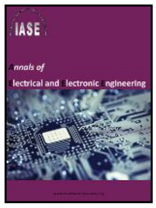
Annals of Electrical and Electronic Engineering, 2(11) 2019, Pages: 1-6

Published by IASE

Annals of Electrical and Electronic Engineering

Journal homepage: www.aeeej.org



Avaricious crowd sensing for interconnection via the internet of computing mobiles using Bluetooth low energy

E. E. Ocampo *

Department of Electrical Engineering, University of Santo Tomas, Manila, Philippines

ARTICLE INFO

Article history:
Received 2 April 2019
Received in revised form
22 August 2019
Accepted 23 August 2019

Keywords:
Crowd sensing
Internet of things
Bluetooth low energy

ABSTRACT

To relief some burden from the smartphones, we envision a new crowdsourcing architecture where low cost, low energy sensors embedded in objects around us i.e. walls, traffic lights, billboards, providing a variety of sensors depending on their context, e.g. Air Quality, temperature. These devices would carry on the sensing, processing, and broadcasting of sensor data using wireless interfaces. Smartphones on the other hand, would opportunistically discover and collect data from these devices to provide a much better, richer and energy-efficient sensing infrastructure. We chose Bluetooth Low Energy as a wireless interface to communicate with smartphones at a very low energy cost. The Internet of Things (IoT) ecosystem is currently in its early stage and already offering Bluetooth Low Energy-based sensory devices which could serve the purpose of off-device sensing. In this paper, we discuss the usage of Bluetooth Low Energy as a new energy-efficient sensing resource for crowd sensing. We focus on defining a unified Bluetooth Low Energy sensing framework, which provides a number of smart sensing schemes to ease the development of energy-efficient and context-aware crowd sensing applications.

1. Introduction

The current day smartphone offers a plethora of services, which has completely revolutionized the way we live, communicate, travel, socialize, organize, and entertain ourselves. Smartphones are becoming powerful sensing devices that are capable of collecting data from the environment around us. A variety of sensors are embedded in modern smartphones such as accelerometer, gyroscope, magnetic field sensor, ambient light sensor, temperature sensor, humidity sensor, GPS, etc. The term "crowd sensing" is now commonly being used to refer to the activity of sharing data collected by individuals with their sensing devices, with the aim to measure phenomena of common interest. Crowd sensing is a new sensing paradigm that mostly makes use of a large number of the user base who willingly participate in sensing applications. However, crowd sensing with smartphone embedded sensors is still limited by 1) the fixed number of sensors on a commodity smartphone 2) the processing and energy requirement of a specific sensing application like image processing on camera, or GPS sampling. These restrictions lead us to the direction of off-device sensing by utilizing external sensory and IoT devices deployed everywhere around us.

Our everyday spaces such as homes, offices, streets, public spaces, etc., are more and more being augmented with ubiquitous sensors to improve our lifestyle and well-being, forming what is known as the Internet of Things (IoT). It has been reported that the connected devices have already surpassed the connected people in 2008, and the number is estimated to reach 50 billion in 2020. Plants are being tagged with moisture sensors to keep your garden lush and green, indoor pollution sensors are being placed to monitor the indoor air quality, automatic parking sensors are being placed on parking spots to detect the utilization activity. These and hundreds of other IoT sensing applications demand dedicated and continuous Internet connectivity, which cannot be always possible, to share their sensor data. If those devices were made to publicly advertise their sensor data in their wireless area network, and nearby smartphones could crowd sense these packets and forward them to the proper servers, the dedicated and continuous internet connectivity requirement could be relaxed. As such, crowd sensing projects would benefit not only from smartphone sensor data but also from the sensor data of everyday IoT devices around us.

Designing off-device avaricious crowd sensing in the IoT ecosystem requires a communication interface that is energy efficient, easily supportive of this feature and popular in most smartphones and IoT devices. In recent years, Bluetooth Low Energy has emerged as a new wireless personal area network (WPAN) technology. Currently, there are more than millions of Bluetooth low energy enabled accessories shipped in 2013 and nearly 2.6 billion expected to ship by 2016. As such, more and more IoT devices are using Bluetooth low energy as their communication protocol. Alongside, almost all the latest versions of mobile platforms have been working to provide native support for Bluetooth low energy, and are already starting to appear in the market.

The focus of this paper is on the utilization of wide-spread availability of smartphones as an avaricious sensor data collection network; with the aid of external IoT sensory devices communicating via Bluetooth low energy to provide wide-spread and energy-efficient crowd sensing applications. In this paper, we design a framework that leverages the usage of Bluetooth low energy as a new sensing resource. Our framework utilizes advertisements in Bluetooth low energy protocol as a means for carrying useful sensor data. Without the need for proper associating with Bluetooth low energy, our approach provides connectivity to any nearby IoT sensor while maintaining low power. Our framework offloads the burden of sensing from

* *Corresponding Author.*
Email Address: e.e.ocampo@ust.edu.ph
https://doi.org/10.21833/AEEE.2019.11.001

smartphone embedded sensors to off-device sensors in the environment. Thus reducing the requirement needed by smartphone processing and power, and adds support to millions on sensor-based IoT devices with different supported sensors. Furthermore, we describe the implementation of the framework into an open-source library called BLESS to support Android-based smartphones. This library can be used by developers and researchers to easily write android applications that provide Avaricious Crowd sensing for IoT without worrying about dealing with low-level maintenance of Bluetooth low energy android functions, or detailed know-how of the communication technology. It also alleviates the issues raised by heterogeneous Bluetooth drivers from different manufacturers. The library acts as a layer to support avaricious crowd sensing out of the box.

In section 2, we briefly discuss the challenges of existing crowd sensing approaches and provide a motivation for the utilizing the Bluetooth low energy enabled devices for crowd sensing. Section 3 introduces Bluetooth low energy and the Advertisement fames. In section 4, we provide the high level requirements which lay the foundation for the Bluetooth low energy framework. In section 5, we propose a Bluetooth low energy Framework which provides means to easily develop Bluetooth low energy sensing applications. In section 6, we present some related works and provide a comparison. In section 7, we discuss future directions and provide a conclusion in section 8.

2. Bluetooth low energy sensing-motivation

It takes quite a lot of effort to design and develop systems to be deployed for large scale crowd sensing experiments. This effort increases mostly because of the heterogeneity of the smartphone hardware and available platforms (Ganti et al., 2011). To easily enroll participants in the crowd sensing experiments, mobile applications must be supported by a variety of available hardware and platforms. The other obstacle is the need for running application continuously in the background putting a burden on the resource-constrained smartphones in terms of energy and computation (Ganti et al., 2011)

The alternative approach of outsourcing sensing tasks to external devices relies on participants carrying both a sensor unit and a smartphone, which creates more burden for the users and only suitable for a selected target group study. Other trends such as the IoT ecosystem relies on a large number of Internet-connected devices, which would require continuous network connectivity and increase the demand for network bandwidth. In contrast, Bluetooth low energy enabled devices to remove the requirement of direct Internet connectivity with their detectability and communication by normal smartphones.

Contrary to the smartphone-based crowd sensing, Bluetooth low energy sensing works by discovering the nearby Bluetooth low energy devices, query data from them and store the desired information. This approach comes under the paradigm of avaricious sensing technique, where the mobile application runs in the background and unobtrusively listens for the desired Bluetooth low energy devices. Once a Bluetooth low energy device is discovered, it exchanges data automatically without any user intervention. The advantage of Bluetooth low energy sensing is that, instead of relying on a large user base, it relies on the deployed low-cost Bluetooth low energy devices. It considers a nearby smartphone user as an opportunity to pass on the data, which then transferred to the cloud for further processing. This approach relieves the requirement of continuous internet connectivity for IoT devices and the need for a large smartphone user base for crowd sensing. It also relieves the participants from the burden of active sensing which costs them battery lives. These two scenarios, i.e. Smartphone sensing vs Bluetooth low energy sensing, are depicted in Fig. 1.

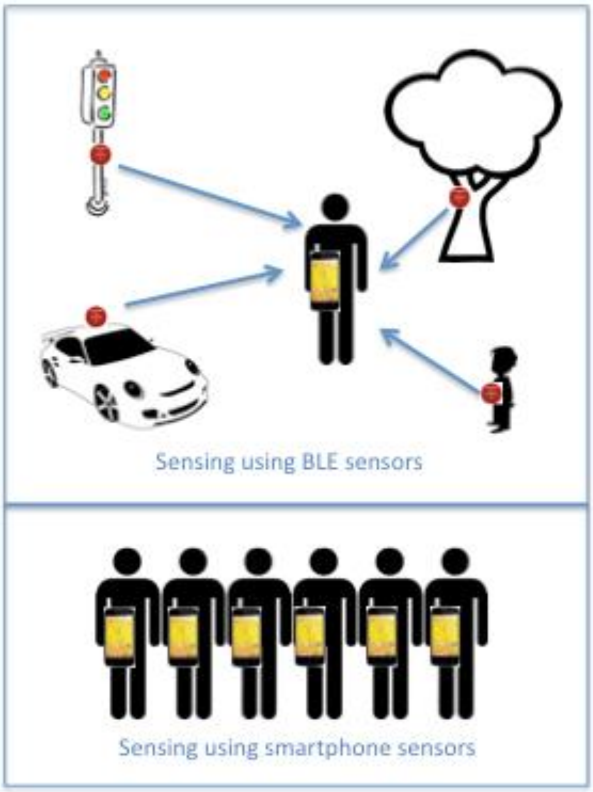


Fig. 1. BLESS architecture releases smart-phones from sensing constraints, and places the burden on nearby widely available IoT sensing devices.

Bluetooth SIG defines a number of application-centric profiles for Bluetooth low energy; including healthcare profiles, sports and fitness profiles, proximity profiles, alerts and time profiles. For all these profiles and relevant applications, a Bluetooth low energy device must be detected and connected to perform data exchange operations. Bluetooth low energy devices work by sending advertisement packets on specific channels. Master devices (smartphones) detect the nearby Bluetooth low energy devices by listening for the advertising messages on the advertisement channels. Once, a device of interest is detected, a communication channel is established by connecting to the device to perform data transfer. Currently, Bluetooth low energy enabled tiny devices are available in the market to track personal belongings and trace lost items. The advertisement packets can also include a limited bit of payload for short data transfer. The nature of Bluetooth low energy discovery process with minimum payload transfer can be exploited to support a certain class of applications. Imagine environment sensing application utilizing tiny Bluetooth low energy devices, mounted with temperature, humidity and $CO2$ emission sensors spread over a geographic region or in an industry vicinity. A simple advertisement sending Bluetooth low energy tags could be used to understand the presence, mobility patterns and crowd behavior in certain events over a period of time. All these applications require considerably less number of participants who are carrying smartphones to detect the nearby Bluetooth low energy devices, store the detections, and periodically send them to the server.

While the Bluetooth low energy technology is still in its infancy, there is a need to provide an easy to use and unified framework for Bluetooth low energy sensing applications. In this work, we particularly focus on the discovery process of Bluetooth low energy devices and propose a unified framework, which integrates various, vendor-specific, Bluetooth low energy scanning libraries and provides a high-level interface to easily develop Bluetooth low energy sensing applications. In addition to that a wide range of smart sensing, and data management operations are supported to further ease the researchers from development and focus more on data analyses.

3. Bluetooth low energy overview

In this section, we provide a technical overview of the Bluetooth low energy technology and how it works. Specifically, our objective is to describe the Bluetooth low energy advertisement process as a preliminary requirement for the avaricious crowd sensing framework explained in the next section.

Bluetooth low energy is an extension to the Bluetooth classic a well-known Personal Area Network (PAN) wireless protocol. It was introduced in the Bluetooth 4.0 specification and is optimized to enable wireless communication between a smartphone and a low duty cycle and low power device located in close vicinity.

While Bluetooth low energy and classic Bluetooth share some similarities such as using the 2.4GHz ISM band and Frequency Hopping Spread Spectrum (FHSS)-based modulation, Bluetooth low energy has 40 channels compared to classic Bluetooth's 79 channels. As such, they are incompatible and cannot communicate. Modern smartphones equipped with Bluetooth 4.0 interfaces, support both Bluetooth low energy as well as classic Bluetooth communications.

Bluetooth low energy devices can operate as a master or a slave; that is, as a scanner, or an advertiser, respectively. As a slave, the Bluetooth low energy device periodically transmits advertisement packets on the 3 advertising channels (i.e. 37, 38, and 39) one after the other, waiting for a connection request from an interested master device with the intention of "pairing". As a master, the device listens for advertising information transmitted by other devices and chooses whether or not it wishes to connect to those slave devices. Smartphones usually operate in master mode, since they are usually more power-efficient and powerful than external sensor devices.

The Bluetooth Specification defines the Bluetooth low energy Advertisement packets as shown in Fig. 2**.** The fixed Preamble and Access Address differentiates the Advertisement packet from other Bluetooth low energy packets. For broadcasting applications, the header of the advertisement packet PDU defines the packet type; whether the device is supposed to establish a paired connection (ADV_IND), or if it is just an advertisement frame for a device that does not support connectivity establishment (ADV_NONCONN_IND and ADV_SCAN_IND). The payload of the advertisement packet contains the advertisers' address along with the user-defined advertised data (AD). The AD Type is set as Manufacturer Specific Data (MSD) which allows for up to 28 Bytes of user-defined information.

The objective of our framework is to utilize the MSD to define sensor values instead of using the connection establishment feature of the Bluetooth low energy standard which may require authentication between smartphone and device. For crowd sensing purposes, by using this feature any smartphone within the proximity of the device can read the sensor value embedded in the advertisement frame. More on this in the next section.

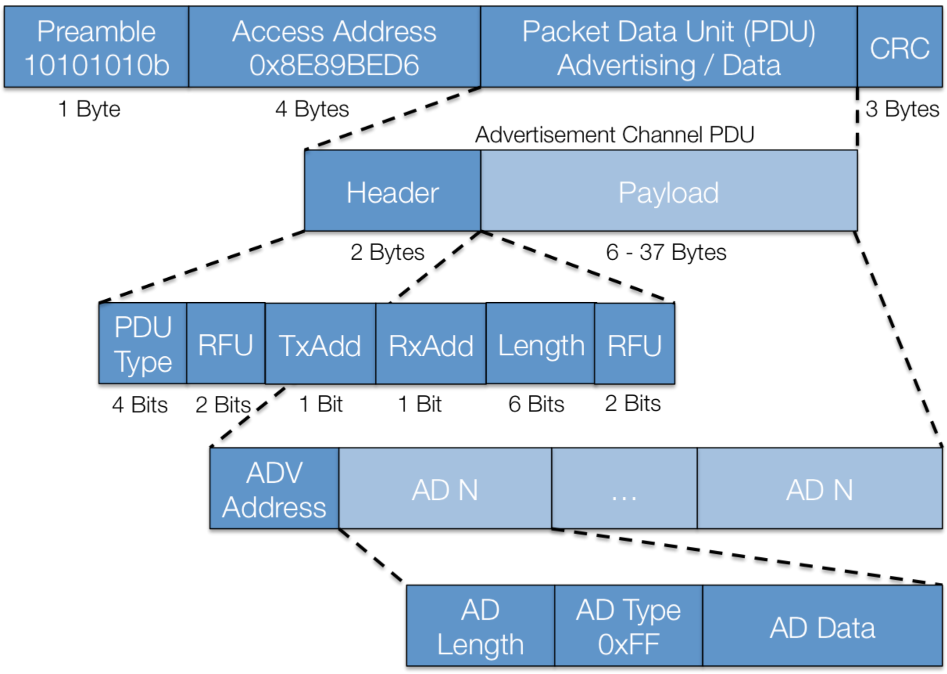


Fig. 2. Bluetooth low energy advertisement frame format.

4. Framework overview

Designing a participatory system is a non-trivial task since it requires a set of computing and design challenges. The sensing application requires considerable design effort to manage data collection for Bluetooth low energy tags, running in the background, as well as managing the data offloading. Unlike other mobile platforms, the Android platform is convenient for such support, as it is being adopted by a number of hardware vendors due to its open-source nature and high-ratings from customers. As a result, the proposed Bluetooth low energy Framework is only targeted for the Android operating system. Before diving into the framework details, it is desirable to discuss the key requirements which have led us to the development of this framework.

1. Development of a Bluetooth low energy sensing crowd sensing application which could work on heterogeneous platforms
2. Continuous data collection on regular intervals or defined schedule
3. Provide smart sensing schemes, such as time-based, activity-based, location-based etc.
4. Auto upload collected data on regular intervals to the predefined server
5. Provide smart data transfer schemes
6. Adaptive scanning based on various parameter changes e.g. Battery level,

Providing all these features in a unified library will make easier to develop crowd sensing applications by setting the parameters as required.

4.1. Bluetooth low energy scanning

The crowd sensing experiments are usually targeted towards a general audience using a wide variety of smartphones. During the development phase, we used Google Nexus 4 as it is shipped with a large number of sensors including Bluetooth Smart, and it also runs the latest version of Android. The latest Android version was 4.3 Jellybeans during the development time. The Bluetooth low energy support was officially provided on Android 4.3+ by Google. The Bluetooth low energy support was further tested on Samsung Galaxy S3 and S4 which were shipped with Android version 4.1.2 and 4.2.2 respectively. Both phones provided Bluetooth Smart hardware support but the native Bluetooth low energy support from the official Android platform was included only in versions 4.3 and later. To overcome this shortcoming, Samsung provided its own APIs to enable developers to make use of Bluetooth low energy features in their applications. Consequently, we also had to patch our framework, with Samsung specific libraries, such that it could be supported by a larger number of devices. The framework is designed in a manner that the support for other vendor-specific Bluetooth low energy libraries could easily be integrated.

4.2. Data management

Someone having experience in crowd sensing application development can easily understand the overhead of maintaining sensed data and arranging in files. The library provides an easy to use data management features, which are easy to enable, configure and saves a lot of coding effort.

4.3. Transport management

The collected data need to be transferred to the server. The data transmission could happen using different transfer modes, and different transfer channels. The availability of transport manager's easily configurable parameters allows doing all this processing without giving extra burden to the application developer.

4.4. Smart sensing

It is not wise to continuously scan for the available Bluetooth low energy devices and collect data. Such aggressive scanning effects battery life and also it generates a lot of useless data. To avoid these problems, it would be beneficial to do smart sensing and making use of the participant's schedule, location, activity to trigger the Bluetooth low energy scanning for data collection.

5. Framework architecture

The Bluetooth low energy framework architecture is shown in Fig. 3. The contribution of this framework is presented in the top layer, which sits on top of the Android platform. In the following sections, we explain each component and its features and how do they interact with each other. The library of the framework allows certain commands to be called specifically as the required application as seen in Fig. 4.

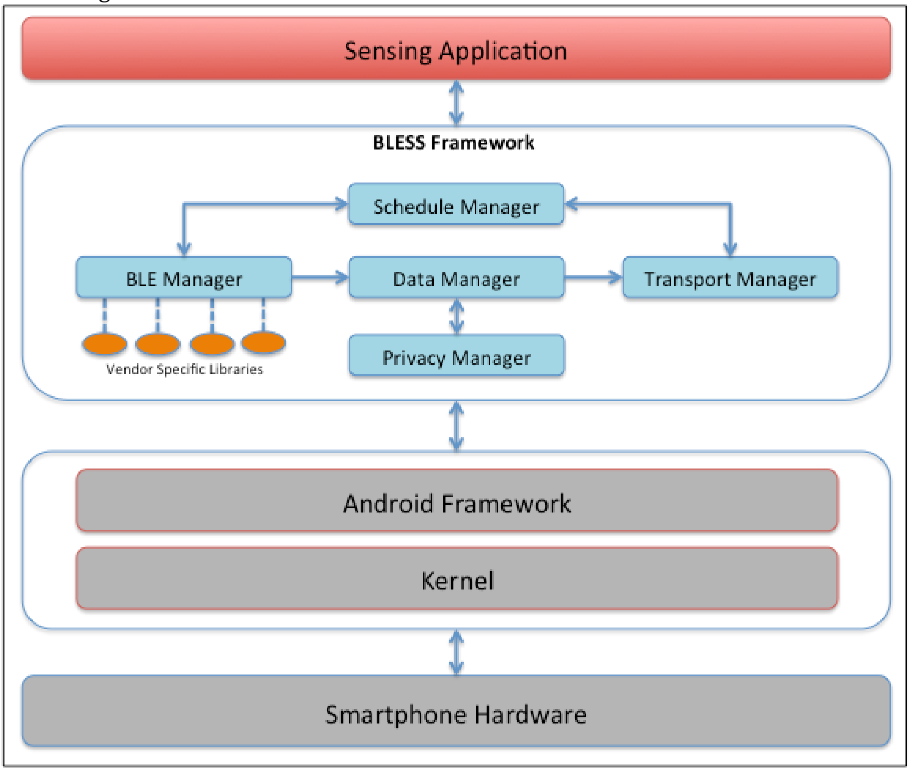


Fig. 3. BLESS framework library architecture.

5.1. Bluetooth low energy manager

This component is responsible to provide a unified interface to perform all tasks specific to Bluetooth low energy scanning. It is intelligent to identify the device and platform version to make use of vendor-specific libraries as mentioned in section \ref{sec:BLE_Scanning}. It provides a simplistic scanning initiation and termination mechanism irrespective of the device on which the framework being used. Furthermore, the Bluetooth low energy manager provides two levels of Bluetooth low energy scanning i.e. aggressive and filter based.

Aggressive scanning means when the scanner is started it will capture all possible Bluetooth low energy devices available within the range. This might be useful for applications being used in a closed vicinity and wants to receive data from all the available Bluetooth low energy devices e.g. inventory management, health monitoring, etc.

Filter based scanning provides a way to define filters to capture only a subset of devices from all the available nearby devices. This feature is suitable for applications deployed on a larger scale and just want to capture Bluetooth low energy devices that are equipped with a specific type of sensors, e.g. temperature, humidity, CO_2 emission, etc.

5.2. Data manager

The data manager is a very important component that stores all the called data as soon as it is reported by the Bluetooth low energy manager. The Data Manager is capable to save the data in different formats e.g. CSV or database as indicated by the application developer. The Bluetooth low energy manager exposes a ScanResultHandler interface which could be used to define different data handling mechanisms easily e.g. displaying the data on the user-interface, making a decision based on available data etc. The Data Manager also provides an option to handle data in batches, rather than saving each reported data item separately which creates a lot of files I/O operations.

```
bleManager = new BLEManager(getApplicationContext());
bleManager.setScanPeriod(5000);
bleManager.setScanResultHandler(new ScanResultHandler(getApplicationContext(), this, mLeDeviceListAdapter));
bleManager.getDataManager().enableLogging(DataManager.FORMAT_CSV);
bleManager.getDataManager().enableBatchUpdate();
bleManager.getTransferManager().enableTransfer(TransferManager.TransferProtocol.HTTP,
                                               "http://yourserveraddress.com/bledata",
                                               "username",
                                               "password");
bleManager.getTransferManager().setDataParams(compressData, deleteAfterUpload, archiveData);
bleManager.getTransferManager().scheduleTransfer(TransferManager.TransferSchedule.HOURLY,
                                                 TransferManager.TransferChannel.WIFI,
                                                 TransferManager.TransferScheme.BULK_TRANSFER);
bleManager.startScanning();
```

Fig. 4. BLESS library commands.

5.3. Privacy manager

Privacy is a major concern in almost all of the crowdsourcing experiments. Participants do not want to be exposed, what data they are collecting and what locations they have been at what times. If privacy and security issues are not properly handled, the data could be leaked at different stages e.g. by acquiring access to the participant's smartphone or by intercepting the data transfer while uploading data to the server. To overcome this issue, the Privacy manager anonymizes the identifiable entities e.g. MAC addresses, location information, etc., and applies data encryption to secure the on-device data and data transfers.

5.4. Transport manager

One of the most important parts of the crowd sensing applications is to transfer the data to the server. It involves careful selection of data channels, transfer mode, data compression, data archiving, and transfer frequency. The transport manager component of our framework provides a selection of all these features.

Transfer frequency

A range of triggers is supported to start the transfer manager on specific time intervals. For certain applications, it might be necessary to receive real-time data and define the transfer frequency on a shorter time scale, assuming the security of internet connectivity. While for other applications it might not be required to fetch the data so frequently and just needed to perform data transfers once or twice per day.

Compression and archiving

For large scale experiments, it is desired to transfer the data in a compressed format and delete the data as soon as it is successfully transferred to the server, to avoid burdening the network bandwidth and user device. On the other hand, in smaller group scale or personal scale applications, it might be necessary to keep the log for certain days to display on-device statistics. The transport manager component provides settings that can be configured to make use of these features as desired.

Transport modes

The data can be transferred to a server via different modes. People prefer one model over the other mainly because of performance and efficiency reasons. In crowd-sourcing experiments, people are also interested in transferring them directly to cloud storage, such as GoogleDrive, SkyDrive, DropBox etc. In this framework, currently, HTTP and FTP transfer modes are possible.

Transport policy

Transfer policy plays the main role in selecting the transfer channel for data transfer. The participants collecting data using smartphones usually want to avoid the data upload using 3G or 4G connections primarily because of the cost factor. In critical applications, where data must be uploaded on time, the 3G and 4G connections are utilized where no Wi-Fi connectivity is available. The Transport manager provides a mechanism to define the transfer policy to be used, such as WIFI_ONLY, 3G_ONLY, WIFI_3G_PRIORITY and 3G_WIFI_PRIORITY. The WIFI_3G_PRIORITY means 3G is of higher priority and sends the data right away using a 3G connection if the Wi-Fi is not available. The 3G_WIFI_ PRIORITY option first checks if the WIFI is not available, and waits for the RETRY_DURATION for Wi-Fi connectivity. As soon as Wi-Fi is connected the data gets transferred. If the Wi-Fi is not available for RETRY_LIMIT, the data gets transferred using 3G. The RETRY_DURATION and RETRY_LIMIT parameters are also configurable.

5.5. Schedule manager

The schedule manager is handling the smart sensing part of the framework. It provides a number of triggers to start the Bluetooth low energy scanner. These smart sensing triggers come handy in scenarios where the application is continuously running in the background and sensing intelligently without any user intervention and without causing unnecessary computational and sensing overhead. The functionality of these triggers are as follows:

Time-based triggers

The simplest possible way to use the scanner is by providing time-based scheduling. It is possible to provide scan durations with scan repeat interval to have a repetitive scanning cycle. It is also possible to provide a schedule in terms of start time and end time and have a repetitive scan during this time. This feature would be useful to restrict unnecessary scanning when the participant is at home or at the office.

Region-based triggers

In certain scenarios, the sensing is only required in specific regions. This option is provided with the help of the latest Geofencing API provided with Android Location APIs. The user of our framework just needs to specify a latitude, longitude and the proximity radius to define a Geofence. The framework will keep a check on the participant's location and as soon as he enters the defined Geofence the Bluetooth low energy scanner will be started. The Bluetooth low energy scanner keeps running periodically as long as the participant stays in the region.

Activity-based triggers

This trigger is based on a new Activity Recognition API provided with Google Play Services. The main function of this trigger is to define scenarios where Bluetooth low energy sensing is required based on user activity. Consider a use case where Bluetooth low energy sensing is required when the user is driving a car. Currently, supported activities include STILL, WALKING, ON_FOOT, ON_BICYCLE, IN_VEHICLE, TILTING and UNKNOWN.

Adaptive scheduling

This smart scheduling trigger works in coordination with battery status. For long-running and computationally demanding sensing experiments, it is important to consider the battery drain. The framework keeps monitoring the device battery and adapts the Bluetooth low energy scanner durations depending on the available battery life. This feature could be disabled by the application developer depending on the requirement.

Custom triggers

The framework is designed in a manner that it is easier to define a custom trigger e.g. by creating a mix of the region, activity, and time parameters.

6. Related works

There are many open-source frameworks that make use of smartphone sensing capabilities for participatory data collection. Some of the frameworks are Medusa (RA et al., 2012), Funf (Aharony and Gardner, 2012), MyExperience (Froehlich, et al., 2007), FeelTheWorld (Phokas et al., 2013), EmotionSense (Rachuri et al., 2010). Some of the frameworks already worked on the notion of triggering mechanism, data management, and easy sensor interfacing but mainly focused on smartphone sensors as a data source. To our knowledge, we are the first to explore the integration of Bluetooth low energy for participatory sensing applications. Our work is mainly focused on providing smart Bluetooth low energy sensing techniques.

Google has recently introduced Eddystone; an open protocol beacon format that can be used to by Bluetooth low energy manufacturers to send data to android and iOS devices. Eddystone can support our architecture by making it easy for developers to send the sensor values they want without needing to go down to the beacon level. We believe that the introduction of Eddystone raises the importance of our architecture and makes it more viable.

7. Future directions

We envision a platform for Bluetooth low energy scanning and data collection system based on the proposed framework. The platform would be composed of three components: 1) A Bluetooth low energy scanning and data collection mobile application. 2) A web server application for new experiments creation. 3) A data analysis tool to provide basic data filtration and analytics.

1. A Bluetooth low energy scanning and data collection mobile application
2. A web server application for new experiments creation
3. A data analysis tool to provide basic data filtration and analytics

The mobile application would make use of the proposed framework and provide a dynamically reconfigurable system, by downloading experiment specific configuration files from the server. Once the configuration file is downloaded, the system configures the application dynamically based on the parameters provided in the file. These configuration settings could include e.g. smart sensing strategies, transport policy parameters, new server details, and all other parameters which were discussed in Framework Architecture in Section 5.

The web server application would provide administrative control to create and define new experiments at run time with all the configuration parameters. These newly created experiments would be available on users' mobile application, from which he/she can select the desired experiments to be a part of.

The analysis tool would also be a part of the webserver application. This tool will allow to get insights into the collected data and play with the data in real-time by applying selected data filtration and analysis techniques.

This platform would allow it to be used for different crowdsourced studies relying on detection and data collection from Bluetooth low energy devices. This approach will save a lot of effort being put independently by the researchers for different crowdsourcing experiments.

8. Conclusion

Considering the recent advancements in the Internet of Things (IoT) ecosystem and the integration of Bluetooth Low Energy support in modern-day smartphones, we realize the potential of Bluetooth low energy-based sensing and its applications. To this extent, we have developed a Bluetooth Low Energy Smart Sensing framework, which provides an easy development environment for Bluetooth low energy-based crowd sensing applications. The framework is composed of a number of components such as a unified Bluetooth low energy scanning interface, Data management, Transport management, and smart sensing triggers for scanning schedule management. To follow the future directions, the Bluetooth low energy smart sensing framework is continuously being developed and it is already being utilized in projects underway at our center. The framework aims to provide basic functionality for each component mentioned in Section 5. The remaining features are being developed and old features are being revised depending on the feedback of framework users. We hope that by releasing this library as open-source, we would get support as well as valuable feedback from the community.

References

Aharony N and Gardner W (2012). Funf developer. Available online at: https://bit.ly/39FzwwJ

Froehlich J, Chen MY, Consolvo S, Harrison B, and Landay JA (2007). MyExperience: A system for in situ tracing and capturing of user feedback on mobile phones. In the 5th International Conference on Mobile systems, Applications and Services: 57-70. https://doi.org/10.1145/1247660.1247670

Ganti RK, Ye F, and Lei H (2011). Mobile crowdsensing: Current state and future challenges. IEEE Communications Magazine, 49(11): 32-39. https://doi.org/10.1109/MCOM.2011.6069707

Phokas T, Efstathiades H, Pallis G, and Dikaiakos MD (2013). Feel the world: A mobile framework for participatory sensing. In the Mobile Web Information Systems, Springer Berlin Heidelberg, Germany: 143-156. https://doi.org/10.1007/978-3-642-40276-0_12

Ra MR, Liu B, La Porta TF, and Govindan R (2012). Medusa: A programming framework for crowd-sensing applications. In Proceedings of the 10th International Conference on Mobile Systems, Applications, and Services: 337-350. https://doi.org/10.1145/2307636.2307668

Rachuri KK, Musolesi M, Mascolo C, Rentfrow PJ, Longworth C, and Aucinas A (2010). EmotionSense: A mobile phones based adaptive platform for experimental social psychology research. In the 12th ACM International Conference on Ubiquitous Computing: 281-290. https://doi.org/10.1145/1864349.1864393