Sensing WiFi Network for Personal IoT Analytics

Utku Günay Acer, Aidan Boran, Claudio Forlivesi, Werner Liekens, Fernando Pérez-cruz, Fahim Kawsar
Bell Laboratories
{firstname.lastname}@bell-labs.com

***Abstract*—We present the design, implementation and evaluation of an enabling platform for locating and querying physical objects using existing WiFi network. We propose the use of WiFi management *probes* as a data transport mechanism for physical objects that are tagged with WiFi-enabled accelerometers and are capable of determining their state-of-use based on motion signatures. A local WiFi gateway captures these probes emitted from the connected objects and stores them locally after annotating them with a coarse grained location estimate using a proximity ranging algorithm. External applications can query the aggregated views of state-of-use and location traces of connected objects through a cloud-based query server. We present the technical architecture and algorithms of the proposed platform together with a prototype personal object analytics application and assess the feasibility of our different design decisions. This work makes important contributions by demonstrating that it is possible to build a pure network-based IoT analytics platform with only location and motion signatures of connected objects, and that the WiFi network is the key enabler for the future IoT applications.**

I. Introduction

The era of the Internet of Things (IoT) has arrived. As more and more everyday objects get a digital makeover, they are reshaping our experience with the physical world through new, useful, exciting and sometimes entertaining smart services. The common facet of all these connected objects (be it a home appliance, a thermostat, a light or a wearable accessory) is that they collect data that is produced *by* or *about* people to offer value-added services. One service that is increasingly becoming popular in this space is the search for physical objects' location and state. Although conceptually simple, this service uncovers numerous application opportunities in the area of physical analytics for personal, public and industrial connected objects. Indeed, we expect that search will become one of the most important services in the connected object space, following the trend that we observed for the search for documents (webpages, blogs, multimedia files, etc.) on the web.

The problem of discovery and search of real-world entities has been extensively studied in the literature. A number of projects have extended the conventional *keyword-based* web search methods to discover a ranked list of connected real world entities (places and things) by leveraging static meta data [1]–[3]. A variant of this approach is to locate and query physical things by processing their sensor streams in real-time [4], [5]. Real-world search has also gained substantial attention from the industry, as multiple commercial products are now available that use Bluetooth tags to locate objects with smart phone applications [6], [7]. The former two approaches, although useful, come with high deployment and management costs due to dedicated platforms (ZigBee, RFID, Mote, etc.). While in the latter, the search range is limited to the smart phones' proximity. Based on these observations, we argue that it is the time to reflect on the role of these dedicated sensing and communication infrastructures that are proven to be limited in scale and adoption.

Fortunately though, WiFi is one of the wireless networks that is ubiquitously available at a global scale. Upcoming WiFi standards are power efficient with long range coverage capabilities (e.g., IEEE 802.11ah) and WiFi chips are becoming impressively small and low-cost[1]. We strongly believe that WiFi network will be the key communication fabric in future IoT applications across domains. Many past works have successfully leveraged WiFi network signals for offering indoor localisation [8]–[10] and thereby uncovered opportunities for interesting location based services including people and space analytics. In this work, we further extend this body of research and explore to what extent WiFi network can be used for sensing and learning about connected objects. Our approach is based on the premise that connected objects' movement data extracted from WiFi network signals carries vital information to model their spatio-temporal usage pattern.

To this end, we present a network-based platform in which WiFi network signals provide the foundation for connected object analytics. Physical objects are tagged with small WiFi-enabled accelerometers that they use to determine their state-of-use. These objects can report their state changes using WiFi management signals [11]. These signals are captured and stored by a local WiFi gateway after annotating them with a semantic location label based on proximity ranging. Personal object analytics applications can search and query the aggregated view of these local data through a cloud-based query server. Accordingly, we offer three contributions in this work.

1) We are the first to show the feasibility of developing a WiFi-only solution for connected object analytics which radically minimises deployment and management cost.
2) We present the design, implementation and evaluation of an end-to-end personal object analytics platform build around only location and motion signatures of connected objects. We show that these two pieces of information are enough to develop useful IoT applications by illustrating a personal object analytics application.

[1] ESP chip used in this work costs $3 per unit including shipping.

3) We demonstrate how the design decisions and corresponding algorithms used in this work address critical technical challenges with respect to energy, inference accuracy and computational overhead.

Finally, this work has also important secondary side-effects for preserving end-users' privacy as sensitive personal data remains in the user premises with limited access and these open up interesting opportunities to develop privacy-aware physical analytics applications in the IoT space [12], [13].

II. Design Challeges & Decisions

The process of locating, querying, and presenting the aggregated states of physical objects poses complex multi dimensional challenges. In the context of this paper, we consider a subset of these challenges which are detailed below[2]. These design cardinals provide the foundation of our system design decisions.

Uniform Features: Physical objects are diverse and heterogenous in terms of functionalities and descriptions. With digital instrumentations, these objects are augmented with additional functional features that are defined by the designers without a common vocabulary [14], [15]. Due to this heterogeneity, defining a uniform search feature is one of the most challenging aspects of designing an IoT system. To address this problem, we have limited the features into two generic ones, namely *location*, and *state-of-use*. To further simplify, the latter is determined by movement only with an assumption that most physical objects' motion characteristics provide enough cues for determining their state-of-use, i.e., if a physical object is detected to have constant motion over a designated time window, the object's is considered to be "in-use" otherwise "not-in-use". For most of the personal objects that we consider in this work, this coarse grained state-of-use is enough. Furthermore, we argue that these two features, i.e., location and state-of-use annotated with time, are sufficient to generate useful physical analytics across various consumer IoT applications, e.g., quantified self, energy awareness, predictive maintenance, etc.

Seamless Configuration: A personal object analytics platform should be easy to setup, function without manual calibration, and should include any objects. Most of the consumer-faced connected objects operate in isolation with dedicated applications. Hence, it is often difficult to interoperate with these objects without contacting their cloud-hosted interfaces. In our approach, we aim to include these connected objects without relying on their native interfaces. In addition, we also aim to include ordinary physical objects that are not connected yet. We achieve this by using a small tag equipped with a WiFi interface, an accelerometer and a battery. This tag is pre-programmed and requires zero configuration. All that a user needs to do is to attach, turn the tag on and associate the tag with an object through a companion application. This tag is then used to extract location and state-of-use data of a physical object independent of its primary established functionalities.

WiFi-based Proximity Ranging and Data Transport: Indoor localisation is a heavily investigated topic with decades of research [8]–[10], [16]. For the scope of this work, we limit our discussion to Received Signal Strength Indication (RSSI) based localisation. One class of indoor localisation is proximity ranging that provides relative location instead of absolute location. Essentially, in proximity ranging, when a mobile object is detected by a single reference node, it is considered to to be collocated with it. When more than one reference node detects a mobile object, it is considered to be collocated with the one that receives the strongest signal. We argue that, for most of the consumer IoT applications, proximity ranging is enough as it can offer room-level location with minimal calibration effort. Past research on proximity ranging has used stationary physical objects as reference points with rich semantic labels [1], [3], [17], [18]. Grounded upon these works, we have also opted for tagging stationary physical objects to act as reference points (e.g., coffee machine area, kitchen, laundry room, etc.). As discussed earlier, we use low-cost WiFi-enabled tags due to the ubiquity of WiFi network. However, WiFi is known to have high energy expenditure. To mitigate this energy challenge we have taken two careful decisions. First, we use one of the WiFi management frames, *probe*, for transporting state-of-use data from connected objects [19] and thus avoid energy demanding application level processes (e.g., HTTP based communication that requires exchange of multiple packets to establish and persist a connection). Second, we use an adaptive technique to put the tag into the deep sleep cycle when no motion is detected. These two optimisation decisions lead to a gain of 30% on the battery life. These probes emitted from the physical objects are also used for proximity ranging by the reference nodes. We rely on a Support Vector Machine (SVM) based classifier for proximity ranging with an accuracy of 84%.

Realtime Query with Dynamic Content: As discussed in [4], there are multiple alternatives for query processing. Pull-based approaches are demand driven, and provide real-time dynamic content with reasonable communication overhead (which is primarily caused by indexing and searching tasks). Push-based approaches on the other hand, may or may not have fresh data (depends on pushing scheme and frequency) and often demand high communication expenditure. In our approach, we have opted for a mixed solution, i.e., objects push their state and location changes at real-time, but the data remains in the local network. When a user queries for data, a cloud-based query server pulls the data from the local network after resolving the host location using an index service. We argue that this decision offers a balanced approach, i.e., minimizes communication overhead without compromising the data freshness.

Privacy-Aware Local Data Storage: Instrumented physical objects essentially collect data to offer value-added services. This ubiquitous data collection, unfortunately, raises the important issue of potential privacy and information exposure risks [20]. There is a danger that people will not adopt

[2] For a complete view on the design space please look at [4].

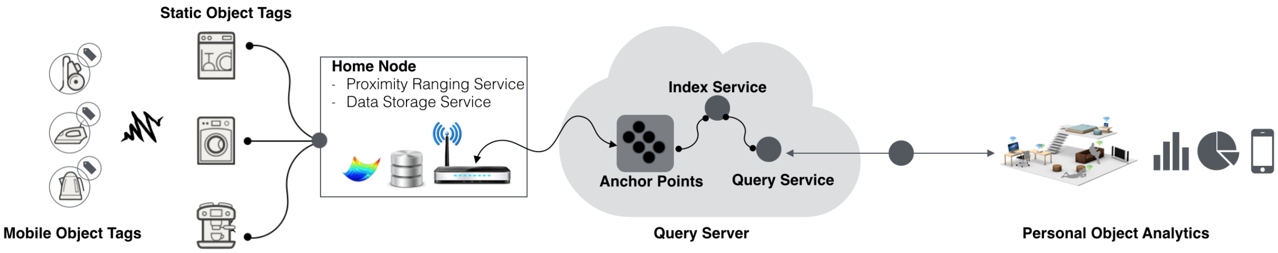


Fig. 1: System Architecture of the Proposed Platform.

connected objects if they do not believe their privacy is well protected. As such, privacy-aware management of personal data (to avoid ubiquitous tracking and profiling) is perhaps the most important challenge for consumer IoT systems. In this work, we advocate for a digital catalogue of personal objects and therefore we have taken system design decisions with very careful assessment of privacy risks. Grounded upon the model of OpenPDS [13], we have opted for a local repository that stores connected objects' location and state-of-use data and puts users at the center of control of their data. To facilitate this, we have designed our system following a cloudlet architecture [21], where the local WiFi gateway acts as a data storage component, runs a proximity ranging algorithm and provides only aggregated views of raw data to external query services. In addition, due to the usage of WiFi management frames for data transport, we also ensure that the data associated with physical objects is geo-fenced and only reaches to nearby gateways running relevant data acquisition components. An advantage of these design schemes is that, it opens up opportunity for willful monetization of personal data as proposed in [12].

III. System Description

The architecture of the platform is illustrated in Fig. 1. Essentially, there are three components that work together to provide the functionalities of the platform, i) *Object Tags* responsible for emitting location and state data of physical objects, ii) *Home Node* responsible for proximity ranging, and local data storage, and iii) *Query Server* responsible for executing application queries on the home node's local storage. In the following, we discuss these components and their implementation details.

A. Object Tags

The purpose of the object tags is to emit the location and state-of-use of the physical objects that are received by the home node. In addition, some object tags also act as location reference points, and receive signals from other tags. Accordingly, there are two types of object tags in our system.

Mobile Object Tags: These tags are attached to physical objects that are mobile in nature, e.g., a coffee cup, a vacuum cleaner, a key, etc. These tags use the built-in accelerometers to detect objects' state-of-use based on their motion signatures, and then emit the state information using WiFi management frames.

Static Object Tags: These tags are attached to stationary physical objects, e.g., a coffee machine, a washing machine, etc., which are often electrically powered. Besides emitting objects' states, there tags can receive signals from mobile object tags and can forward them to the home node through persistent connections.

Object tags perform two important functionalities, i.e., determine the state-of-use, and then emit that state data respectively. For determining the state-of-use, we leverage only accelerometer based motion signature. This is captured by modelling the velocity of physical objects with sinusoidal waveform of acceleration in the 3D space, and is calculated using equation 1.

$$v = \frac{g * \bar{a}}{2\pi f} \tag{1}$$

$$a = \sqrt{x^2 + y^2 + z^2} \tag{2}$$

Here, v is the velocity, g is the gravity, $\bar{a}$ is the average acceleration (calculated with equation 2) over a time window and f is the acceleration waveform over that time window. This movement gives only a coarse grained state-of-use of physical objects (i.e., if motion is detected over a time window the object is considered to be "in-use" otherwise "not-in-use"). However, it is neither complete nor generally applicable to all kind of physical objects. Although additional sensors can be used for detecting more advanced states, we have limited ourselves to this basic motion signature in this work.

The object tags leverage the IEEE 802.11 standard's management frames to propagate the states to the home node. In particular, a Probe Request Frame, also referred as probe, is a WiFi management frame that can be either directed to a specific network, by indicating its SSID, or broadcast to any network within range. Fig. 2 illustrates the structure of a probe request frame. The body of a probe consists of a set of *options*, and each option contains three fields - `Type, Length, Value`. `Type` indicates what the option is about, `Length` indicates the length of the `Value` in bytes. An object tag sends probe requests every time the state of the object changes. In our design, we use a dedicated SSID and a single bit in the probe in that SSID value to reflect the state of the object (1 being "in-use" and 0 being "not-in-use"). One of the side-effects of using probes for data transport is that, it's propagation range is spatially constrained, thus providing an implicit geo-fencing of the data emitted by a physical object

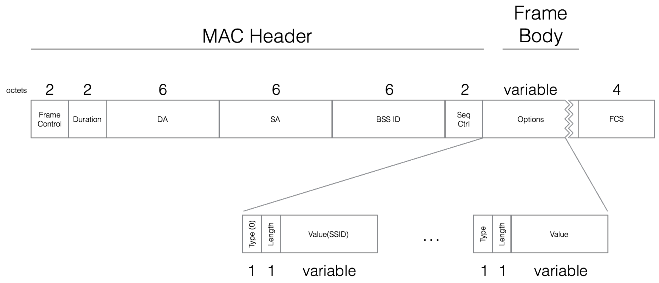


Fig. 2: Frame Format of IEEE 802.11 Probe Request.

The location of an object is determined through the RSSI at the receiving ends (home node and multiple static object tags).The calculation of RSSI is enabled through Radiotap, the de-facto standard for encapsulating 802.11 frames for injection and reception [22]. This encapsulation includes a 8-bit *Antenna Signal* value that provides the RSSI. Home node uses RSSI vectors to determine the proximity of a physical object with respect to one of the stationary objects used as location references.

Static object tags have an additional role in our platform, i.e., they are capable of receiving data from mobile object tags. Every time a static object tag receives a probe from a tag, it forwards the probe packet to the home node. The packet includes the state of the tag, and the RSSI value annotated with the MAC address, timestamp, and a sequence number. The sequence number is represented by 12 bits in the `Sequence Control` field of the probe. Even though the reception time of the same probe signal at various static object tags differ slightly, they contain the same sequence number. The home node uses the sequence number and the timestamp information to assemble a RSSI vector representing single signal received by multiple static object tags. This vector is fed into the proximity ranging service to find the relative location of the object.

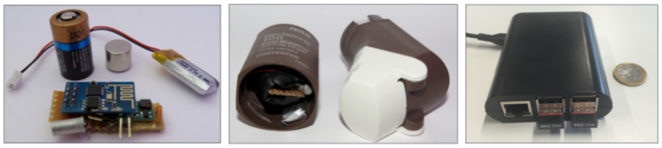


Fig. 3: Object Tags, (a) ESP SoC construction for a Mobile Object Tag, (b) An instrumented Inhaler with a Mobile Object Tag and (c) Raspberry PI 2 Mobel B based Static Object Tag with two WiFi interfaces.

We have implemented the mobile object tag with ESP8266 node [23] as illustrated in Fig. 3(a). This node is a low-cost SoC with integrated WiFi, flash and MCU. It can be programmed directly by flashing code onto the MCU. In our prototype, we used this chip to tag a number of mobile personal objects, such as an asthma inhaler as shown in Fig. 3(b), a note book, a scooter, a vacuum cleaner, a coffee cup, etc. The ESP8266 based nodes have been supplemented with an accelerometer (Freescale MMA8452Q) to support state-of-use determination. The accelerometer is also used to implement a deep sleep strategy that allows the ESP8266 to react to different movement profiles of the tag (e.g. sleeping during quiet periods, waking on impulse or tap movements). This strategy maximises the battery life of the tag.

The static object tags are implemented using a Raspberry PI 2 Model B [24]. The tags are powered and have two identical WiFi interfaces (Ralink 5370), one used for capturing probes coming out of mobile object tags, and the other for maintaining a persistent connection to the home node over websockets. Fig. 3(c) illustrates a prototype of static object tag. The probe capturing functionality is implemented in C++ using the *libtins* library [25]. The connection to the home node is built with Node.js [26]. We used this to tag a number of stationary mobile objects including a coffee machine, a washing machine, a dish washer etc.

B. Home Node

Home node is the heart of our platform. It is designed following a cloudlet architecture [21] in which functionalities such as data storing, processing, etc. are not implemented in a remote server in the cloud but at the edge of the network. The home node hosts two important services. First, a proximity ranging service that assigns a relative location to an object using the network sensed information. Second, a data storage service that stores objects' information locally. We describe these two services next.

Proximity Ranging Service: A proximity algorithm provides a symbolic location relative to a reference point. In our platform, static object tags are used as reference points. When more than one static object tags detect a mobile object, it is considered to be collocated with the one that receives the strongest signal. However, simply selecting the strongest signal does not necessarily yield a reliable ranging performance due to the non-deterministic signal propagation characteristics of WiFi signals. So, we formulate this proximity ranging aspect as a multi-class classification problem by dividing our space in K non-overlapping zones, where each zone is represented by a static object tag with a semantically rich location label, e.g., living room, kitchen, laundry room, hallway etc. Each of these zones has a distinct radio map (RSSI) signature which is constructed by a minimal calibration phase, e.g., placing a tag at different locations within a zone and across zones multiple times and recoding the transmissions signals received at all the zones. After this training phase is completed, our algorithm runs at real-time, i.e., it takes as input the RSSI vector (assembled by combining the RSSI values from the static objects tags that detect a mobile object) and tells us in which zone the mobile object is located in a easily understandable way (e.g., the key is in the hallway.).

Our algorithm relies on Support Vector Machine (SVM) for the classification, and specifically leverages one-vs-one classification scheme [27], [28]. One-vs-one classification uses an underlying binary classifier to compare every pair of classes and followed by a majority voting scheme to decide the winning class.

For the SVM we have used a universal Gaussian kernel:

$$k(\mathbf{x}_i, \mathbf{x}_j) = \exp(-||\mathbf{x}_i - \mathbf{x}_j||^2/2/\sigma^2) \tag{3}$$

where $\mathbf{x}_i$ represents the vector of the RSSI values of a mobile object tag received by the static objects tags.

Although the number of classifiers grows by K^2 with increasing K, the computational complexity of the training and test phase are comparable to other alternatives, e.g., one-vs-all approach [29] or the multiclass SVM [30] that grows linearly with K. This is because each of the $K(K-1)/2$ classifier uses less training data. Other algorithms can replace the SVM as the baseline binary classifier, such as: Random Forests [29]; Gaussian Processes [31]; Boosting [32]; or, Deep Belief Networks [33]. But given the dimensionality of the input space (less than 10) and the amount of training examples (in the hundred-thousand samples), none of the above should provide statistically significant improvements over the SVMs results.

Data Storage Service: This service stores the information about the physical objects. For every signal sent out by an object tag, there is an entry in this storage that comprised of the state and location of the object annotated with time and an object identifier. This local database also contains meta data about every object including a user friendly name, ownership, object type and object description. This service has an external facing interface that connects to the anchor points in the query server (see next). This interface replies to external queries with the information stored in this storage, however, only aggregated data is served. External queries do not have access to the raw data available in this storage.

Home node is implemented using an off-the-shelf WiFi gateway, called Meshlium from Libelium [34]. A Meshlium gateway has all the properties of a 'real' WiFi access point and features a 500MHz x86 processor, two WiFi interfaces (one of which acts as an access point by default) and runs an embedded Debian Linux operating system. Both services are implemented with Node.js and a SQLite database [35] is used to store data.

C. Query Server

This component offers query services to external applications and comprises of the three elements as described below.

Anchor Points: As discussed earlier, all the data about physical objects remains at the home node, i.e., at the edge of the network close to the tracked physical objects. However, to support queries on this data by the external applications, the query server needs an access mechanism to different home nodes. This is achieved by anchor points. An anchor point maintains a persistent connection with a home node using web sockets with failover support. Over this connection, an anchor point forwards queries to a home node and receives the query responses. Anchor points are implemented using Node.js, and run in a virtual server that hosts the the query server components.

Index Service: Our platform uses local storage to maintain data about physical objects in the home node. An external application posts a query to query server, and the query server uses this index service to locate the home node that contains the data relevant to the query. Index service is essentially a key-value store. The keys correspond to the physical objects' identifiers while the values correspond to the home node. When a new object tag (mobile or static) is registered in the system, the home node sends a registration message together with the object identifier, timestamp and ownership information to the anchor point that in turn relays it to this index service. We have used B+ tree to implement this index service [36], [37].

Query Service: This is essentially a query broker for our platform. When a query request is received from an external application, the query service first initiates a short-lived connection with the index service to retrieve the index of the home node and the corresponding anchor point. It then initiates another short-lived connection with the anchor point that eventually forwards the query to the home node. The query may be about a single object, a number of objects or all the objects that are associated with a home node. The query can request the state and location of a physical object (or a set of objects) at a specific time instant, within a time interval (both usage and location traces). The query service is also capable of limiting the number of queries that can be made per unit time by an external application. Once the query service receives the responses from the home node via the anchor point, it forwards them to the external application. This service is stateless and is implemented in virtual servers. This makes it easy to scale the service as required. The Query service is implemented with Node.js.

IV. Prototype Application

In the earlier section, we have described the architectural components of our platform which, as a whole, enables the development of privacy-aware personal object analytics applications. In this section we briefly present a mobile application (developed both for Android 4+ and iOS 8.0+) that showcases the feasibility of building this class of applications with the proposed platform. The experimental settings include 5 static objects and 5 mobile objects instrumented with object tags, and a Meshlium-based home node. The application features a HTML5 front-end and offers the following functionalities.

Seamless Configuration: One of our design challenges is to reduce the complexity of end-user deployment and configuration. This is achieved with our ready-to-run object tags together with this application. Once a user attaches an object tag (mobile or static) to a physical object, the user can use this application to register the object in the system following two simple steps i) search and discover the new tag and ii) provide a friendly name (as illustrated in Fig. 4(b)). This phase assumes that the mobile application is connected to the home node, i.e., a regular WiFi access point. The access point pushes

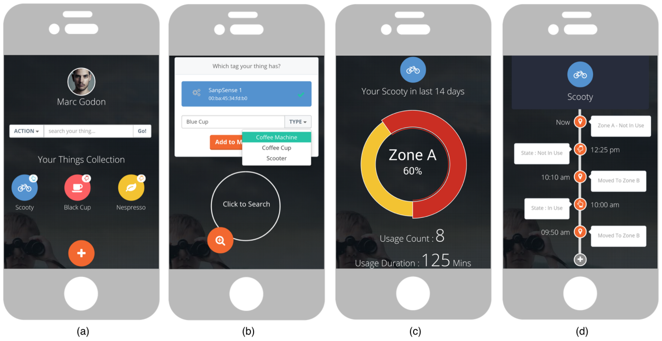


Fig. 4: Prototype Personal Object Analytics App - (a) Dashboard, (b) Registration Phase, (c) Analytics Summary, (d) Usage Timeline.

the new tag to the application for registration. However, after this registration phase is completed, the application does not need to be connected to the home node any longer and can query for the object's status and location through the query server.

Object Dashboard and Summary View: The application offers an object dashboard that shows all the objects that are in the home node as illustrated in Fig. 4(a). A user can pick any object to see its spatio-temporal usage information in an aggregated fashion as shown in Fig. 4(c).

Real Time Search: Finally, a user can search for the real time status of any physical object. The result is presented in a timeline that includes current spatio-temporal usage together with last five entries in the system as depicted in Fig. 4(d).

V. Evaluation

In the earlier sections, we have covered the end-to-end design and implementation of the proposed platform including it's multiple architectural components and models. In this section, we evaluate a subset of these components that we consider are most interesting in the context of this work, i.e., power profile of the WiFi-enabled tag, inference accuracy of the proximity ranging algorithm and computational overhead of the home node. To have a realistic assessment of these aspects we have made some assumptions: *"a regular household has a single home node, 100 connected objects equipped with WiFi-enabled accelerometers and 5 of these objects act as location reference points (augmented with static object tags). There stationary objects are placed at different locations across the household to represent different areas. Each of these connected obejcts can have 10 updates per day for both state-of-use and location changes."*

Although, the number of updates assumed here is relatively small, we argue that most of the household objects are not moved or used on a daily basis. Therefore, this estimation accurately captures the dynamics of a modern home [38].

A. Energy Assessment of the Object Tag

One of the design decisions that we have taken in this work is to use WiFi-enabled sensor tags due to the ubiquity and wider range of the WiFi infrastructure. However, compared to other wireless standards (e.g., Bluetooth) WiFi has relatively higher energy expenditure. Although we have used one of the low-power WiFi chips available today, in this section we evaluate its power profile based on our design decisions (e.g., using probes instead of application processes with deep sleep cycles) and assess its feasibility in practical applications.

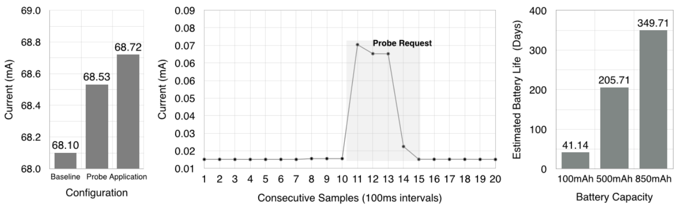


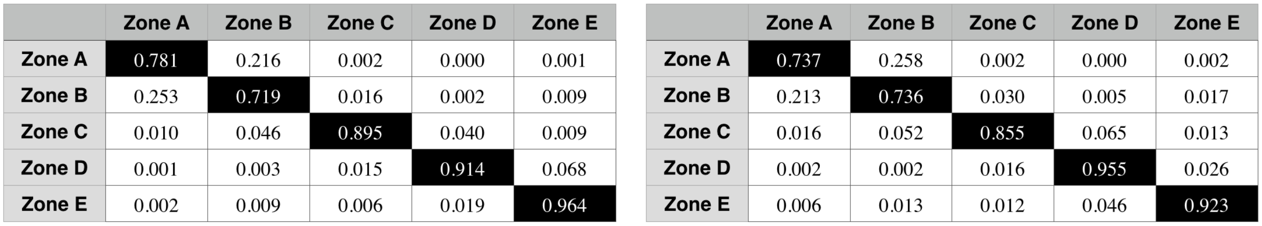
Fig. 5: Energy Assessment of a Mobile Object Tag - (a) Power Profile of Different Test Configurations, (b) Understanding Probe's Behaviour and (c) Estimated Battery Life based on Sensing Loads.

We begin by looking at the power profile with different task configurations, baseline with the various background tasks of the MCU, application process that continuously sends a TCP packet of 49 bytes (the size of the probe frame) every 30 seconds and probe based data transport at the same 30 seconds interval. We used data logging meter (Agilent 34410A) to log the power consumption data every 100ms and each test was run for a period of 30,000 samples (at 100ms intervals). Note that this artificial test configuration is aimed at uncovering the benefits of our design decision of using probe to transport state-of-use data. Fig. 5(a) shows that in relative terms (i.e. when the idle background current consumption is removed) the probe based data transport approach offers approximately 30% improvement in the energy use. To further understand the power profile of a probe, we looked at individual probe's behaviour (depicted in Fig. 5(b)) and observed that a probe exhibits "one peak" - for the outgoing probe request.

Finally, in Fig. 5(c), we show that the battery life based on a 850 mAh battery is estimated to be approximately a year based on a premise that a household object reports on an average 10 updates (state-of-use and location changes) with probes per day. This is based on the published deep sleep consumption of 78uA, a transmit consumption of 70mA, a 80% efficiency of the regulator. The ESP8266 is a very capable platform for WiFi-based sensing. Although the absolute energy gain with probe based transport can be considered small, for applications with long reporting interval the approach is attractive.

B. Inference Peformance of the Proximity Ranging Algorithm

To evaluate the performance of the proximity ranging algorithm, we have divided our office space in five non-overlapping zones (Fig. 6), each represented by a static object tag. For this assessment, we again adopted an artificial setting which was used to gather data and to validate the performance of the classifier. During the data collection phase, we walked around each zone with three tagged mobile objects (an inhaler,



	Zone A	Zone B	Zone C	Zone D	Zone E
Zone A	0.781	0.216	0.002	0.000	0.001
Zone B	0.253	0.719	0.016	0.002	0.009
Zone C	0.010	0.046	0.895	0.040	0.009
Zone D	0.001	0.003	0.015	0.914	0.068
Zone E	0.002	0.009	0.006	0.019	0.964

(a)

	Zone A	Zone B	Zone C	Zone D	Zone E
Zone A	0.737	0.258	0.002	0.000	0.002
Zone B	0.213	0.736	0.030	0.005	0.017
Zone C	0.016	0.052	0.855	0.065	0.013
Zone D	0.002	0.002	0.016	0.955	0.026
Zone E	0.006	0.013	0.012	0.046	0.923

(b)

Fig. 7: Confusion Matrix of the Proximity Classification for the (a) Scooter and (b) Inhaler (b) based on 140K Observations.

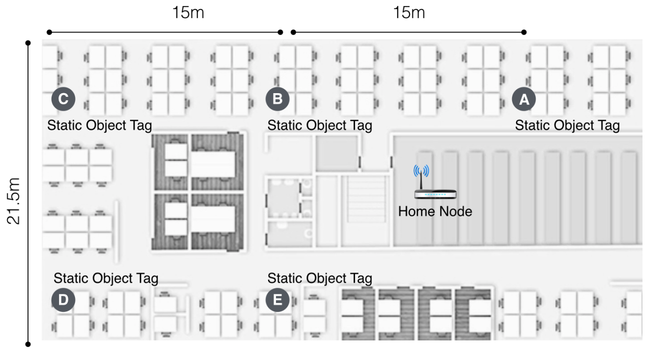


Fig. 6: Experimental Settings for Evaluatuing the Proximity Ranging Algorithm.

a scooter, and a keyring). These tags were shielded in a plastic enclosure. Each of these objects transmitted probes every 100ms for an hour in each zone. We recorded the closet zone manually as ground truth and annotated the records with start and end time of the dwell period in each zone. This phase yielded in a large collection of probe transmission records of 5 hours emitted from three mobile objects and captured by 5 stationary objects. This data is then is passed through a pre-processing phase in which we leveraged the sequence number and the timestamps to create an ordered collection of probes for all transmissions as seen by different static objects.

We used the probes emitted by the keyring to train the SVM classifier, and used the transmissions from the other two objects to test the classifier's performance. To fix the hyper parameters of the SVM, namely σ in equation 3, we have used a validation procedure, in which we used 80% of the data for training and 20% for testing. In Fig. 7, we show the confusion matrices between the five zones, rows represent the estimated zones and columns the true zones. It can be seen that the errors follow the map illustrated in Fig. 6 and the largest errors between Zone A and B are due to the fact that these zones do not have a physical separation as the other three. On an average our algorithm achieved 84% classification accuracy.

C. Space and Communication Overhead in the Home Node

Finally, we provide a mathematical analysis on the computational overhead incurred at the home node due to our design decisions. Based on the assumptions mentioned earlier, i.e., 100 connected objects (n), 5 static objects (m), and 10 status updates (λ) per day, we observe that the overall traffic introduced by our platform in the WiFi network is about 45.5 KB on a daily basis[3]. In addition, every probe from a mobile object tag prompts a maximum of m state update messages from the static objects tags. Every state update includes a timestamp, object id, RSSI measurement, sequence number and state. Timestamp is a 64 bit value in new systems, we use MAC address of the tag to identify the object which takes 48 bits and the state is a one bit value. RSSI takes 8 bits and the sequence number is a 12 bit value. Since the state update is delivered over a persistent connection between the static object tag and home node over Wi-Fi, it includes TCP, IP, 802.11 and Radiotap headers as well. In total, every state update message takes 821 bits. Assuming static object tags can reach the home node in a single hop, the amount of state update overhead in a day mounts up to a little over 481 KB. Note that this traffic is contained in the local network and does not go beyond the home node. Current WiFi routers are more than capable to handle such load in the network.

Each record in the home node database includes a timestamp, ID for the object, state, and location. In our system, a mobile object can be collocated with one of the m static objects. Hence, $\lceil \log_2 m \rceil$ bits are needed to represent the location of an object. Given there are $n\lambda$ in average updates sent by the tags in day, the storage required to store all the updates from object tags is $n\lambda\,(113 + \lceil \log_2 m \rceil)$ bits. With the same numbers as above ($n = 100, m = 5, \lambda = 10$), our system writes 14.16 KB per day to the database. In our prototype, we use Meshlium WiFi gateway as a home node. A Meshlium gateway has approximately 9 GB of available storage. Assuming we can use 5 GB for object storage, our system can seamlessly store data for 370255 days.

VI. Limitations and Future Work

There are a number of limitations of our research presented in this paper. Using WiFi probe as a data transport mechanism is inherently insecure as an adversary can easily snoop on the updates from connected objects. However, currently we do not offer any encryption in our platform. We have shown that avoiding application level communication process can lead to substantial energy gain. However, it is indeed arguable whether probe, a WiFi management protocol, can be subverted

[3] Recall that the size of each probe frame is 49 bytes including the Radiotap header.

in this fashion against its original use. Probe is not the only way to avoid application level processing. We can use other management frames of WiFi standard or the data frame to transport state data, which can then be used for relative location annotation through proximity ranging.

In our current prototype, we have used only one home node assuming the data ownership implicitly belongs to a single household. However, our query server is designed to support connected objects that can move from one home node to other. The index service is able to locate the correct home node for querying when multiple home nodes contain an object's spatio-temporal usage data. However, this requires a clear model and definition of ownership which are currently not supported in our platform.

As a future avenue of our work, we like to address the above limitations, i.e., i) adding an encryption scheme to our transport mechanism while exploring other WiFi frames for data transport and ii) scaling our system to multiple home nodes to support objects' mobility. Furthermore, we aim to deploy our system in multiple households to qualitatively assess the feasibility of our system.

VII. Concluding Remarks

In this paper, we propose a platform that leverages WiFi network to trace the location and state-of-use of connected objects. We highlight a number a design decisions including opportunistic data transport, privacy-aware storage, network-based proximity inference, etc. that provide a foundation for many future IoT applications beyond spatio-temporal analytics of connected objects. Finally, this work makes an important stride by demonstrating how wireless network can be used as a true platform for designing user-centered applications in the IoT space.

References

[1] K.-K. Yap, V. Srinivasan, and M. Motani, "MAX: Human-centric Search of the Physical World," in *The 3rd International Conference on Embedded Networked Sensor Systems (SenSys 2005)*, 2005, pp. 166–179.

[2] C. Frank, P. Bolliger, C. Roduner, and W. Kellerer, "Objects Calling Home: Locating Objects Using Mobile Phones," in *The 5th International Conference on Pervasive Computing (Pervasive 2007)*, 2007, pp. 351–368.

[3] H. Wang, C. Tan, and Q. Li, "Snoogle: A Search Engine for the Physical World," in *The 27th IEEE Conference on Computer Communications (INFOCOM 2008)*, 2008, pp. 13–18.

[4] K. Romer, B. Ostermaier, F. Mattern, M. Fahrmair, and W. Kellerer, "Real-Time Search for Real-World Entities: A Survey," *Proceedings of the IEEE*, vol. 98, no. 11, pp. 1887–1902, Nov 2010.

[5] J. Nickels, P. Knierim, B. Könings, F. Schaub, B. Wiedersheim, S. Musiol, and M. Weber, "Find My Stuff: Supporting Physical Objects Search with Relative Positioning," in *The 2013 ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp 2013)*, 2013, pp. 325–334.

[6] Sticknfind. [Online]. Available: http://www.sticknfind.com

[7] Tile. [Online]. Available: http://www.thetileapp.com

[8] J. Hightower and G. Borriello, "Location systems for ubiquitous computing," *Computer*, vol. 34, no. 8, pp. 57–66, Aug 2001.

[9] H. Liu, H. Darabi, P. Banerjee, and J. Liu, "Survey of Wireless Indoor Positioning Techniques and Systems," *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 37, no. 6, pp. 1067–1080, Nov 2007.

[10] G. Zanca, F. Zorzi, A. Zanella, and M. Zorzi, "Experimental Comparison of RSSI-based Localization Algorithms for Indoor Wireless Sensor Networks," in *The Workshop on Real-world Wireless Sensor Networks (REALWSN 2008)*, 2008, pp. 1–5.

[11] "Wireless lan medium access control (mac) and physical layer (phy) specifications," *IEEE Std 802.11-2012 (Revision of IEEE Std 802.11-2007)*, pp. 1–2793, March 2012.

[12] G. Kortuem and F. Kawsar, "Market-based user innovation in the Internet of Things," in *The 2nd International Conference on the Internet of Things (IoT 2010)*, 2010, pp. 1–8.

[13] Y.-A. de Montjoye, E. Shmueli, S. S. Wang, and A. S. Pentland, "openPDS: Protecting the Privacy of Metadata through SafeAnswers," *PLoS ONE*, vol. 9, no. 7, 2014.

[14] F. Kawsar, "A Document based Framework for User Centric Smart Object Systems," Ph.D. dissertation, Waseda University, 2008.

[15] L. Atzori, A. Iera, and G. Morabito, "The Internet of Things: A Survey," *Computer Networks*, vol. 54, no. 15, pp. 2787–2805, Oct. 2010.

[16] S. Hara and D. Anzai, "Experimental Performance Comparison of RSSI- and TDOA-Based Location Estimation Methods," in *2008 IEEE Vehicular Technology Conference (VTC Spring 2008)*, 2008, pp. 2651–2655.

[17] K. Koile, K. Tollmar, D. Demirdjian, H. E. Shrobe, and T. Darrell, "Activity Zones for Context-Aware Computing," in *The 5th International Conference on Ubiquitous Computing (UbiComp 2003)*, 2003, pp. 90–106.

[18] F. Kawsar, K. Fujinami, and T. Nakajima, "A Lightweight Indoor Location Model for Sentient Artefacts Using Sentient Artefacts," in *The 2007 ACM Symposium on Applied Computing (SAC 2007)*, 2007.

[19] R. Chandra, J. Padhye, L. Ravindranath, and A. Wolman, "Beacon-Stuffing: Wi-Fi without Associations," in *The 8th IEEE Workshop on Mobile Computing Systems and Applications (HotMobile 2007)*, 2007, pp. 53–57.

[20] J. H. Ziegeldorf, O. G. Morchon, and K. Wehrle, "Privacy in the Internet of Things: threats and challenges," *Security and Communication Networks*, vol. 7, no. 12, pp. 2728–2742, 2014.

[21] M. Satyanarayanan, P. Bahl, R. Caceres, and N. Davies, "The Case for VM-Based Cloudlets in Mobile Computing," *IEEE Pervasive Computing*, vol. 8, no. 4, pp. 14–23, 2009.

[22] Radiotap. [Online]. Available: http://www.radiotap.org/

[23] Esp8266. [Online]. Available: http://www.esp8266.com

[24] Raspberry Pi. [Online]. Available: https://www.raspberrypi.org

[25] libtins. [Online]. Available: http://libtins.github.io

[26] Node.js. [Online]. Available: https://nodejs.org

[27] B. Schölkopf and A. Smola, *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond.* Cambridge, MA, USA: MIT Press, 2001.

[28] F. Perez-Cruz and O. Bousquet, "Kernel methods and their potential use in signal processing," *IEEE Signal Processing Magazine*, vol. 21, no. 3, pp. 57–65, 2004.

[29] L. Breiman, "Random Forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[30] J. Arenas-Garcia and F. Perez-Cruz, "Multi-class support vector machines: a new approach," in *2003 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2003)*, vol. 2, 2003, pp. 781–784.

[31] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. MIT Press, 2006.

[32] Y. Freund and R. E. Schapire, "Experiments with a New Boosting Algorithm," in *The Thirteenth International Conference on Machine Learning (ICML 1996)*, 1996, pp. 148–156.

[33] G. Hinton and R. Salakhutdinov, "Reducing the Dimensionality of Data with Neural Networks," *Science*, vol. 313, no. 5786, pp. 504 – 507, 2006.

[34] Libelium. [Online]. Available: http://www.libelium.com

[35] SQLite. [Online]. Available: https://www.sqlite.org

[36] R. Elmasri, G. T. J. Wuu, and Y.-J. Kim, "The Time Index: An Access Structure for Temporal Data," in *The 16th International Conference on Very Large Data Bases (VLDB 1990)*, 1990, pp. 1–12.

[37] C. H. Goh, H. Lu, B.-C. Ooi, and K.-L. Tan, "Indexing temporal data using existing B+ -trees," *Data & Knowledge Engineering*, vol. 18, no. 2, pp. 147 – 165, 1996.

[38] F. Kawsar and A. B. Brush, "Home computing unplugged: Why, where and when people use different connected devices at home," in *The 2013 ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp 2013)*, 2013, pp. 627–636.