flow the application uses SIPDroid, an open software distribution that includes basic SIP and Voice over IP (VoIP) functions. The civic address is used to route the call to the correct PSAP in accordance with the i3 standards. Both civic address and indoor location are displayed on the computer screen of the PSAP operator.

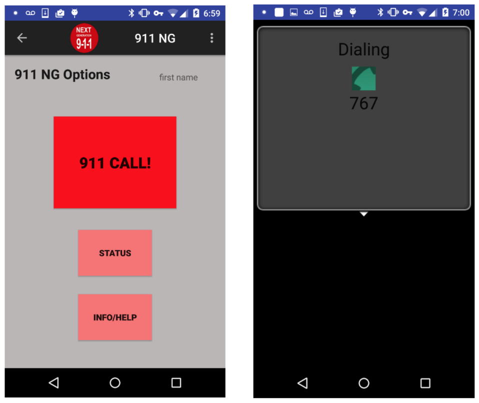


Fig. 7. User App main screen and call screen

B. Test App

To test the location algorithms, a set of test positions was selected. Testers stood at each test position and triggered a scan for BlueTooth beacons, in the same way that the call application does. The resulting set of data is sent to an“experiment database”. The algorithm developer can use the datasets in this database as input to different algorithms. Since the exact location of the test position is recorded, it can be compared to the results of the algorithm's calculation and the accuracy of the algorithm can be calculated. The measure of accuracy is taken to be the distance between the actual test position and the position calculated by the algorithm. The screen of the Test App is shown in Figure 8.

C. Deployment App

The positions within the building at which the BLE devices are deployed are identified in advance and are marked on building maps. It was decided to space them evenly on outer walls and in hallways, stairwells and restrooms. The positions were numbered sequentially. The location database associates each numerical identifier with the description of its location. This description includes the address of the building, the floor, and the room or area in which the device is deployed. The location database also includes the (x,y) coordinates of the device as measured from a building origin that is identified on the building maps. The latitude and longitude of this origin are also included in the database, enabling the use of algorithms

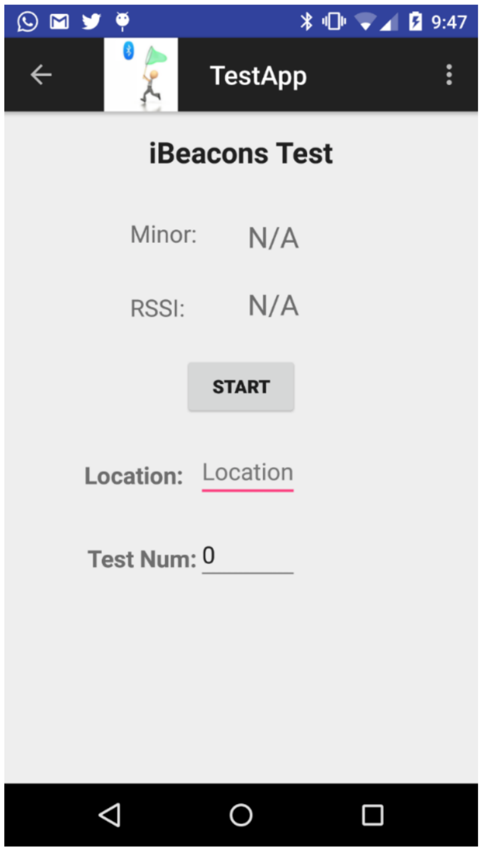


Fig. 8. Test App main screen

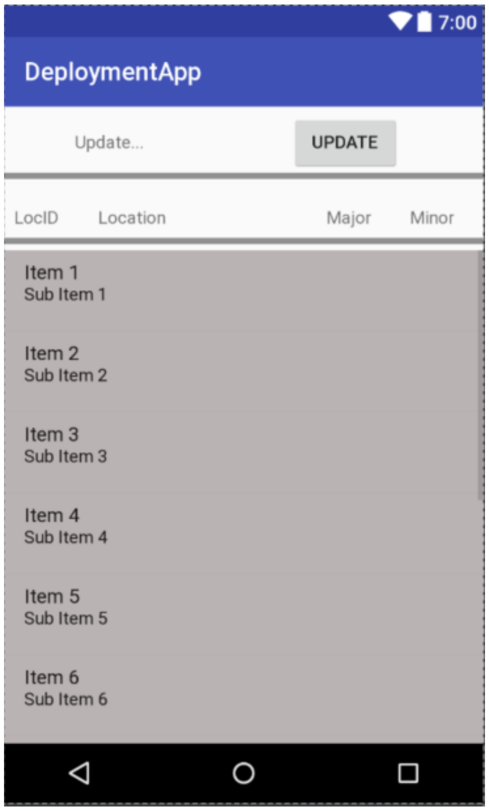


Fig. 9. Deployment App

that calculate the location of the caller as well as the location of the nearest BLE device.

The initial deployment and subsequent replacement of BLE devices involves physical labor including: finding the location at which the device is to be placed; climbing a ladder to place the device on a wall, door lintel, or ceiling; recording the device identifier and associating it with the position on the map. The Deployment App was designed to reduce the risk of errors during the association of device identifier with device location. It communicates with the Location database to provide legitimate selections to the deployer and to record the deployer's selections in the database. Using the app, the deployer selects his/her numerical position then selects the device identifier from a drop down menu that contains the identifiers of only the available and un-deployed devices. The screen of the Deployment App is shown in Figure 9. A web application that accomplishes this same task has also been created. It requires the user to authenticate him/herself and thus makes the process both more secure and simpler to use.

D. Future Smartphone Application Development

The caller's application was designed to work with the platform that consists of the array of BLE devices and the database. Other applications that use the platform could be created: Applications that help visually impaired people find their way through buildings or that provide controls of building environments are two examples. Features and functions are also being added to the Indoor Location phone application: A new version that updates the location of a caller who may be fleeing an assailant or a fire is under development. Two more applications that use the platform are under development today. One needs to access the temperature and humidity data being collected in the hierarchical deployment described in Section IV. The other is a web-based smartphone application that makes a WebRTC [9] call to a WebRTC-based PSAP that needs to obtain the caller's indoor location.

A set of APIs has been developed that allows the developer to extract data from the database without creating sql requests whose format and content may change when changes are made to the structure of the database. The APIs use standard URIs and JSON format to exchange information. The requests that an application will be able to perform using the APIs as of this writing support indoor location discovery, temperature and humidity retrieval, and algorithm grading. Location discovery for example is used to send the set of iBeacons detected by the smartphone application to the location server. The server sends back an answer including the indoor location of the user. The APIs associated with the platform will be made available to the public.

VII. The Database

The location and environmental data is stored in a MySQL database running on a server on the Amazon Web Services

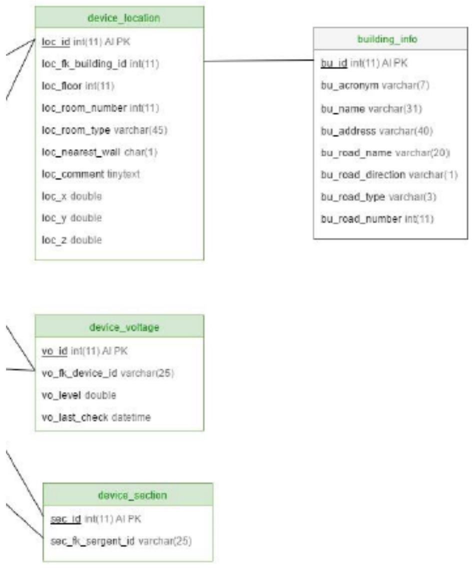


Fig. 10. Portion of Database Common to iBeacons, BLE Devices and Gateways

cloud. The database was designed to enable easy data entry, access and querying. Ease of maintenance and expansion were also considered in the design. The database is divided logically into three parts: One part contains information that is unique to the gateways; one part contains information that is unique to the iBeacons and other BLE devices; the third part includes data common to iBeacons, BLE Devices and gateways. Figure 10 shows the portion of the database that hold data applicable to all three types of device. The entire diagram is available at https://github.com/IIT-RTC-Lab/Indoor_location_database. A naming convention has been specified to enable ease of use and to guide developers as the system is expanded and is also available on github.

Information associated with the gateways includes their ID, battery status, deployment status and code version number. BLE device data includes the device ID as well as the temperature and humidity data. Data common to gateways, iBeacons and other BLE devices includes building identification, location inside the building, and battery levels.

Three types of actors will need access to information from the database: Application developers; Operations and Maintenance personnel; and Researchers. Each interacts with the database via the Location server using the APIs referred to in Section VI. Users can refer to the documentation describing the APIs to find the appropriate requests, inputs and outputs.

VIII. The Algorithms

Four algorithms for determining the location of the caller were tried. The first three identifies the location of the iBeacon closest to the caller. The fourth identifies the location of the caller using x,y coordinates and a pre-determined building floor origin. A brief overview of the algorithms follows.

A. *First step - identifying the correct floor*

For each of the four algorithms, the first step is to determine the floor from which the call is made. Three different first-step algorithms were tested: The Power Sum, Floor Count and Weighted Average. The tests were conducted at 48 test positions for a ten second interval. The POWER_SUM method sorts the detected devices into sets that correspond to the floors on which the devices are located. The sums of the RSSI values in each set are compared and the floor corresponding to the set whose sum is the maximum will be selected. The FLR_CNT method counts the number of devices on each floor whose signals are received. Again the detected devices are sorted into sets that correspond to the floors on which the devices are located. Then the floor associated with the set with the largest number of devices is identified as the floor from which the call is made. The POWER_SUM and the FLR_CNT each generated the correct floor in greater than 92% of the trials. The WTD_AVG method uses a function of the RSSI as a weighting factor on the floor number. The floor chosen is the weighted average of the floors. This method has proven accurate in 100% of the trials.

For each of the four the algorithms, the first step is to determine the floor from which the call is made. Three different first-step algorithms were tested: The Power Sum, Floor Count and Weighted Average. The tests were conducted at 48 test positions for a ten second interval. The POWER_SUM method sorts the detected devices into sets that correspond to the floors on which the devices are located. The sums of the RSSI values in each set are compared and the floor corresponding to the set whose sum is the maximum will be selected. The FLR_CNT method counts the number of devices on each floor whose signals are received. Again the detected devices are sorted into sets that correspond to the floors on which the devices are located. Then the floor associated with the set with the largest number of devices is identified as the floor from which the call is made. The POWER_SUM and the FLR_CNT each generated the correct floor in greater than 92% of the trials. The WTD_AVG method uses a function of the RSSI as a weighting factor on the floor number. The floor chosen is the weighted average of the floors. This method has proven accurate in 100% of the trials.

B. *Second step - identifying the location on the floor*

Once the floor has been picked, a process for finding the room or x,y coordinates on that floor is needed. Four different methods were tested. A description of the metric used to judge the accuracy of each of these four algorithms is given in Section VIII-C. A description of each of the four algorithms follows. The Maximum_RSSI algorithm, MAX_RSSI, selects the BLE device with the highest RSSI during a 10-second sample period. The algorithm notes the identity of the device associated with that value and queries the database for the description of the indoor location associated with that identity. The Average_Power_in_dB algorithm, AVG_DB, selects the device with the highest average power as measured in dB over the sample period. The third algorithm, Average_Power_in_dB algorithm, AVG_MW, selects the BLE device with the highest average RSSI as measured in milliwatts. The values of the RSSI's are first converted to milliwatts, then the milliwatt values are averaged, the maximum is found, and the associated BLE device is chosen to be the one closest to the caller.

The fourth algorithm, Trilateration with Least Squares fitting, TRI_LS, attempts to locate the caller rather than the beacon closest to the caller. The location is given as a floor number together with the caller's x,y coordinates on that floor relative to a pre-defined origin. The x,y coordinates of each BLE device are recorded in the Location database along with the description of the room and floor on which the device is deployed. After the floor is identified, a process of trilateration and least squares fitting is used to calculate the estimated x,y coordinates of the caller.

C. *Measurement of algorithmic accuracy*

To measure the accuracy of the various algorithms, a set of test data was used. The data was collected at pre-defined test positions whose locations were themselves recorded in a separate database. Tests were run on the subset of results for which the correct floor had been calculated. The number of data points in all cases was at least 44 of the total 48 depending on which floor picking method was used. Two datasets have been created. Dataset I uses 20 BLE devices per floor and 48 test positions. Dataset II reduces the number of BLE devices to 10 per floor by powering down half of the beacons used in the first dataset and keeps the same 48 test positions. The algorithms are run successively over each dataset and their accuracies are compared. These tests were made to study the effects of beacon density but were not optimized for the best building coverage. We learned from these tests that the coverage pattern needs to be studied as well. Walls, structural elements and other impediments to radio transmissions need also to be considered.

Figure 11 is an example of a floor map used in the testing. The blue numbers are the BlueTooth locations. The red numbers are the tester locations. The map itself is stored as a set of rectangles representing rooms and corridors with their southwest corner coordinates and northeast corner coordinates.

The accuracy measurement is defined relative to a proximity radius. A result is said to be 'Accurate to A' if the calculated location of the tester (or of the closest beacon to the tester) is within 'A' meters of the actual location of the tester (or the beacon closest to the tester.) The experimental results for the four algorithms are shown in Tables I and II. Table I shows that, in the three cases in which the nearest beacon was identified, 80% of the algorithm's calculations were accurate to within 10 meters, while 100% of the calculations were accurate to within 17 meters. Whereas in the case where trilateration was used to calculate the location of the caller, 100% of the calculations were accurate to within 10 meters. In the low density case, presented in Table II, the first three calculations were not impacted by the configuration change, while the trilateration calculation was.

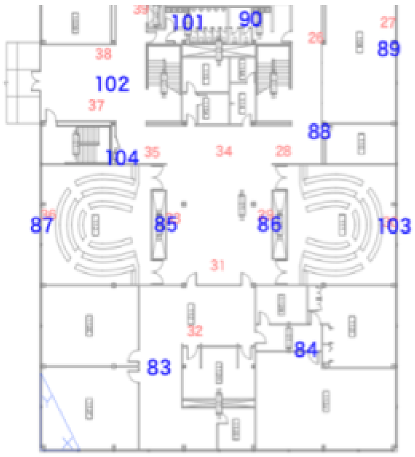


Fig. 11. Example of a floor map used for testing the accuracy of the algorithms

TABLE I
HIGH DENSITY DATASET I (20 DEVICES PER FLOOR)

Algorithm	Accuracy to 10 meters	Accuracy to 17 meters
MAX_RSSI	80%	100%
AVG_DB	80%	100%
AVG_MW	80%	100%
TRI_LS	100%	100%

TABLE II
LOW DENSITY DATASET II (10 DEVICES PER FLOOR)

Algorithm	Accuracy to 10 meters	Accuracy to 17 meters
MAX_RSSI	80%	100%
AVG_DB	80%	100%
AVG_MW	80%	100%
TRI_LS	85%	100%

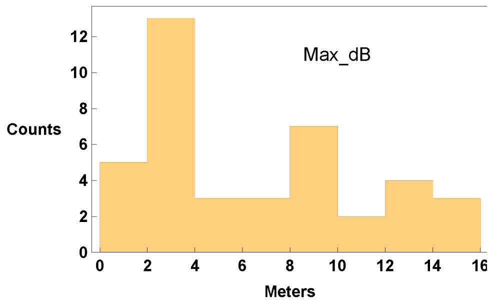


Fig. 12. Histogram of results in dataset I using algorithm Max_dB

The results for the high density distribution for algorithms 1-3, when the floor had been correctly selected, are given on the histogram in Figure 12. The results for the high density distribution of algorithm 4 are in Figure 13. The high density distribution was selected for graphing since in the low density experiment, the building-specific considerations such as walls and other structural elements had a greater impact on the results.

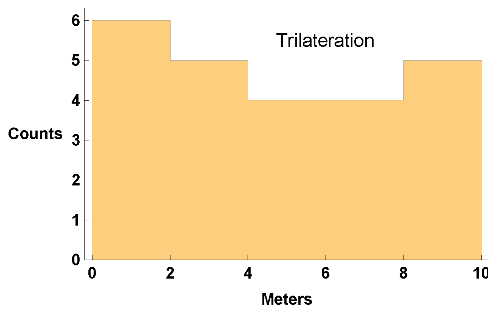


Fig. 13. Histogram of results in dataset I using algorithm TRI_LS

IX. Related Work

Our indoor location system differs from other location based services that are commercially available such as those used in shopping malls, factories, and museums. The services that use BLE to identify the location of a consumer are often concerned with using the BLE capability to invite shoppers to visit their stores or to give information about a museum display to people close to it. The services are generally focused on communicating with a person who is carrying a BLE-enabled device. Our system on the contrary does not try to communicate or connect with the caller, or to learn anything about the caller. The caller triggers events by making an emergency call and the system provides information to the caller's phone so that the phone can complete the call to an emergency services answering point in accordance with industry standards. There are commercial services designed specifically for emergency calls as well. To our knowledge these are enterprise-based and require the phone used by the caller be a device provided by or registered with the enterprise that owns the network. Our system registers the locations and identifications of BLE devices but stores no information about phones. There are also a number of applications that provide the outdoor location of a caller to the operators of private security businesses. These includeVIZSAFE, RapidSOS, BlueLight and SaferMobility. To our knowledge, these rely on GPS data and databases that provide a correspondence between geographic coordinates and street addresses. An IETF draft entitled "Indoor Location Mechanisms for Emergency Services" [10] defines the problem statement and classifies types of solutions for providing indoor location and it emphasizes the security and privacy considerations that cellular providers will need to address when providing the indoor location of emergency callers.

X. Conclusion

The system that has been developed is consistent with the Roadmap described in [1]. It provides 100% accuracy to within 21 meters in all cases tested and to within 10 meters when the trilateration with least squares algorithm is applied. The delay from the time the caller presses the emergency button to the time the call is received at the PSAP is about 3 seconds. We will continue to experiment with different distribution patterns for the BLE devices and with the hierarchical configuration described in Section IV, collecting new datasets in order to learn which ones yield the most accurate location information. Special consideration will be given to the position of walls, columns, water pipes and other structural elements that attenuate radio signals and thus impact the accuracy of the location-calculating algorithms. We will also continue to test the relative accuracy of the algorithms using the new datasets. The algorithm that uses trilateration with least squares fitting has yielded the most accurate results in most cases and the team will continue to develop it and to study cases in which it does not yield the best results in order to understand what aspects of the distribution or the algorithm cause the least accurate results. A beta test is proposed with the University's Public Safety Department in which the team will install its WebRTC PSAP and Web-based phone app for use in parallel with existing systems. Finally, a system that collects temperature and humidity data while supporting the Indoor Location Service is being installed in a campus building and the data collected will be used by building engineers and by colleagues in the structural engineering program.

References

[1] M. H. Dortch. (2014, November) Roadmap for improving e911 location accuracy. [Online]. Available: http://apps.fcc.gov/ecfs/document/view?id=60000986637

[2] "Apco." [Online]. Available: https://www.apcointl.org/

[3] "Nena." [Online]. Available: https://www.nena.org/

[4] "It real-time communications lab." [Online]. Available: http://appliedtech.iit.edu/rtc-lab/

[5] "Detailed functional and interface standards for the nena i3 solution." [Online]. Available: http://www.nena.org/Standards

[6] C. Davids, J. M. Valdecantos, B. Dworak, C. Tovar, B. R. Nandakumar, and M. Patil, "Dispatchable indoor location for mobile phones calling for emergency services," IPTComm '15: Proceedings of the Principles, Systems and Applications on IP Telecommunications. ACM, October 2015.

[7] "Bluetooth core specification." [Online]. Available: https://www.bluetooth.com/specifications/bluetooth-core-specification

[8] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler, "Sip: Session initiation protocol. rfc 3261 (proposed standard)."

[9] "Webrtc." [Online]. Available: https://webrtc.org/

[10] "Indoor location mechanisms for emergency services." [Online]. Available: https://tools.ietf.org/html/draft-marshall-ecrit-indoor-location-00