IMPROVING LOCALIZATION AND ENERGY EFFICIENCY OF SMARTPHONE APPLICATIONS

Swadhin Pradhan

IMPROVING LOCALIZATION AND ENERGY EFFICIENCY OF SMARTPHONE APPLICATIONS

Thesis submitted in partial fulfillment
of the requirements for the award of the degree

of

Master of Science

by

Swadhin Pradhan

Under the supervision of

Prof. Niloy Ganguly

Department of Computer Science and Engineering

Indian Institute of Technology Kharagpur

January 2015

APPROVAL OF THE VIVA-VOCE BOARD

Certified that the thesis entitled **"Improving Localization and Energy Efficiency of Smartphone Applications"** submitted by **Swadhin Pradhan** to the Indian Institute of Technology, Kharagpur, for the award of the degree of Master of Science has been accepted by the external examiners and that the student has successfully defended the thesis in the viva-voce examination held today.

(Member of the DAC) (Member of the DAC)

(Member of the DAC) (Member of the DAC)

(Supervisor)

(External Examiner) (Chairman)

Date:

CERTIFICATE

This is to certify that the thesis entitled **"Improving Localization and Energy Efficiency of Smartphone Applications"**, submitted by **Swadhin Pradhan** to the Indian Institute of Technology Kharagpur, Kharagpur for the partial fulfillment of the award of the degree of Master of Science, is a record of bona fide research work carried out by him under my supervision and guidance.

The thesis in my opinion, is worthy of consideration for the award of the degree of Master of Science in accordance with the regulations of the Institute. To the best of my knowledge, the results embodied in this thesis have not been submitted to any other University or Institute for the award of any other Degree or Diploma.

Niloy Ganguly
Professor
Department of CSE, IIT Kharagpur

Date:

DECLARATION

I certify that

(a) The work contained in the thesis is original and has been done by myself under the general supervision of my supervisors.

(b) The work has not been submitted to any other Institute for any degree or diploma.

(c) I have followed the guidelines provided by the Institute in writing the thesis.

(d) I have conformed to the norms and guidelines given in the Ethical Code of Conduct of the Institute.

(e) Whenever I have used materials (data, theoretical analysis, and text) from other sources, I have given due credit to them by citing them in the text of the thesis and giving their details in the references.

(f) Whenever I have quoted written materials from other sources, I have put them under quotation marks and given due credit to the sources by citing them and giving required details in the references.

Swadhin Pradhan

ACKNOWLEDGMENTS

.

I sincerely acknowledge Dr. Niloy Ganguly for his guidance and supervision in the development of this thesis. I appreciate his professionalism, planning, and constant involvement in this thesis. He has not only been helpful in the technicalities, such as formulating the problems studied in this thesis and developing solutions to the problems, but also supported me in various walks of my personal life. I would also like to express my sincere gratitude to Dr. Bivas Mitra who has also helped me through various aspects of this thesis. I was fortunate to work with Dr. Romit Roy Choudhury and Dr. Pradipta De without whose knowledge and assistance this study would not have been successful.

I am grateful to the members of my Departmental Academic Committee (DAC): Dr. Rajib Mall, Dr. Arobinda Gupta, Dr. Animesh Mukherjee, and Dr. Sudipta Mohapatra for their suggestions about my work at various occasions. I also express my gratitude to the staff and the other faculty members of the Computer Science and Engineering Department (CSE), for helping me in many academic and non-academic endeavors.

I am grateful to be a member of *CNeRG* research group, IIT Kharagpur where I met some inspirational people. Working in this group was extremely satisfiable and I had a memorable time with Sourav Dandapat, Parantapa Bhattacharya, Suman Maity, Abir De, Saptarshi Ghosh, Sudipta Saha, Rajib Maiti, Surjya Ghosh, Sandipan Sikdar, Tanmoy Chakrabarty, Kaustav Rudra, and Mayank Singh. Special thanks to Sourav-da ('da' refers to elder brother in Bengali) who has been a constant source of joy and motivation, and to Suman who has been immensely helpful . Moreover, I am glad to have Jayanti, Indrasish, Shamit, Abhrajit, Durga-da and Joy-da as friends who helped me in tackling difficult situations and provided me with much needed happiness.

Finally, my deep gratitude goes to my parents. I am very grateful to them for standing by me in everything I have done and giving me whatever they can. They have always provided me continuous support, encouragement and their love. I believe that nothing would be possible without the presence of them in my life.

Swadhin Pradhan
IIT Kharagpur, India.

ABSTRACT

Smartphones are becoming truly ubiquitous devices [33] on which people connect, collaborate, and perform various operations using different context-aware applications. Users' location information is one of the most important requirements of this context awareness. There has been a huge surge of location-aware apps in smartphone app space [56], e.g., Foursquare [7], Google Navigation etc.. Efficient deployment of these location-aware apps requires accurate micro-level identification of locations. In this thesis, we solve this problem by using the concept of *virtual landmarks* [71, 79]. The concept behind virtual landmark is the following. Due to the availability of the embedded sensors (accelerometer, gravity, gyroscope, magnetometer), the smartphones have the ability to recognize the ambience and behavior of users. Consequently, the smartphones can *listen* to the distinguishable environmental signatures to identify a given location. The places might be a corner of a corridor, a GSM blind spot or a specific Wi-Fi zone. In this thesis, we specifically build up a framework to discover such *virtual landmarks* and demonstrate its utility in the development of next generation apps. We rigorously test the stability of these virtual landmarks across different devices, people, and time. As a proof of concept, we develop a prototype system *RetailGuide* using landmarks to facilitate smart retail analytics cum recommendation service.

However, despite the exponential growth of the app market [59], their utility has been restricted by the constraint of limited battery life of smartphones [58]. Cellular radio interfaces on smartphones consume significant amount of energy, aided by growing number of network centric apps. Low utilization of radio resource contributes to higher energy wastage. High speed cellular access links push the bottleneck to the network core risking poor bandwidth utilization of the access link. In addition, small sized packet transmission from different apps leads to waking the interface frequently. In this thesis, we improve the radio usage by aggregating packet transmission from multiple applications. We distinguish between foreground and background apps to introduce differential time delays in transmission such that user experience is minimally impacted. We propose three online packet scheduling techniques, and show the benefits of the methods on different application categories through simulation and real experiments. In summary, the thesis sheds light on the ways to improve localization and cellular resource consumption in smartphone through novel ideas and real world deployments.

Contents

Author's Biography

Swadhin Pradhan received his B.E. degree in Information Technology from Jadavpur Univeristy, Kolkata in 2011. He was awarded University Gold Medal for topping his class. From May 2011 till February 2012, he worked at Interra Systems India Pvt. Ltd., Kolkata, as a Software Engineer. Then he joined Indian Institute of Technology Kharagpur as Junior Research Fellow in a project sponsored by Vodafone. While working in this project, he joined MS (by research) program in the Department of Computer Science & Engineering in July, 2012. His research interests revolved around improving smartphone applications by modifying their localization techniques and network resource usage.

Contributory Publications (Accepted/Communicated)

1 S. Pradhan, A. Balashankar, S. Dandapat, B. Mitra, N. Ganguly, "(Stable) Virtual Landmarks: Enhancing Localization centric Smartphone Applications", communicated to *IEEE Transactions on Mobile Computing*.

2 S. Pradhan, S. Dandapat, B. Mitra, N. Ganguly, P. De "Aggregating Inter-App Traffic to Optimize Cellular Radio Energy Consumption on Smartphones", *International Conference on COMmunication System & NETworkS (COMSNETS)*, 2015, (A poster version has been showcased in *XRCI* (Xerox Research Centre India) Open, 2014).

3 S. Pradhan, A. Balashankar, B. Mitra, N. Ganguly, "(Stable) Virtual Landmarks : Spatial Dropbox to enhance Retail Experience", *International Conference on COMmunication System & NETworkS (COMSNETS)*, 2014.

List of Figures

List of Tables

Chapter 1

Introduction

Mobile devices such as smartphones provide significant convenience and capability to the users. A recent market analysis [33] shows that the smartphone market is the fastest growing segment of the mobile phone market; by the end of 2013, 6% of the global population own a tablet, 20% own a PC, and 22% own a smartphone, i.e., one in every five people in the world now carries a smartphone [61]. Smartphone runs OS which provides full-fledged app development platform, and coupled with exotic components such as Camera and GPS, have unleashed the imagination of app developers. According to a new report [59], the app market will explode exponentially to a 38 billion industry by 2015, riding the huge growth in popularity of smartphones. Moreover, most of these smartphone apps differ distinctively from their desktop counterparts, by being more context-aware. Hence, there has been a huge surge of these context-aware apps in smartphone app space [56], e.g., *Foursquare*, *Google Navigation*, *Noom* etc.. Particularly we find that, users' accurate location information is one of the most important features of this context awareness.

In spite of the increasing popularity of these kinds of location-centric apps, location services currently provided by the state-of-the-art smartphones are not quite good enough both in terms of accuracy and energy usage. Different GPS aided techniques [75, 80] have partly solved the problem, but the energy cost is high. Most importantly, for the case of indoor localization, it fails miserably [43, 66, 34, 65]. There have been different infrastructure based efforts [32, 73, 81] to solve the problem, but again extra expenditure for these kinds of deployments actually inhibit wide-spread adoption [74]. However, current generation of smartphones are equipped with different kinds of sensors like gyroscope, accelerometer, magnetometer etc. These sensors can listen to different sensory signature of the environment and can track users' movement via dead-reckoning with a very low energy footprint [19, 79]. If we can map these signatures to the corresponding locations, then these can act as virtual landmarks understood by the smartphones. However, the heterogeneity of different smartphone hardware and softwares can not guarantee the identification of these kinds of landmarks across different settings. To be universally effective, such *virtual landmarks* need to exist across various devices as well as people using those devices. This unique sensor-aided technique of location tracking by building smartphone sensor based virtual landmark database can benefit both indoor and outdoor localization.

On the other hand, despite the incredible market penetration of smartphones and exponential growth of the app market, their utility again has been restricted by the constraint of limited battery life. As such, optimizing the energy consumption of millions of smartphone apps is of critical importance. However, the quarter million apps developed so far are largely developed in an energy oblivious manner. Poorly written apps can sap 30% to 40% of a phone's battery [58]. Battery lifetime is a common cause of frustration among smartphone users. However, smartphone data traffic has been growing steadily due to the increasing popularity of cloud based applications [48]. As the adoption of cloud based services grows, the data traffic from smartphones will keep rising. A majority of these cloud based apps [48], like mailing, online storage, or social networking apps, run as background services, whereby the application thread wakes up intermittently to synchronize with the server.

Unnecessary long stay of cellular radio at high energy state is one of the main reasons behind this significant energy depletion in smartphones [58]. This happens because the different background services run continuously and frequently wake up the network card [77] leading to high energy usage. As the different services are independent of each other, they are generally not in-sync and underutilizes the available network bandwidth by wasting ramp-up energy[1] and tail

[1]Ramp-up energy is the energy cost for radio to jump to a high-power state from idle state.

energy[2] [20]. Interestingly, it is observed [77] that, most of the cases network radio goes into the highest energy state to serve only one or two apps and thus a considerable amount of energy is wasted. Reducing tail time energy wastage has been addressed primarily by (a) dynamic adjustment of the tail time timer by observing traffic patterns [51, 70], and (b) using the tail time for transmissions [20, 55]. In order to fill the tail time with transmissions, the classical approach is to aggregate packets from a single application either by delaying packets [21], or reorganizing computation and communication leading to higher batch efficiency [82, 87].

However, if we can aggregate different network requests across multiple applications intelligently, we can significantly save the duration of cellular radio at high energy state, without losing significant ramp up or tail energy due to frequent switching of states. This methodology of saving energy is hinted by Falaki et. al. [40] and partially tackled by Yong Cui et. al. [84] in variable wireless bandwidth settings. The challenge is that in this process of energy saving via optimizing cellular radio usage, we should not degrade the user experience of the apps. Higher packet aggregation across applications requires that requests from different apps may need to be delayed to synchronize the transmissions. The delay limit for an app is calculated based on the user's interaction with the app, i.e. a measure of user experience. Proposing different delay-aware algorithms to address the trade-off of energy saving vis-a-vis user experience, is non-trivial.

Hence, if we want to look at our solutions for improving efficiency of smartphone applications at a broader perspective, we are actually trying to improve the smartphone performance from two different ways (1) application level, and (2) kernel level. Application level efficiency is achieved through identifying smartphone sensor based landmarks and kernel level efficiency is achieved through devising smart network packet scheduler which will reside in a middle-ware.

[2]Tail energy is the energy wastage incurred during the period of time, i.e. tail time, when radio remains in high-power state after a communication session has ended.

1.1 Differences with other Related Works

The idea of landmark for navigation or localization is pretty old [29, 52, 78]. Due to low accuracy (∼10m) in some outdoor scenarios [6], unavailability in indoor scenarios and high power consumption, GPS is far from the ideal. Interestingly, this motivated researchers to revisit the idea of landmarks for localization. We can find the essence of landmarks in recent localization works which are based on ambience signature. Some of the recent localization or place recognition systems have been EZ localization [32], GSM signal fingerprinting [16], Surroundsense [19], RF based techniques [81] and Wi-Fi based schemes [73]. Moreover, a few works augment urban dead-reckoning [85] to improve indoor localization using mechanical sensors like accelerometer and gyroscope. Although distinguishing signature is the core of any landmark, landmark can be more than a vector of signatures. This idea of landmark for simultaneously localizing object is first explored by robotics community through the works of SLAM [38]. However, they are concerned about finding visible landmarks through costly sensors. Their goal was appeased easily as the mechanical movement of robots help them to do precise dead reckoning. However, the concept of invisible landmarks through the cheap smartphone sensors are brought forward by the authors of UnLoc [79].

They, like us, use different sensor signature to form landmarks which provide regular check points and helps in reducing the localization error. But, they are only confined to the localization problem for their experiments and also silent about the impact of heterogeneity on this kind of system. This work has broadened the horizon by exploring different interesting implementation avenues like retail, and showed through a set of experiments that we can find a set of stable landmarks in spite of the heterogeneity. We have also developed prototype apps to demonstrate the usability of stable virtual landmarks.

On the other hand, conserving cellular network card energy has focused primarily on two techniques - tail time adaptation, and traffic aggregation. The design of Tail Optimization Protocol (TOP) by Qian et al. [70] leverages the Fast Dormancy feature and usage pattern to predict long idle periods when the card can be immediately switched off, thereby eliminating any tail time wastage [70]. A further generalized approach for predicting network idle time is proposed by Kim et al. [51]. Further optimization are introduced by better utilization of cellular bandwidth in Bartendr [72] and LoadSense [30], where free channel time and gateway server load is used to trigger transmissions. However, a series of work, starting with Tailender, has observed that utilizing the low bandwidth channel during tail time for transmission can significantly save energy. Tailender showed that by clubbing traffic one can reduce energy by 35% for email applications, 52% for news feeds and 40% for web search [20]. Techniques for batching periodic jobs [68] are proposed in [27]. TailTheft [55] uses the tail time for pre-fetching and delayed transfers, showing the energy benefits if we adjust the tail timer according to application behavior.

Going beyond single app scheduling, in this work, we show that significant energy savings can be achieved by scheduling packets across apps. For a multi-tasking smartphone user this is a natural usage behavior. We also show that even in presence of small packet sizes, significant energy savings can be gained. Our work also takes into account the typical multi-tasking behavior of smartphone users, and considers the method to ensure user satisfaction by minimally disrupting foreground job.

1.2 Objectives of the thesis

The objective of the thesis is to propose two different work items related to improving the location accuracy and energy usage of smartphones respectively. This will in turn improve the efficiency of current smartphone applications, by making them efficient in terms of context-awareness and energy saving.

- Efficient Localization in Smartphones: The first study shows that localization can be improved using different embedded sensors in smartphones which can detect environmental signatures. We describe a methodology to detect smartphone sensor based virtual landmarks using adaptive clustering algorithm and validate its accuracy in the indoor and the outdoor scenario.
- Energy Efficient Cellular Radio Usage in Smartphones: The second problem focuses on solving the problem of energy drain in cellular radio due

to running background processes, by intelligent scheduling of different concurrent resource requests. Our goal here is to maximize the radio resource utilization, which in turn will reduce tail energy wastage, by aggregating packets across multiple apps in a smartphone, without hampering the user experience.

- Developing prototypes as proof of concepts : To validate the efficacy of virtual landmarks in localization, we have developed a retail app called *RetailGuide* and a virtual sign creator app called *SignFinder*. On the other hand, we have also collected network usage traces in smartphone to show that our algorithms provide satisfactory energy savings in the real world.

1.3 Contributions of the thesis

Considering the broad objectives stated in the previous section, the particular contributions of the thesis are detailed as follows.

(1) Improving localization efficiency in smartphone apps via virtual landmarks

Landmarks are made of distinguishable unique features, which differentiate themselves from the immediate surroundings. Recent research works reveal that like us humans, it is actually possible by the sensors on smartphones to identify landmarks [71, 79]. This can create the possibility of applications in the areas of location based social networks, augmented reality, gaming, retail etc. However, to make such applications a reality, a particular landmark need to be stable across mobile phones as well as users carrying those mobile phones etc.

We specifically build up a framework to discover such stable landmarks and demonstrate its utility in the development of next generation apps. In order to identify such virtual landmarks, we employ a clustering algorithm to perform non-intuitive feature combination of sensors like Accelerometer, Gyroscope, Magnetometer, Light, Sound, Wi-Fi, GSM signal strength etc. We describe a methodology to detect these virtual landmarks using adaptive clustering algorithm which is similar to the scheme proposed in [79] and validate its accuracy in both indoor and outdoor scenarios. We perform extensive experimental studies with our developed android applications on Samsung smartphones to understand the internal dynamics of the stability of these virtual landmarks and its dependence on different parameters like devices, time, and persons.

As a proof of concept, we developed a prototype system *RetailGuide* using landmarks to facilitate smart retail analytics cum recommendation service. We also did detailed experiments in the corridors of department building, a outdoor market area, and a nearby retail store, and showed that *RetailGuide* app works with high accuracy and robustness, both indoor and outdoor. Moreover, to demonstrate the importance of this concept of landmarks in smartphone app space, we have shown two use cases in automatic creation of *virtual signboards* for giving directions to locations of interest and localization in post disaster situations for rapid rescue operations.

(2) Reducing cellular radio energy consumption via aggregation of network packets across smartphone applications

Cellular radio interfaces on smartphones consume significant amount of energy. Growing number of network centric apps has led to active research looking for energy efficient solutions in managing the radio resource. Low utilization of radio resource contributes to higher energy wastage. High speed cellular access links push the bottleneck to the network core risking poor bandwidth utilization of the access link. In addition, small sized packet transmission from different apps leads to frequent activation of the interface. Here, we improve the radio usage by aggregating packet transmission from multiple applications. We distinguish between foreground and background apps to introduce differential time delays in transmission such that the user experience is minimally impacted. We propose three online packet scheduling techniques, and show the impact of the methods on different application categories. Simulation driven experiments using synthetic trace show energy gain of 40% over earlier works, while experiments using trace data gathered from real usage show 10% improvement.

1.4 Organization of the thesis

The rest of the thesis is organized as follows:

In **Chapter 2**, we present a detailed literature survey of the related works on the aspects of localization and energy in smartphone applications.

Chapter 3 points out the problem of localization in smartphone applications and discusses the concept of smartphone sensor based landmarks to solve this issue. We also elaborate the details of localization centric android applications, namely, *RetailGuide* and *SignFinder*, using this concept of virtual landmarks, which are developed and thoroughly tested in different scenarios by us.

Chapter 4 investigates the problem of energy wastage in cellular radio of smartphone applications and proposes a practical solution to enable aggregation of network packets across multiple applications to counter the problem.

Finally, **Chapter 5** concludes the thesis by summarizing the contributions and indicating a few issues for future work that have been opened up from the studies in this thesis.

Chapter 2

Literature Survey

As reported in the previous chapter, the main contribution of this thesis is to improve the localization techniques and cellular resource consumption techniques of current smartphone applications. In this chapter, we present a comprehensive survey of the various works done in the field of localization using smartphones and cellular radio energy reduction strategies. The organization of the survey is as follows. First, we state several limitations affecting the current generation of smartphone applications. Specifically, we try to highlight the problems of inefficient localization and cellular energy wastage. We then briefly review different kinds of state-of-the-art localization strategies used in smartphones with their corresponding benefits and limitations. Following this, we present a brief description of cellular radio energy model and finally arrive at a detailed description of different network energy reduction strategies adopted for smartphone applications. Moreover, apart from focusing on the related studies, we also try to put forward the intuition and implication of our proposed solutions towards improving the performance of current smartphone applications.

2.1 Problems in Smartphone Applications

The smartphone is to the 2010s as the Internet was to the 1990s and 2000s. The increasing popularity of smartphones with their sensing capability and the availability of *application distribution* channels, such as, the Apple AppStore [3] and the Google Play [10], is giving both researchers and developers a unique opportunity to deploy mobile applications at an unprecedented scale. As of July 2013, the Google Play store has officially published over one million apps and there has been over 50 billion downloads [11]. So, smartphone app industry is growing rapidly and expected to become a 40 billion dollar industry by 2015. However, there are certain roadblocks which need to be tackled to attain such growth. In the following, we are enumerating a few of these.

1. Energy: Battery life of current generation of smartphones is one of pressing problems limiting its wide-spread adoption [58]. Even a smartphone having 1500 mAh battery can last only upto 6 hours [63] if used continuously. Surprisingly, thousand of apps currently in the app market space do not take care of this issue and can contribute for energy loss upto 40% [63]. Recent surveys [13] show that users are very conscious about the energy usage of a particular app and several battery saver apps like *Battery Doctor*, *DU Battery Saver* etc. are helping them. So, energy footprint of any app has become a good indicator of the popularity of an app. Therefore, both researchers and developers are trying to develop many tools and strategies [17, 58] to address the energy problem in smartphones.

2. Privacy and Security: In this age of identity thefts [8] and surveillance [12], privacy and security concerns are gaining importance. Users invariably leak many sensitive information to the third party apps and sometimes a few rogue apps [5] take advantage of that. However, most of the times apps need coarse information of users' activity to provide more personalized experience, e.g., a restaurant finder app gives location and interest specific recommendations or a music app automatically selects songs according to the *mood*. So, there is a trade-off between misuse leading to identity theft and proper usage for improving user experience. Researchers [15, 44] have attempted to strike this balance, but no single solution is able to catch the imagination of all.

3. Location Sensing: Context awareness is the main reason for the transformation of a mobile phone into a *smart*phone. Apart from the user's activities in smartphone, location of the user also plays important part for defining the context. So, smartphone makers or smartphone OS developers always have given importance to location sensing. GPS and cellular tower triangulation are the traditional solutions. But, due to huge energy drain coupled with absence of required accuracy [28] in these methods, location sensing has become a major problem. Most importantly, as indoor localization is becoming an indispensable part of current generation of smartphone applications, the need for alternative lightweight solution has become more urgent.

4. Scalability and Adaptability : Smartphone applications are also troubled by traditional issues like scalability and adaptability. The problem is more acute due to resource shortage and huge heterogeneity in OS or screen sizes like never before [24]. Recent breakdowns in popular apps like *WhatsApp* [14] or *BBM* [4], have actually established this issue. Both researchers and developers are trying to find different solutions for this pressing problem of smartphone applications.

Apart from the above issues, there are also many more problems affecting current generation of smartphone applications. However, in this thesis, the focus is mainly on the problems of location sensing and energy in smartphone applications. Moreover, we mainly tackle the problem of energy wastage due to network usage, which is the second most important reason of energy depletion in smartphones [28]. In the following sections, we will describe different related studies in these fields.

2.2 Different Localization Techniques

After the transformation of cellphone to a smartphone, a phone has become more than a mere calling device. Locating oneself in unknown places or finding route to a specific destination has been one of the important needs of the users. For this reason, smartphone based localization has caught the attention of both developers and researchers from very early days. Following are the two main classes of localization techniques in today's smartphone.

1. Outdoor Localization Techniques : For navigation purposes in unknown places, GPS chip was introduced in the smartphone. GPS uses signals from four revolving satellites to triangulate the location and works reasonably well in general outdoor situations [6]. However, it suffers terribly in dense building area due to multi-path effect [6] and causes almost 20m error in localization. Moreover, in smartphones, it also suffers from signal receiving problem and has very high energy footprint [28]. Due to these limitations, cellular or wireless tower based triangulation [83] has become popular. Although it has lesser energy footprint than GPS, this technique suffers from an error around 100m. However, in some other cases, a smartphone OS gives options to users to choose among these two techniques keeping in mind their requirement [2]. Moreover, researchers also tried different avenues to reduce the usage of GPS for localization through assisted GPS based system [80], where GPS readings are taken periodically without degrading the accuracy significantly. Some people also attempted to propose a collaborative system where neighboring users transfer GPS information among them for localization to reduce redundant queries and in turn the overall energy consumption [75]. In assisted GPS mechanism, researchers also proposed the use of embedded sensors like accelerometer or gyroscope for finding direction or path traversed by the user which helps in localization [19]. However, these types of collaborative or assisted methods although look good in theoretical perspective but fail in practical scenarios due to unavailability of peers or good trade-off metric of performance and energy.

2. Indoor Localization Techniques : Due to unavailability of proper signals of GPS satellites or cell tower, in indoor scenario most of the discussed outdoor localization techniques fail miserably [6]. In general, some of the methods take help of infrastructure or peculiar signal variation to localize people indoors. Both researchers and developers have tried to propose different methods of indoor localization which will help different applications. For example, EZ localization [32] scheme uses Wi-Fi RSSI to efficiently localize, whereas [67, 81] utilize location beacons and receivers for indoor localization; [16] uses GSM signal or [31] uses FM radio signal for fingerprinting and locating users. RF based or Wi-Fi or FM based schemes either suffer from infrastructure dependence or high calibration time, while localizing people with corresponding places. Moreover, researchers have tried to propose ambiance signature based indoor localization techniques. Surroundsense [19] uses the characteristics of places to identify location. A few works augment urban dead-reckoning [85] to improve indoor localization using mechanical sensors like accelerometer and gyroscope. These systems although depend upon the signature of surroundings, they do not explicitly bring the concept of landmark, which is the main idea of our solution.

In this thesis, although we mainly concentrate on smartphone based human localization, it is necessary to briefly introduce the idea of Simultaneous Localization and Mapping (SLAM) idea in this context. The necessity stems from the issue that our idea of landmark based localization is certainly similar to SLAM but completely different. SLAM is a highly popular and successful technique in robotics, which allows a robot to simultaneously discover landmarks and build a map-representation of an indoor environment [60]. However, SLAM typically depends on using explicit environment sensors, such as laser range finders and cameras. Moreover, the rotation of the robot wheels offer a precise computation of displacement [39]. To localize and create maps, statistical techniques are used to approximate using Kalman filters, particle filters and scan matching of range data. They provide an estimation of the posterior probability function for the pose of the robot and for the parameters of the map [49, 50]. Unlike SLAM, our idea uses smartphone sensors to compute the displacement and direction of users; the landmarks are essentially ambient signatures or user-activities.

2.3 Landmarks and their use in Localization

The idea of landmark for navigation or localization is pretty ancient. From the pole star guiding the sailors to helping out today's busy teens to find the common meeting place, landmarks have always been an integral part of our daily life. Moreover, migratory birds find their winter abode [78], desert ants find their food [26], or honey bee tracks back their way back to home [29] using spatio-temporal landmarks. Even human minds keep track of some routes or places in terms of landmark maps [52]. We combine this idea of landmark of the environmental signatures that can be identified by the embedded sensors of smartphone, which we call as *virtual landmarks*. The idea of virtual landmark has been introduced [79], where a sensory signature cluster of a particular location can be identified by smartphone sensors. The concept behind virtual landmark is the following. Due to the availability of the embedded sensors (accelerometer, gravity, gyroscope, magnetometer), the smartphones have the ability to recognize the ambiance and behavior of users. Consequently, the smartphones can listen to the distinguishable environmental signatures to identify a given location. The places might be a corner of a corridor, a GSM blind spot or a specific Wi-Fi zone.

Although distinguishing signature is the core of any landmark, landmark can be more than a vector of signatures. This idea of landmark for simultaneously localizing object is first explored by robotics community through the works of SLAM [38]. However, they are concerned about finding visible landmarks through costly sensors. Their goal was achieved easily as the mechanical movement of robots help them to do precise dead reckoning. In this work, we use different sensor signature to form landmarks to provide regular location fixes. We are not confined to indoor localization for our experiments, and have also considered the impact of heterogeneity on this kind of system. Our work has broadened the horizon of applications of localization by exploring different interesting implementation avenues like retail, and showed through a set of experiments that we can find a set of stable landmarks in spite of the heterogeneity.

Next, before discussing the works related to efforts directed towards smartphone energy usage reduction, we present a background of cellular architecture and energy model of smartphones.

2.4 Background of Cellular Systems

$3G$ cellular data networks have recently witnessed a rapid growth, especially due to the emergence of smartphones. In this work, we focus on the UMTS (the Universal Mobile Telecommunications System) 3G network, which is among the most popular 3G mobile communication technologies [1]. Compared to Wi-Fi, $3G$ systems operate under more radio resource constraints. To efficiently utilize the limited radio resources, UMTS introduces for each *user equipment* (UE, i.e., a smartphone) a *radio resource control* (RRC) state machine that determines radio resource usage affecting device energy consumption and user experience. Following are the brief descriptions of $3G$ $UMTS$ architecture and $3G$ RRC energy model.

2.4.1 UMTS Architecture

As illustrated in Figure 2.1, the UMTS network consists of three subsystems: User Equipments (UE), UMTS Terrestrial Radio Access Network (UTRAN), and the Core Network(CN). UEs are essentially smartphones carried by the users. The UTRAN provides connectivity between a UE and the CN. It consists of two components: base stations, called Node-Bs, and Radio Network Controllers (RNC), which control multiple Node-Bs. Most UTRAN features such as packet scheduling, radio resource control, and handover control are implemented at the RNC. The centralized CN is the backbone of the cellular network. In particular the GGSN (Gateway GPRS Support Node) within the CN serves as a gateway hiding UMTS internal infrastructures from the external network.

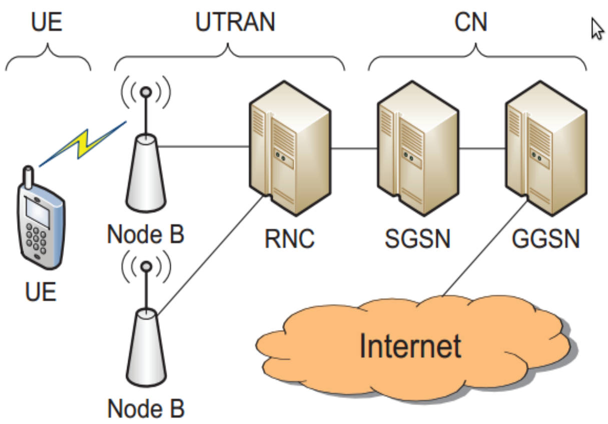


Figure 2.1: The UMTS Architecture

2.4.2 UMTS $3G$ Network Model

In the context of UMTS, the radio resource refers to WCDMA codes that are potential bottleneck resources of the network. To efficiently utilize the limited radio resources, the UMTS radio resource control (RRC) protocol introduces a state machine associated with each UE. There are typically three RRC states as described below and as shown in Figure 2.2.

IDLE. This is the default state when a UE is turned on. The UE has not yet established an RRC connection with the RNC, thus no radio resource is allocated, and the UE cannot transfer any user data (as opposed to control data). So, in this state, no bandwidth is allocated as well as no power is used.

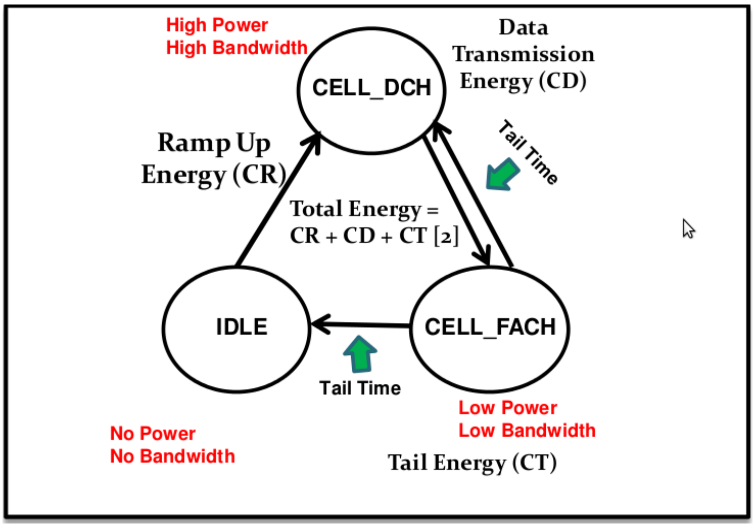


Figure 2.2: 3G Energy State Diagram

CELL_DCH. The RRC connection is established and a UE is usually allocated dedicated transport channels in both downlink (DL, RNC to UE) and uplink (UL, UE to RNC) direction. This state allows a UE to fully utilize radio resources for user data transmission. A UE can access HSDPA/HSUPA (High Speed Downlink/Uplink Packet Access) mode, if supported by the infrastructure, at $CELL\_DCH$ state. $CELL\_DCH$ is the high throughput state where packets are transmitted and it consumes around 800 mW power, as shown in Figure 2.2.

CELL_FACH. The RRC connection is established but there is no dedicated channel allocated to a UE. Instead, the UE can only transmit user data through shared low-speed channels that are typically less than 15kbps. $CELL\_FACH$ consumes much less radio resources than $CELL\_DCH$ does. $CELL\_FACH$ consumes around 460 mW power for its operation and requires low data throughput.

In the 3G state machine, there are two types of state transitions. State promotions and State demotions. State promotions are $IDLE \rightarrow CELL\_DCH$, $CELL\_FACH \rightarrow CELL\_DCH$, and $IDLE \rightarrow CELL\_FACH$ where switching is done from a state with lower radio resource to higher radio resource. However, State demotions are $CELL\_DCH \rightarrow IDLE$, $CELL\_DCH \rightarrow CELL\_FACH$, and $CELL\_FACH \rightarrow IDLE$ which works in the reverse direction. The energy consumed in these state transitions are directly proportional to the energy difference between state, e.g., $CELL\_FACH \rightarrow IDLE$ is energy wise cheaper than $CELL\_DCH \rightarrow IDLE$ transition.

2.4.3 UMTS $3G$ Energy Model

We model the energy consumption using the UMTS three-state ($CELL\_DCH$, $CELL\_FACH$, $IDLE$) 3G network card operating model and corresponding energy values for state transitions [18] illustrated in the Figure 2.2. For calculating energy usage in 3G card, we consider the equation, Total Energy Consumption = $C_R + C_D + C_T$, where C_R is the ramp up energy ($IDLE$ to $CELL\_DCH$), C_D is the data transmission energy, and C_T is the tail energy (in $CELL\_FACH$) [18]. C_D consists of two parts, namely actual data transmission energy (C_B) and data-transferring state (in $CELL\_DCH$) maintenance energy (C_M). From [20, 69], we have taken the corresponding values: $C_R = 3.5$ J, $C_T = 0.62t$ J, $C_B = 0.25x$ J, and $C_M = 0.8$ J, where x denotes the amount of Kilobyte transferred and t is the time spent at $CELL\_FACH$ state due to tail time.

2.5 Reduction of Cellular Radio Energy

To reduce overall energy consumption in smartphones, energy wastage in cellular connectivity is an important step [28]. Prior works on conserving $3G$ network card energy have focused primarily on two techniques - tail time minimization, and traffic aggregation. Following are the detailed description of the strategies.

Radio Tail Time Minimization Strategies: The design of Tail Optimization Protocol (TOP) by Qian et al. [70] leverages the Fast Dormancy feature[1] and usage pattern to predict long idle periods when the card can be immediately switched off, thereby eliminating any tail time wastage [70]. A further generalized approach for predicting network idle time is proposed by Kim et al. [51].

[1]In UMTS, *Fast Dormancy* is a mechanism for a handset to notify the cellular radio for immediate radio resource release.

RadioJockey analyzes program behavior to determine communication spurts to enable faster switch off [17]. Further optimization are introduced by better utilization of cellular bandwidth in Bartendr [72] and LoadSense [30], where free channel time and gateway server load is used to trigger transmissions.

Traffic Aggregation Strategies: A series of work, starting with Tailender, has observed that utilizing the low bandwidth channel during tail time for transmission can significantly save energy. Tailender showed that by clubbing traffic one can reduce energy by 35% for email applications, 52% for news feeds and 40% for web search [20]. Xu et al. focuses on the behavior of email applications on smartphones, and proposes techniques to reduce energy cost of email sync by 50% [82]. Background jobs on smartphones lead to unnecessary wakeup of 3G radio, and often these jobs are periodic in nature [68]. Techniques for batching periodic jobs are proposed in [27]. TailTheft uses the tail time for pre-fetching and delayed transfers, showing the benefits on per application behavior [55]. Going beyond single app scheduling, in this work we show that significant energy savings can be derived by scheduling packets across apps.

Catnap system is proposed as a solution for better utilization of high speed access link in the context of high speed Wi-Fi, and slow broadband Cable/DSL [37]. In principle, Catnap is closest to our approach. Although Catnap concludes that efficacy of their approach is limited to applications with large packets, we show in this work that even if the packet sizes are small, significant energy savings can be made. Our work also takes into account the typical multi-tasking behavior of smartphone users, and considers methods to ensure user satisfaction by minimally disrupting foreground job.

2.6 Summary

In this survey, in Section 2.1 we have pointed out that several problems are limiting the adoption of smartphones. As the problem space is too huge to tackle in a single thesis, we have attempted to address the two pressing issues of location sensing and energy wastage in smartphone applications. In Section 2.2, we describe in detail different localization techniques and their shortcomings.

Later, in Section 2.3, we have provided the essence of our solution which will get more attention in next chapter. In the following Section 2.4, we have illustrated the background information of cellular systems which will help to understand the problem and solution more clearly. Finally, in Section 2.5, we describe different proposed strategies to reduce cellular energy consumption and later pointed out the main intuition of our solution.

Chapter 3

Localization using Virtual Landmarks

3.1 Introduction

Smartphones are becoming truly ubiquitous devices [33] on which people connect, collaborate, and perform various operations. So, smartphone app centric approach to provide personalized experience to users has gained immense popularity. People now-a-days rarely travel around a new city without using *Google Maps* or enter into a shopping mall without checking in *FourSquare* or *Facebook*. An important requirement for a large class of app to function smoothly is to understand the location of the mobile phone accurately. Hence, there is a need for developing accurate micro-level localization schemes.

One elegant way to perform micro level localization is to introduce *virtual landmarks* [79, 71], which a sensory signature cluster of a particular location that can be identified by smartphone sensors. The concept behind virtual landmark is the following. Due to the availability of the embedded sensors (accelerometer, gravity, gyroscope, magnetometer), the smartphones have the ability to recognize the ambience and behavior of users. Consequently, the smartphones can *listen* to the distinguishable environmental signatures to identify a given location. The places might be a corner of a corridor, a GSM blind spot or a specific Wi-Fi zone.

In this chapter, we propose a thorough virtual landmark enumeration procedure via clustering sensor data and evaluate our algorithm in both indoor and outdoor space. It is important to note that, in order to effectively use them as virtual information area, these landmarks need to be stable - that is, the the mobile phone based landmark signatures need to be trustworthy (a) across different smartphones, manufactured by different vendors (b) across different users. We carried a detail test of stability of landmarks across different factors and discovered interesting insights like device hardware specific heterogeneity mostly affects the stability.

Figure 3.1 shows a concept image of a shopping area annotated with virtual landmarks. The figure shows different landmarks at different places of the shopping area, e.g. a magnetometer landmark nearby mobile section or a sound landmark near customer care. A few of the landmarks may be overlapped like Wi-Fi landmark (denoting a specific Wi-Fi zone) and Gyroscope - Accelerometer landmark near one book section.

An important use case of virtual landmark can be to track shoppers' activity. As there is very less retail analytics application in the market, we can use these virtual landmarks to track shoppers' activity. So, as an use case of virtual landmarks, we can have a retail analytics app. Following this vision, we have developed a prototype *RetailGuide* app. *RetailGuide* app identifies and then utilizes these virtual landmarks as dropboxes distributed in physical space, where users can drop their comments about something nearby. To further show usability of virtual landmarks, we have built a prototype *SignFinder* app to identify real world signs like toilet or office dynamically, using the virtual landmarks. Moreover, we have described different use cases of virtual landmarks in preventing location cheating in location based social networks like *FourSquare*.

The rest of the chapter is organized as follows. Section 3.2, we focus on the architecture of our landmark enumeration system, *Landmarker*. Next, in Section 3.4 focuses on the overview of the workings of a *RetailGuide* app based on virtual landmarks. In the subsequent sections, we discuss the different smartphone based landmarks, metrics, experimental setup, results, *SignFinder* app, cheating prevention mechanism and future works.

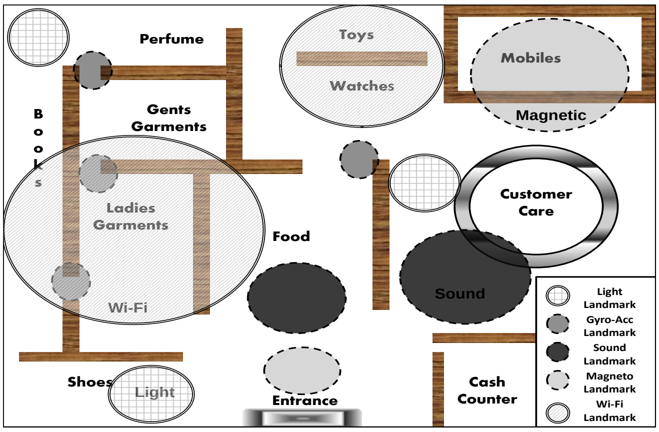


Figure 3.1: Concept image of a landmark augmented shopping mall needed for *RetailGuide* Application. The shapes of the landmarks are shown as circular, but in reality it can take any geometrical shape under some area bound.

3.2 Landmarker:Architecture and Methodology

In order to identify the above mentioned virtual landmarks, we built a landmark identification service called *Landmarker*. As discussed earlier, *Landmarker* is the main component of *RetailGuide* app. We outline the broad structure of the organization and working of the landmark identification service, i.e., *Landmarker* and consequently dive deep into the design specifics of each of its components.

3.2.1 Description of Design Specifics of *Landmarker*

As a part of landmark identification service, the sensor data of mobile phones are collected and send to the cloud server for landmark pruning. On the cloud, we cluster the processed sensor data to get the sensory landmarks. The landmarks which recur in multiple traces are more probable to be correct and are considered stable. These stable landmarks are then stored in a cloud database.

Architecture Outline Figure 3.2 shows the overall architecture of *Landmarker*. As shown in step 1 of figure 3.2, the first step includes the collection of sensor data from different devices and extraction of specific features after proper sampling and noise removal. Then, k-means clustering algorithm is used to cluster the sensor data in higher dimensional feature space. Next, we use the dead

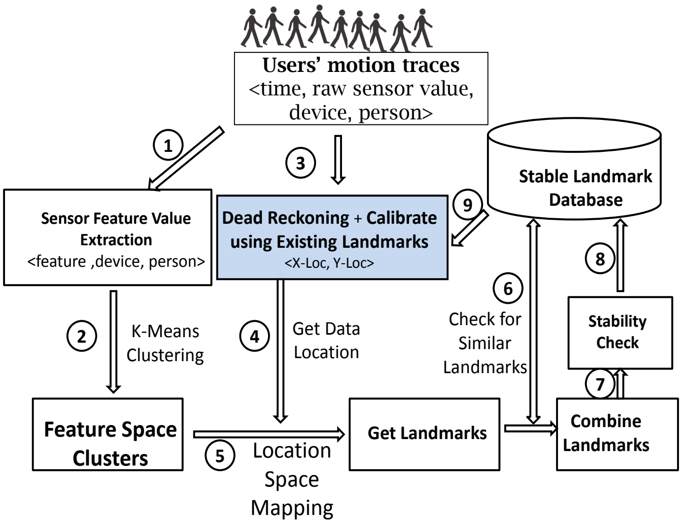


Figure 3.2: Architecture of Landmarker : the landmark pruning system

reckoned location estimate of the sensor data and map the clusters in geographical location space, as mentioned in steps 3 and 4 of figure 3.2. Clusters in both, location and sensor space correspond to landmarks. Lastly, we comb through different traces and obtain the stable landmarks to store them in a database (steps 6, 7 and 8 of figure 3.2). After the initial bootstrapping process, we can use these landmarks to provide the interesting service of minimizing the error of location estimates, which in turn helps to find more stable landmarks later. This feedback loop is completed through the step 9 of figure 3.2. Various processes involved in the architecture are explained in detail in the following subsections.

3.2.2 Sensor Data Collection

We need to collect the raw sensor data from various sensors in the mobile phone. Every collected sensor data tuple is denoted by $<$ Time stamp, Sensor Value, Device ID, Person ID $>$. The sensor data collection process can be further divided into the following:

(a) Data Sampling: The entire experiment is performed on android platform hence we use android APIs to collect data from various sensors. We collected data with the fastest permissible data delay in android, i.e. $SENSOR\_DELAY\_FASTEST$, for each sensor via the sensor manager module. However, the sampling rates of the hardware sensors (e.g. Accelerometer) differ across devices. Generally, sampling rate of Accelerometer ranges between 90 Hz - 100 Hz, whereas for Magnetometer it is around 40 Hz - 50 Hz. The collected sensor data is post-processed and the final data is prepared assuming that it is sampled at a constant frequency of 50 Hz for uniformity in data analysis. This means the same reading of Accelerometer appears twice in processed data. Noise removal is done by passing it through a low-pass butter-worth filter. The data are then normalized within a range of [-1,1] thus ensuring an uniform scale for clustering in feature combination scenario.

(b) Dead Reckoning: Each of the raw sensor data tuple is automatically not annotated with any location space co-ordinate. But, for the formation of landmarks from clusters, it is necessary to have these co-ordinates. So, we use dead reckoning, which helps to figure out an approximate path trace taken by the user using the accelerometer, gyroscope and compass readings. We could have integrated acceleration twice and obtained the distance if it were a robot or car. But this approach is erroneous when applied to smart phones, as shown in [79]. We instead employ a pedometer algorithm which counts the steps taken by the user. We find the peaks in the accelerometer-z data points and estimate the stride length based on the number of steps taken per unit time [41]. The method of dynamic time wrapping as discussed in [54] is used for removing noise and false peaks. By reading the compass readings provided by the phone, we obtain the direction of motion. As described in detail in [79], this can be affected by the magnetic fluctuations indoors and therefore we have further removed this noise by opportunistically comparing with the angle calculated from the gyroscope readings and removing the extra bias. Thus we annotate each sensor data tuple with a relative (x,y) co-ordinate.

3.2.3 Sensor Feature Extraction

For each sensor, we capture different values through android APIs corresponding to users' movements or surroundings. For example, in case of accelerometer, we obtain the acceleration values at X, Y, and Z axis. Then, a function suitably combines all or part of these values in raw or derived form to capture the distinctive aspect of this captured sensor dataset. For example, the captured accelerometer readings are represented as modulus value ($\sqrt{acc_x^2 + acc_y^2 + acc_z^2}$)[79] and SMA [1] ($|linearAcc_x| + |linearAcc_y| + |linearAcc_z|$) [2], as shown in the table 3.1 [19, 57, 86]. Next, we select a suitable fixed length window and calculate mean and standard deviation of the values under that window which comprises the feature set of the particular sensor. Thus, to get proper information from the collected sensor data, we have extracted several features specific to a particular sensor, as described in the table 3.1. These are chosen after reviewing previous works in the area of activity recognition. Feature extraction from the sensor data tuples is done to be used for clustering in order to obtain the landmarks.

Table 3.1: Features selected for Landmark Identification

Sensor	**Feature**
Accelerometer	$\sqrt{acc_x^2 + acc_y^2 + acc_z^2}$, $|linearAcc_x| + |linearAcc_y| + |linearAcc_z|$ (SMA)
Magnetometer	$\sqrt{mag_x^2 + mag_y^2 + mag_z^2}$, $\frac{d}{dt}(mag_y)$
Gyroscope	$gyro_z, RotationMatrix_z$
Sound	Sound intensity (in dB)
Light	Intensity
Wi-Fi	RSSI
GSM	Signal Strength

[1] It is an efficient depiction of the energy of motion. SMA is defined as the acceleration magnitude without gravitation value summed over three axes within each window normalized by the window length [86].

[2] Linear acceleration (linearAcc) is the raw accelerometer readings without gravitation.

3.2.4 Feature Based Clustering

To find any unique characteristics of the landmarks, we mine the features extracted from sensor data. We employ feature based clustering method (k-means clustering) on this processed sensor data. In this clustering, different dimensions are the features of different sensors. Here, the similarity between two points in feature space is measured using euclidean distance. However, in case of Wi-Fi, we introduce a special measure which is elaborated below :

AP Similarity Signature : The value of the access point similarity signature of Wi-Fi at two locations l_1 and l_2 is calculated as given in [79] by the formula

$$S = \frac{1}{|A|} \sum_{\forall a \epsilon A} \frac{min(f_1(a), f_2(a))}{max(f_1(a), f_2(a))} \tag{3.1}$$

where f_1, f_2 are the RSSIs of the APs at two locations l_1 and l_2 respectively and $A = A_1 \cup A_2$ denotes the total number of access points at two locations l_1 and l_2. Thus, there is low distance between locations which have similar set of APs with approximately the same RSSI in the Wi-Fi feature space.

We have used the following subprocesses while implementing k-means :

Selection of k : The optimal k is chosen by clustering random samples of the data for different k and choosing the one for which the intra-cluster centroid distance is minimized as mentioned in [64].

Selection of initial seed : As given in [25], the initial seeds are chosen by clustering over random samples of data and choosing the centroids of the cluster which performs best according to the distance metric given in.

But, it has to be noted that not all clusters identified from k-means qualify as candidates for landmarks. Dense clusters which can be easily distinguished from its neighborhood clusters are chosen. This is quantified by the low average intra-cluster centroid distance and by the high average inter-cluster centroid distance. Thus these two characteristics essentially ensure the separation of a landmark in feature space. Once, the clusters satisfy these threshold conditions (properly normalized, by taking note of the dimensionality), they are then passed on for location space mapping. We have tried out other clustering algorithms which yield similar results (EM algorithm, hierarchical clustering, DBScan), but they did not suit the purpose or did not provide desirable results.

3.2.5 Clusters to Landmarks

We check if the clusters thus obtained, transform into spatial landmarks. For this a mapping of each of the cluster-points in the feature space is done to the location i.e (x, y) coordinates, as show in figure 3.3. These location points are now fed to the augmented k-means algorithm for clustering. The clusters thus

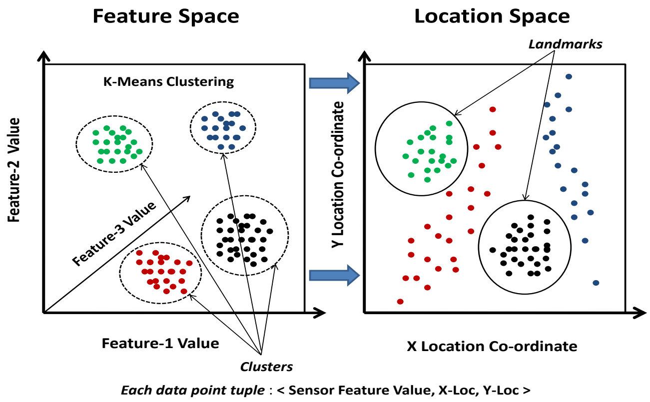


Figure 3.3: Landmark Formation : Feature space to Location space Mapping

formed are an ideal candidate for a landmark, as they contain points which are near in both feature and location space. In figure 3.3, some of the clusters in feature space are scattered in location space and hence they are not considered as landmarks. We restrict the area of these landmarks by applying an appropriate area threshold in location space.

3.2.6 Combining Landmarks

Stability of the landmarks identified on a mobility trace has to be determined. For this it is necessary to combine similar landmarks from different traces into a single stable landmark. The traces are logged by changing devices, person, and time. A landmark would be considered stable if the landmark (if it) appears at roughly same location across various traces. The combination of similar landmarks from different traces are illustrated in figure 3.4. The combination is done by taking the average of the corresponding points in the two landmarks. The basic assumption is that errors produced in different samples are independent. Once the combination is done, we further prune the points which do not fit in the landmark area threshold. It may so happen that a landmark may not occur in all traces. We call a landmark stable if it appears at least in 50% of traces. The landmarks which are identified to be stable across different heterogeneous traces are stored in a database.

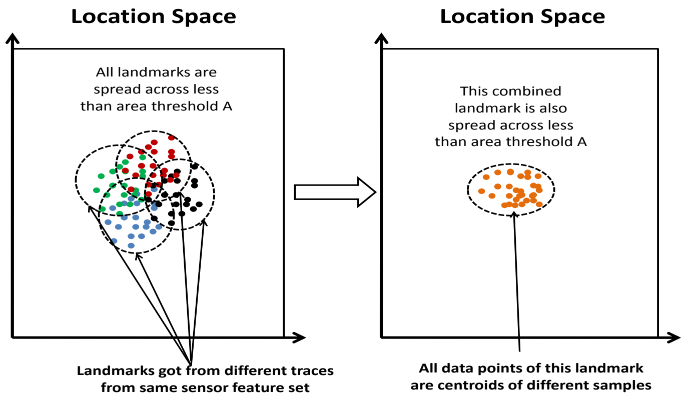


Figure 3.4: Combining landmarks from different traces to find stable landmarks

3.2.7 Calibration of location of sensor data

Simple dead reckoning based location estimation will lead to error, as we have discussed in the above section. To overcome this problem, after the initial identification of a set of landmarks with estimated location, we continuously re-calibrate their positions with respect to newly arriving data points by using the concept of Simultaneous Localization and Mapping (SLAM) [38]. When a new trace generates a landmark, it is tagged as one of the existing landmarks (say) x, if it is close to x in both feature and cartesian space. Simultaneously we correct the given location by shifting the subsequent location points by the difference in the two (previous and current) estimates of the landmark location. Interestingly, this calibration step also helps in the convergence of the locations of a stable set of landmarks.

3.3 Implementation Details of *RetailGuide*

This section describes the implementation details of the developed application *RetailGuide*. Figure 3.5 actually depicts the modules of the *RetailGuide* application which comprises of the landmark pruning module named *Landmarker* (discussed in Section 3.2), comment and offer managing service, front-end customer side android app and mall owner side interface. Each of these is explained in detail along with the challenges faced while building them.

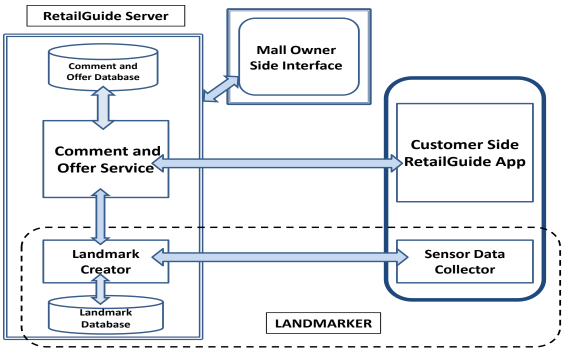


Figure 3.5: Implementation blocks of *RetailGuide* App with landmark pruning system *Landmarker*

***Landmarker* :** The *Landmarker* service runs the clustering and landmark identification service. Its input is the set of sensor data sent by the phone recorded during a user trace, via the *Sensor Data Collector* module residing on customer's smartphone, as shown in figure 3.5. Then this sensor data is sent to the server side, where the clusters generated using k-means clustering and the activity trace are stored in the database using *Landmark Creator* module, as shown in figure 3.5. Subsequently, with time database of stable landmarks are created and saved in the database. The process is described in detail in section 3.2.

Comment and Offer Service : This module manages the comments sent by the customers and offers pushed by the mall owner. It appropriately tags comments or offers to the nearest virtual landmark pruned by *Landmarker* module, as depicted in figure 3.5. Moreover, the database for storing offers and comments is the repository for fast access. It is maintained in MySQL with relational tables maintained each for user, traces, landmarks and comments. This database acts as a way of communication between the App, *Landmarker* and the feedback sent to the users and mall owners. A PHP push service is running as a cron job which frequently polls the database for changes. If a users location has been updated, then corresponding location specific notifications are sent to the user. This is accomplished by the use of Googles free Cloud Messaging Service.

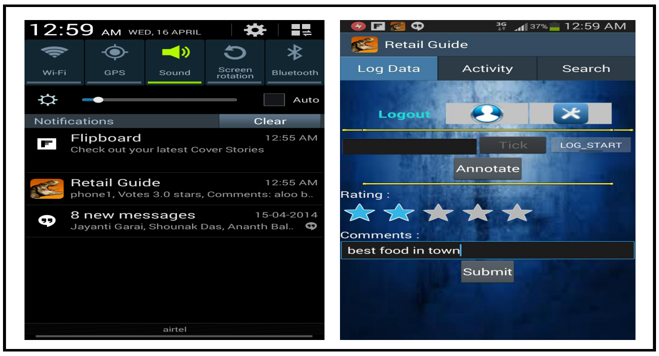


Figure 3.6: Screenshots of *RetailGuide* Android Application.

Customer Side *RetailGuide* App : The Android App built for the front-end of RetailGuide is a light weight android application shown in figure 3.6. It is used to collect the sensor data in the background via a specific module shown in figure 3.5. It also collects user activity of the app, like comments inserted by the user, notifications received and changing the parameters of logging sensor data. It records the energy levels of the phone for evaluation purposes. Thus a user, on the front end is seeing just a text-box and a couple of buttons to start/stop logging as shown in figure 3.6. This easy to use interface makes the interaction time faster. Once, the user enters a mall, he starts logging. He can move around the mall at various speeds and comment about things he sees, likes or dislikes. Suitable location updates are sent to the server. If there are any relevant notifications that can be sent to the user, they are pushed to the phone. This relevancy is determined on the server, by the location coverage of the reviews or offers (by the mall owner) in the database.

Mall owner side Interface : The mall owner perspective, shown in figure 3.7, includes tools for visualization of the various kinds of retail analytics, like landmarks, location heat maps, accuracy and sentiments of the comments entered, the number of offers pushed to the user. This module interacts with server side of *RetailGuide* to get relevant information as described in figure 3.5. This interface, which is implemented using PHP, makes it easier for the owner to visualize on Google Maps. She can integrate it with the main distribution system and can pinpointing the location he wants an offer to be relevant. He also has the option to set the coverage of an offer, which determines how often it is pushed to the customer's app.

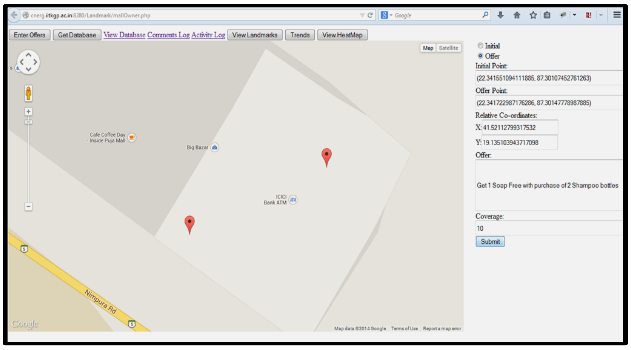


Figure 3.7: Mall owner side Interface for *RetailGuide* App

3.4 Smart phone based Landmarks

In this section, we will restate the idea and semantics of a smart phone based landmarks. It will help in understanding how these landmarks may help an application running in phone to understand the actual context of user.

3.4.1 Classification of Smart phone based Landmarks

We can classify different landmarks pruned from smart phone sensor data in the following way:

1. **Sensors' feature based Landmarks**

   (a) *Single Sensor Landmarks* Landmarks which are created from the clusters of single sensor information, e.g. , GSM blind spots (from GSM chip), a low-lit area (from light sensor), an area with unique magnetic fluctuation (from magnetometer) etc.

   (b) *Multi Sensor Landmarks* Landmarks which are generated from the multi-sensor clusters in higher sensory dimensions, e.g., an accelerometer-gyroscope-light induced landmark etc. Generally people slow down in a low-lit corridor while walking. It can result in this type of landmark.

2. **Spatial Landmarks**

   (a) *Universal Landmarks* Some sensory signatures which correspond to a particular landmark is universal in nature, i.e., independent of the location, time etc. For example, unique signature created by phone's accelerometer in stairs or in speed-breaker are unique across universe.

   (b) *Local Landmarks* However, some sensory landmarks are pretty much unique to the locality or the time domain they belong. They are learned dynamically from the environment and later stabilized over time. For example, a specific set of Wi-Fi access point MAC signatures or a specific sound intensity of a place can uniquely identify a place in a local domain.

3. **Perceptual Landmarks :**

   (a) *Highly Perceptual Landmarks* Corridor corners or indoor entrances which can also easily be perceived by human senses, can be categorized as highly perceptual landmarks.

   (b) *Least Perceptual Landmarks* Some Wi-Fi or magnetometer signature based landmarks can be classified in this category of landmarks.

3.5 Experimental Setup

We conduct our experiments by collecting the motion traces of participants with smart phones in user's hands. The phones used are Samsung Galaxy S2 I9100G and Samsung Galaxy S3 I9300. These phones provide us with sensors such as accelerometer, gravity, gyroscope, magnetometer, orientation, sound, light, Wi-Fi and GSM. Both of the phones are upgraded to android 4.1.2 (Jelly Bean). In our experiments, we collect these sensors' data while walking with the phone held in the hand, facing upwards. We use *RetailGuide* app [9] with *Landmarker* service running in the background to conduct our experiments. The data recorded internally is sent to the *RetailGuide* server. The server side code is written using php and MATLAB, and implements the dead reckoning, clustering, and landmark signature-matching algorithms. We assume constant orientation of the phone for easy understanding of the setup[54].

We have conducted our experiments in three places to explore the robustness and usability of the system in different scenarios. Initial sets of experiments are done inside the Computer Science and Engineering Department, IIT Kharagpur and an outdoor market (named Technology Market) for fully understanding the dynamics of stable landmarks. Subsequently, we mainly concentrated on the usability of *RetailGuide* app through our experiments. Following are the description of experiments performed in three different places:

Department Experiment: We have conducted different sets of experiments in the corridors of the Computer Science Department (500 m^2 area in indoor environment) (See figure 3.12). These sets of experiments are done for testing the feasibility of *Landmarker* service. To understand the dynamics of stability of these sensor landmarks and usability of *Landmarker* service, we performed experiments where the user traces the same path multiple times. We have also done our experiments with *RetailGuide* app in this indoor corridor by mimicking a shopping mall scenario. In both the cases, data were collected multiple times on two devices, at two different times of the day - *morning* and *night* by four volunteers. This area of experiment had Wi-Fi connectivity.

Market Experiment: For stability analysis of landmarks in outdoor scenario, we have conducted different set of experiments in around 600 m^2 area in an open market area having a few permanent shops (Technology Market). Here also, data were collected multiple times on two devices, at two different times of the day - *morning* and *night* by four volunteers. We mainly concentrated on the creation of the landmarks and their stability in outdoor settings. This outdoor area of experiment also had the Wi-Fi connectivity.

BigBazaar Experiment: To test the practical usability of our *RetailGuide* app, we have conducted an experiment in a nearby Big Bazaar (one of the largest shopping mall chains in India) outlet. We have done our experiments in a floor having different sections using *RetailGuide* app. Four volunteers helped us to test the app extensively. We have also asked the volunteers to express their feelings about the different sections of the mall through comments in *RetailGuide* app. Some of the snapshots during the experiments are shown in the figure 3.8. This indoor shopping area did not have the Wi-Fi connectivity and we used $3G$ connectivity for the experiments.

We have also simultaneously collected GPS readings to compare with our results. We have used *GPS Logger for Android* app for logging GPS readings in both *Department Experiment* and *BigBazaar Experiment* for *RetailGuide* app. We have also recorded the battery levels of running *RetailGuide* app in different experiments for calculating the energy footprints.

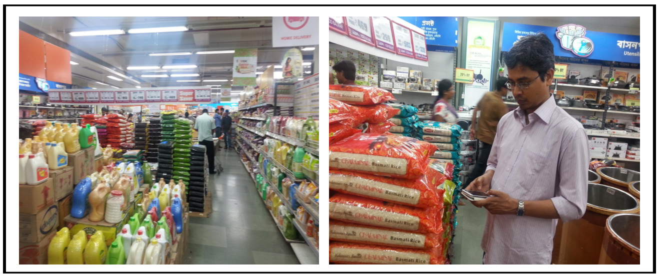


Figure 3.8: Snapshots of *RetailGuide* experiments in BigBazaar

For ground truth recording of actual location, we have numbered the different points in the experimental areas and asked the volunteers to input the numbers when they cross them. And we also created a php based server side interface, shown in figure 3.7 to push offers to specific users and monitor the experiments.

The tuned values of landmark area threshold, feature space nearness value and confidence count are taken as 6 m^2, 0.7, and N/2 (where N is the number of traces) respectively. Reasons behind these parameter selection is elaborated in the following section.

3.6 Parameters of the system

In order to discuss the results and core issues of the problem, we would like to introduce a few parameters which are crucial to detect and characterize virtual landmarks.

(a). Area threshold for landmark - This is defined as the area covered by a particular landmark. Since the effect of a sensor eases out slowly a threshold need to be defined to mark the area of a stable landmark. While conducting experiment, we have found that, except few, in most of the cases, the effect of a sensor is most pronounced within a 6 m^2 area. Hence most of the sensor-spatial clusters built from light sensor or gyroscope sensor, will generally cover around 6 m^2 as shown in figure 3.9. However, a few exceptions like Wi-Fi landmark cluster might cover an area of close to 30 m^2 as shown in figure 3.9. Since majority of single sensor clusters are around 6 m^2, multi-sensor landmark clusters are also mostly around 6 m^2. For this reason, in order to maintain uniformity, we have assumed that a stable landmark covers an area of 6 m^2. Moreover, if we have taken wider area for landmarks like 10 m^2 then we might have lost the required localization accuracy. Wider area landmarks would not have helped in pinpointing location more accurately. As this choice of area has provided us with satisfactory number of landmarks, we have continued with this value.

The cdf graph in figure 3.9 shows that various single sensor features are clustered around different areas in the location space, from the *Department Experiment*. The reason of this variation is due to different level of sensitivity to environment of different sensors.

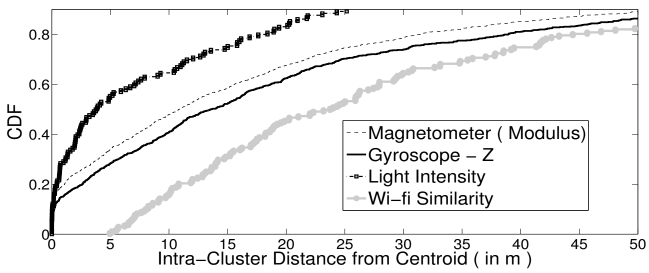


Figure 3.9: CDF of landmark cluster area for different sensors

(b). Number of Features to be Combined - In order to discover landmarks, sensor data from nine sensors are collected. For each sensor, several statistical measures like mean and standard deviation are collected. Therefore each location is characterized by f (= *no. of sensors* $\times$ *statistical measures*) number of features. A group of points may get clustered based on a subset of features – therefore in order to find the best clustering condition one has to exhaustively look into all the subsets which would be 2^f - 1.

This would explode the feature space and hence optimal clustering may not be feasible. But, interestingly we have found that we do not have to consider all the subsets. If we consider only *two* or *three* features in the combination, it suffices our purpose. Both the count of clusters as well as landmarks (co-located clusters) decrease with the increase in combination counts, as shown in figures 3.10 and figure 3.11.

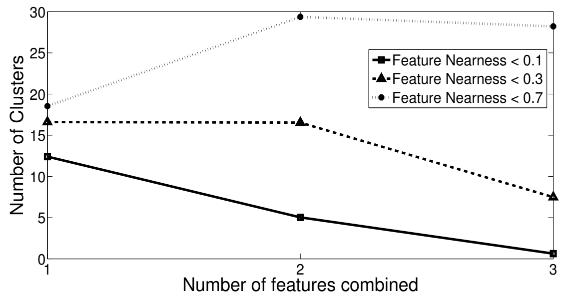


Figure 3.10: Average number of clusters in our experiments. The increase in the number of clusters for parameter nearness 0.1, is due to too much relaxation of the constraint. We are taking too many insignificant clusters into account.

For experimental purpose we have taken only the *mean* of all the sensor data and combined different sensor's data to discover clusters. In figure 3.10 it is seen that the number of clusters formed decrease with the increase in the number of features combined for clustering. Moreover, we also see similar trend in the case of stable landmarks (clusters which clusters both in feature and location space), as shown in figure 3.11. Most importantly, we see a dramatic decrease in the number of landmarks when we combine two or three features. This means that there is no need to explore the combinations which comprise of more number of features.

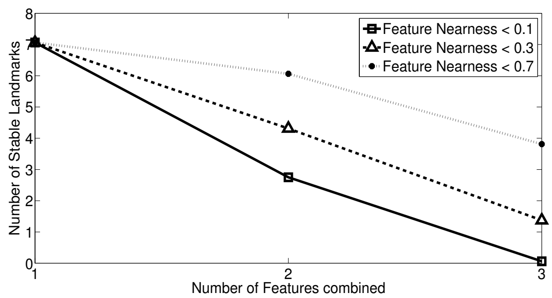


Figure 3.11: Average number of landmarks in our experiments

(c). Feature Space Nearness - It is the measure to determine whether two points belong to the same cluster. That is, given a cluster which has been built considering some sensor features, all pairs of points in that cluster need to be less than the specified threshold – which is termed as *Feature Space Nearness*. The higher the value of feature space nearness, the closer the data points are in feature space. For this parameter, we have chosen a value of 0.7, as this is a moderate trade-off value between the quality and quantity of clusters, as shown in figure 3.11.

(d). Confidence count - It is the number of path traces in which a landmark is found, for example, if N traces are considered, confidence count N/2 means that the landmark has been detected in at least N/2 traces. The higher the value of the confidence count corresponding to a landmark, the higher its probability of being stable. We have chosen the confidence count parameter value as $N/2$ due to the availability of stable landmark clusters, shown in the figure 3.17.

3.7 Experimental Results

In this section, we first evaluate the potential of discovering stable landmarks using mobile phones and then check the efficiency of *RetailGuide* to provide smarter analytics and accurate recommendations. In order to accomplish an overall evaluation, we perform the following investigations.

1. The subset of sensors responsible for most stable landmarks.
2. The effect of heterogeneity on formation and stabilization of landmark
    - change in user collecting the data
    - change in the time of the day
    - change of the devices
3. What is the effect of heterogeneity on landmark based localization system?
4. The impact of indoor and outdoor scenarios on landmark creation or stability.

5. How accurately does *RetailGuide* reflect users' trails and sentiments in an indoor space?

   - Accuracy of Comments and Pushed Offers in *RetailGuide*
   - Effect of Stability of Landmarks on the functionality of *RetailGuide* system
   - Energy Cost of *RetailGuide* system

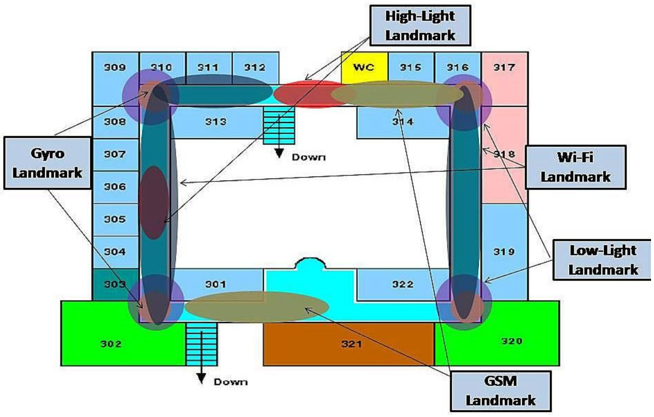


Figure 3.12: Landmark annotated Indoor area map. This is the map of Second Floor, Computer Science and Engineering Building, Indian Institute of Technology, Kharagpur.

3.7.1 Sensor wise Analysis

There are different possible sensor features and their combinations. The question is that which feature set of sensors gives more stable landmarks consistently. For this, we have pruned different landmarks found from different sensor feature sets. We have done the experiments both indoor (*Department Experiment*) and outdoor (*Market Experiment*). The landmarks found in the *Department Experiment*, which is done in the indoor department corridor, is shown in figure 3.12. We have found a total number of 13 stable landmarks which are formed using single or multiple sensor features. Main contributory sensors for these stable landmarks are depicted in the figure 3.13. In the outdoor *Market Experiment*, we have found 7 stable landmarks and the main contributory sensors for the stable landmarks are shown in the figure 3.14[3]. We can see that inertial sensors like gyroscope, magnetometer, or software sensor like orientation (rotation matrix) primarily produce stable landmark. Moreover, some of the features like sound intensity does not give any stable landmark in indoors, even if we couple them with the stable landmark generating features. In *Market Experiment* there is a presence of numerous light based stable landmarks and some sound based stable landmarks due to distinctive lighting and sound situation in the open market area. These results can help us identify the set of sensors which can be turned off in case of low battery power situation and prune the dataset required to be send to the server.

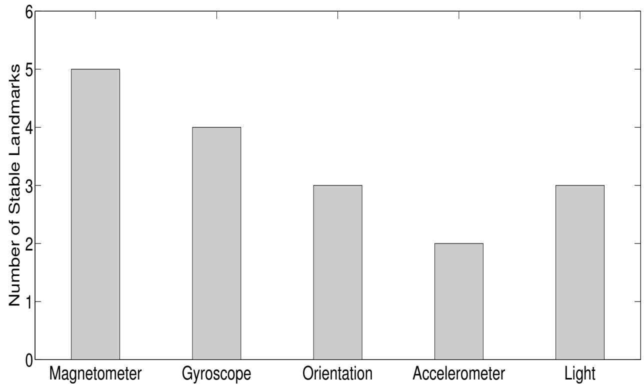


Figure 3.13: Sensor-wise Stable Landmarks for the *Department Experiment*

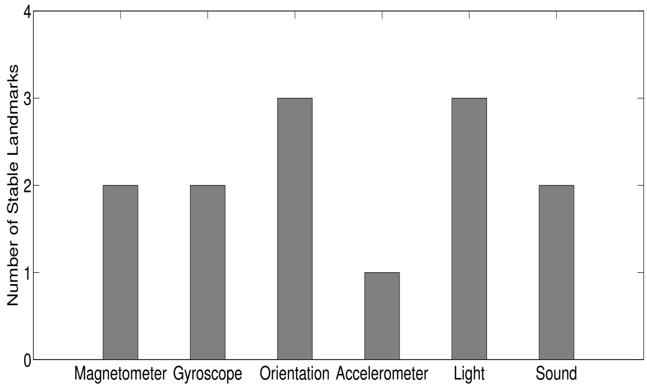


Figure 3.14: Sensor-wise Stable Landmarks for the *Market Experiment*

[3]Total sum of individual sensory landmarks is greater than the number of actual stable landmarks is because multiple sensors sometimes constitute one landmark.

3.7.2 Analyzing different types of Landmarks pruned

In this subsection, we will dissect the semantics of different types of landmarks and their respective contribution to the overall landmark population. Figure 3.15 shows the characteristics of landmark population in the department corridor area and figure 3.16 shows the different types of landmarks in the open market area.

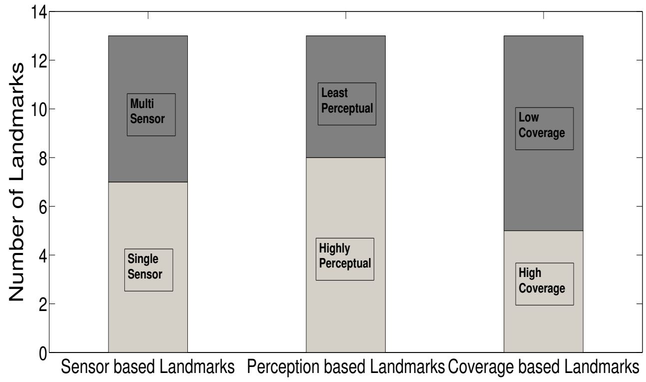


Figure 3.15: Different types of Landmarks in Department Experiment

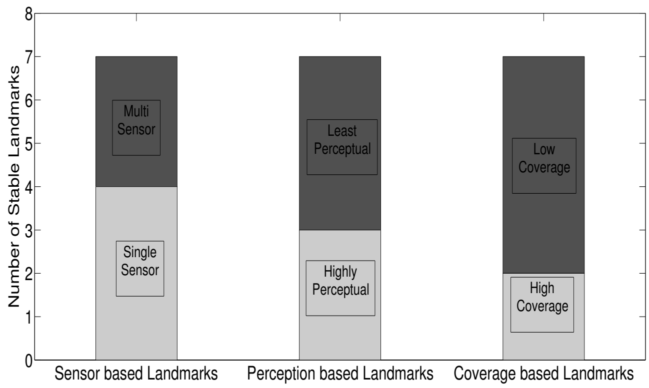


Figure 3.16: Different types of Landmarks in *Market Experiment*

From figures 3.15 and 3.16, we found that we can obtain a substantial amount of single sensor landmarks, due to varied environment of the experimental areas. If we have time constraints, single sensor based systems also can work really well in indoors. It is interesting to note that, we get a good amount of highly perceptual landmarks[4] like corners or dark regions or sound zones, with respect

[4]Corridor corners or indoor entrances which can also easily be perceived by human senses, can be categorized as highly perceptual landmarks.

to least perceptual landmarks [5], in both of the experiments. Moreover, we get similar number of coverage based landmarks[6] in both the experiments.

3.7.3 Effect of Heterogeneity in *Landmarker* System

We investigate the effect of heterogeneity on the stability of landmarks pruned using *Landmarker* in this section. By stability of landmark in face of heterogeneity, we mean that the landmarks are invariant in spite of changing the devices, the time frames or persons carrying the devices in different experiments. The following results are from the *Department Experiment* using *Landmarker*.

(a) Person Heterogeneity: We felt that the variation of the walking style, movement speed of different persons can have an impact on the stability of landmarks. Therefore, we have conducted a small-scale experiment to collect traces with four persons. Figure 3.17 shows that the number of landmarks obtained by different users decreases as the confidence count of the landmarks increases. However, most of the users obtain roughly same number of landmarks. The number of landmarks obtained at confidence count N/2 is reasonable and on manual inspection are found to be of 'optimal' size (not too large or almost invisible); Here N is the total number of traces collected from the users i.e. each user contributes N/4 traces. Hence, N/2 is considered as default confidence count. The graph shown in inset of figure 3.17 shows that we are getting around 12 stable landmarks, which is considerably high for such a small indoor space. Each individual users besides discovering these stable landmarks also identify several 'unstable' landmarks.

(b) Time Heterogeneity : In this case, we have studied the effect of time of a day on the stability of landmarks. We have taken two time periods, i.e. day period (10 a.m. - 1 a.m.) and night period (8 p.m. - 11 p.m.), for collecting the traces using mobile devices. The intuition behind this experiment is that the

[5] Some Wi-Fi or magnetometer signature based landmarks are classified in this category of landmarks.

[6] High coverage landmarks have an area around 6 m^2 and low coverage landmarks have an area around 2 m^2.

signatures like light, sound etc. change with the time of the day, e.g., a busy shop becomes silent at night.
Although the count of the landmarks does not vary much; unlike previous case, the comparison for actual landmarks in figure 3.18(a) reveals that approximately 33% of the landmarks are stable. It is lower than the case of different users.

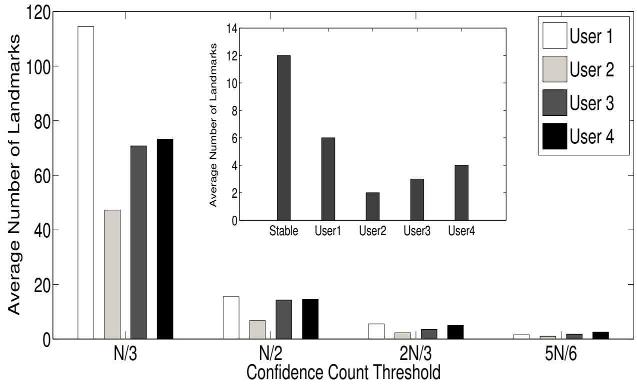
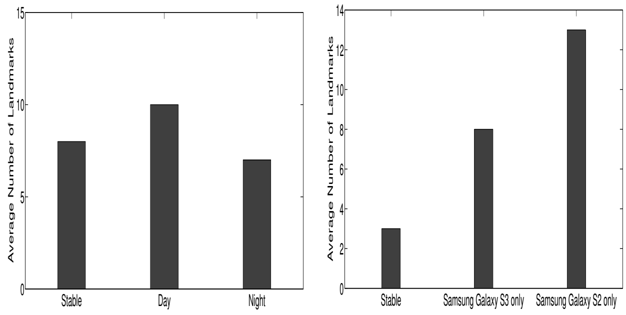


Figure 3.17: Number of landmarks for different users. Here, N is the total number of traces. That means each of the user has moved N/4 number of times in the designated area. Inset figure shows the comparison of number of stable landmarks and user specific unstable landmarks

(c) Device Heterogeneity : We have repeated the experiments of collecting traces with two devices, namely Galaxy S3 and Galaxy S2, to test the effect of change of device on the stability of a landmark. Figure 3.18(b) shows that we get around 3 stable landmarks in this indoor space, which is considerably less than the earlier two cases. So, we can conclude that the effect of device heterogeneity has the most impact on the stability of landmarks.

It is interesting to note that even though both the devices are from the same manufacturer and same series, there have been a considerable difference of the hardware, subsequently, the landmarks. So, the inherent difference of sensitivity and precision of different sensors has a telling impact on the stability of landmarks. On the other hand, the effects of change of time and persons, are significantly lower than the case of devices. Therefore, if we want to create a corpus of stable landmarks to augment the location based services, we have to organize it with respect to different class of devices or a set of sensors, as hinted by [35].

Energy and Accuracy: ***Landmarker*** **vs GPS** It may seem that the advan-



(a) (b)

Figure 3.18: (a). Comparison of number of stable landmarks and time specific unstable landmarks (b). Comparison of number of stable landmarks and device specific unstable landmarks

tages of landmarks might be just that of its capacity to become virtual dropboxes. In order to justify the localization using landmarks instead of using GPS, we did a comparative study of GPS and Landmarker both in aspects of energy consumed as well as localization accuracy. This comparison results are from *Market Experiment*. Figure 3.19 shows that the energy consumed by *Landmarker* without Wi-Fi is almost comparable with GPS even with majority of the sensors constantly polling at a frequency of 1Hz. But, figure 3.20 clearly notes that this is justified because of the accuracy of the *Landmarker* based system compared to that of the GPS. Interestingly, although most of the time our system works well in terms of localization accuracy, but sometimes GPS based system gives lower error. It is due to unavailability of satisfactory number of stable landmarks in some areas. This makes it clear that in order to implement a virtual dropbox system, virtual landmarks are necessary because of its localization as well as abstraction powers.

3.7.4 Comparison of Indoor and Outdoor Scenario

In this section, we will discuss in detail about the results regarding the landmark creation and localization error found from our *Department Experiment* and *Market Experiment*. These results will shed light upon the workings of *Landmarker* system in indoor vis-a-vis outdoor scenarios.

Analyzing stability of landmarks in Indoor and Outdoor: In general,

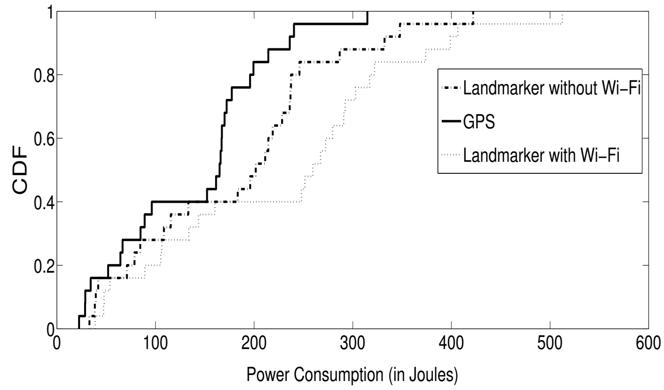


Figure 3.19: Comparison of Energy Consumption in *Landmarker* service and *GPS based localization.*

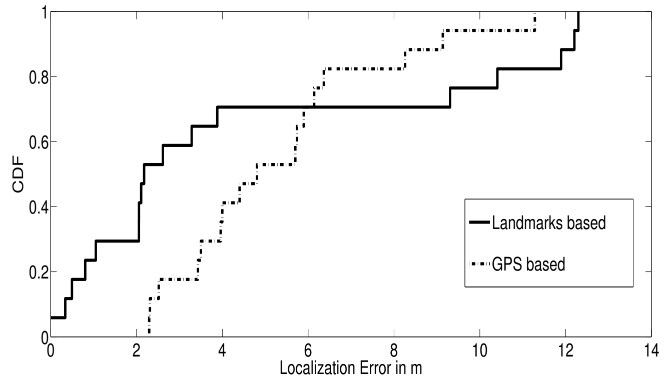


Figure 3.20: Comparison of Localization Error in *Landmarker* service and *GPS based localization.*

indoor space contains more landmarks than the outdoor space, which we can get to know from figure 3.21. The reason is the availability of more signatures in the indoor space. But, if we increase the confidence count values, the difference of the number of landmarks decrease.

Moreover, the stability of the landmarks are also observed more in the indoor space as the change of the surrounding environment is more drastic in outdoor space. On an average, the difference of number of landmarks is around 6, even in this small area experiments. Moreover, we are able to find a considerable amount of landmarks in outdoor space also, which can easily augment GPS in outdoors.

Comparing Localization Error in Different Settings: We have created a stable landmark database using a specific triplet of < Device, Person, Time

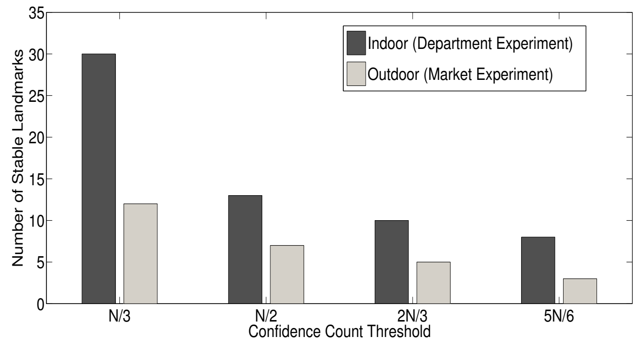
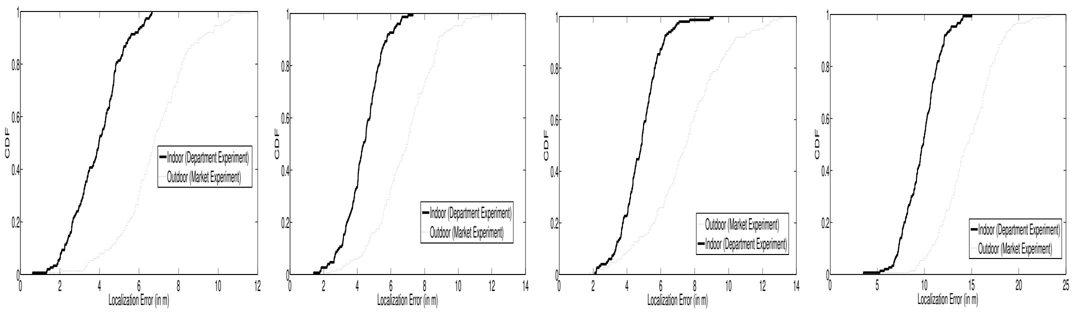


Figure 3.21: Number of Stable Landmarks in Indoor and Outdoor Experiment Spaces



(a) Localization Error if we measure without changing anything

(b) Localization Error if we measure by only changing person

(c) Localization Error if we measure by only changing time

(d) Localization Error if we measure by only changing device

Figure 3.22: Localization Error (in m) in different settings in both indoor and outdoor spaces. We build our landmark based localization system at a specific (Device, Time, Person) setting and we change one parameter to see the effect.

>. In order to understand the impact of an individual, the time and device, we change any one of these three parameters and test the deviation from ground truth (identified landmark), i.e. localization error. In figure 3.22, we can see that if we change device, person or time, the localization error will increase. However, the effect of device change on error is the most significant, which is in line with our previous findings. This trend also remains similar in also outdoor experiment, as shown in figure 3.22. In general, we can see localization error in outdoor hovers around $\sim 10m$ and in indoor the localization error is around $\sim 5m$.

3.7.5 Analytics from *RetailGuide*

The task of *RetailGuide* is to properly identify retail space by running the background *Landmarker* service. We initially performed a pseudo experiment in Departmental corridors to test its performance. In this *Department Experiment*, users roam around with smart phones running *RetailGuide* app in the department corridor, which mimics a shopping mall situation in a controlled manner. Corridor corners are named as different sections of a shopping mall like food, clothing, utensils, and cosmetics. Users also comment while moving and get relevant offers cum recommendations. In this experiment, we have considered around 12 landmarks, which we use as dropboxes of comments.

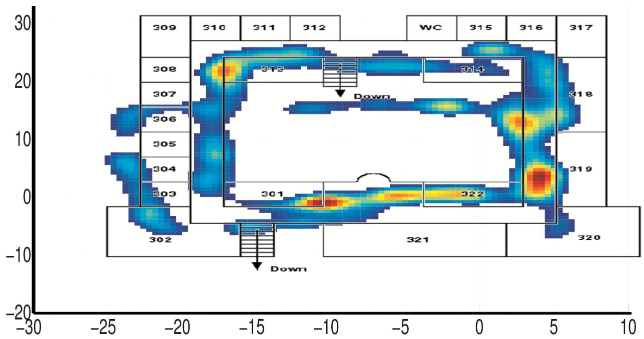


Figure 3.23: Users' movement heat map found from *RetailGuide* Application.

A user's trail is inferred from his movement from the latest landmark. Figure 3.23 shows the heat map of users' movements in the corridor inferred from nearest landmark locations. Clearly, it contains some error as most of the users' movements are rectangular. From the estimation of the position of an user so derived, any comment she posts is tagged with that location by the cloud service. The service also accordingly attach this comment to the nearest landmark. The efficiency of the *Landmarker* algorithm would be measured in terms of the number of times it is attached to the correct landmark. Figure 3.24 illustrates example of comments posted by users at different locations. The circle shaped dots in the figure 3.24 denote correct location tagged comments and star shaped dots denote erroneous location tagged comments. In general, we get around 75% accuracy in attaching a comment to the correct landmark. After this success of pseudo-experiment, we did an experiment in a nearby shopping mall with a set of users to observe the efficacy of *RetailGuide* app in a real world scenario, which

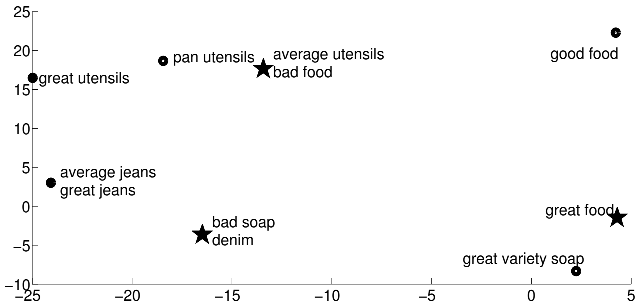


Figure 3.24: Comments of different users through *RetailGuide* Application.

we name as *BigBazaar Experiment*. Users who participated in the experiment are told to express their views through comments about the mall while roaming around with *RetailGuide* application running smartphones.

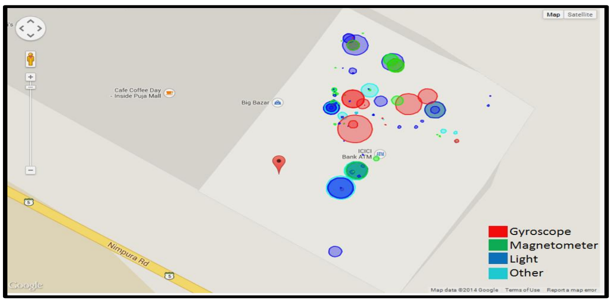


Figure 3.25: Landmark annotated Big Bazaar (Shopping Mall) area map. This is the Google Map snapshot of the Big Bazaar shopping mall building in Kharagpur, India.

Sentiments of Comments in *BigBazaar Experiment*: In order to test the accuracy of The *RetailGuide* in a real mall scenario, we used the parameters already tested in the simulated Department scenario. We chose Big Bazaar, Kharagpur, India, as our experimental location. Each of the four volunteers were instructed to roam around the mall in a random manner in the aisles. They also commented naturally expressing their views about the things they saw as they went around. We also had inserted a bunch of offers in the database, which can be pushed to the user along with the relevant feedback comments based on their location. As the users started their movement, the landmarks database started getting populated and we obtained around 15 stable landmarks. We also observe that the number of comments generated varies from user to user, both in number as well as sentiment, as shown in figure 3.26) This validates that the perception of each user varies and is valuable to the mall owner.

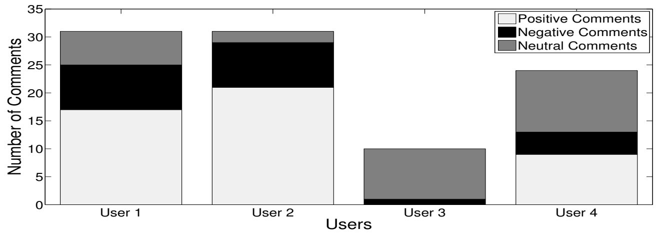
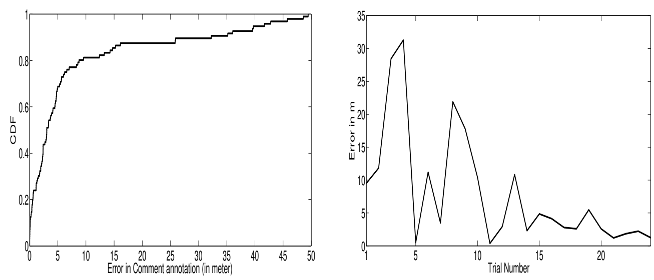


Figure 3.26: Sentiment of Comments of different users through *RetailGuide* Application.

Accuracy of Annotation of Comments in *BigBazaar Experiment*: In our system, each comment is annotated to the nearest landmark which acts as a virtual dropbox. Comments are given by the users via client side of *RetailGuide* app and these are sent to the *RetailGuide* server for the annotation. We argued that there are enough landmarks to make these comments relevant in a mall scenario. We measure the relevance of comments by the distance between the actual location where the comment was made and the location of the landmark to which it is annotated. The measurement of this “error” metric can be seen in 3.27(a). It can be noted that almost 80% of the time, this error is less than 10 meter which qualifies as being relevant to the comment. Also this error is decreasing with trials, as can be seen in 3.27(b).

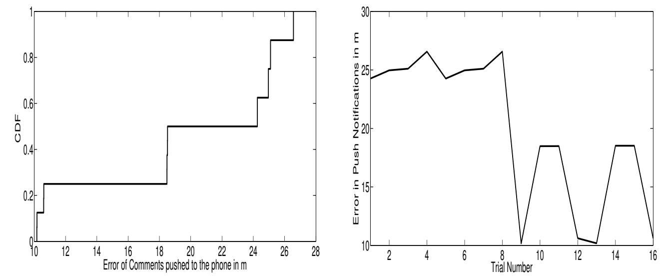


(a) (b)

Figure 3.27: (a). CDF of Accuracy of Tagging of Comments to particular locations of different users through *RetailGuide* Application. (b). Change of Accuracy of Tagging of Comments with Trials.

Accuracy of Pushed Comments and Offers: *RetailGuide* system works better because of increasing comments and traces of the users, it becomes imperative to attract the user to give his data. We argue that this benefit comes in the form of location based reviews and comments of users along with relevant offers pushed by the mall. *RetailGuide* server pushes the relevant offers and comments to the customer side app. This pushing mechanism works based on the location updates sent by the phone during the course of the mobility trace. Each review has a given coverage area depending on its content and relevance. The accuracy of pushed comments is measured by the distance between the user's current location and the location where the comments was inserted. But, note that these offers are in virtual dropboxes. Therefore, the system searches for the nearest such dropbox based on the location and pushes the reviews in those dropboxes. It can be seen in figure 3.28(a) that this error is within the limits of the same locality (say aisle) and also this error is decreasing with more trails as shown in figure 3.28(b).

Moreover, mall owner puts some offers at different locations to attract more customers. Whenever these offers are pushed to the phone, the user can distinguish them as relevant or irrelevant. It can be seen in figure 3.29 that the error of pushed offers is well within the bounds of human perception as majority of these offers are qualified as relevant offers.



(a) (b)

Figure 3.28: (a). CDF of Accuracy of Pushed Comments to particular locations of different users through *RetailGuide* Application. (b). Change of Accuracy of Pushed Comments with Trials.

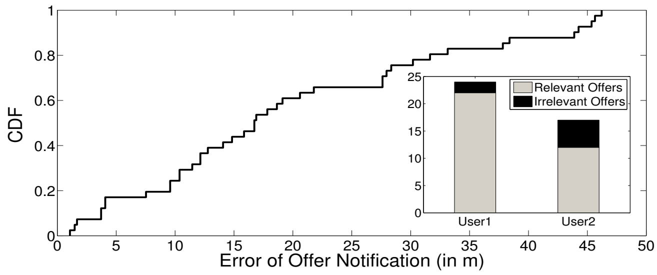


Figure 3.29: CDF of Accuracy of Pushed Comments to particular locations of different users through *RetailGuide* Application. (b). Change of Accuracy of Pushed Comments with Trials.

3.8 Other Possible Applications

In this section, we will discuss some of the initial ideas of applications which can be built easily using smartphone based virtual landmarks. As proofs of concepts, we have also shown some initial experimental results to trigger further explorations.

3.8.1 *SignFinder* App : Generating Virtual Signboards

Besides RetailGuide, the *Landmarker* application can be used to build various other location based application. One such can be *SignFinder* where the app would help in creating virtual signs. We discuss an outline and some initial experiments to prove the feasibility of such app. The motivation behind such app is stated next.

In emerging economies, signs identifying locations of interest is largely missing. For example, names of streets are hardly written on the walls of houses/corners located on the corresponding streets, in most of the smaller (some bigger) railway stations one would not find pointers identifying the ticket counter. The huge penetration of smartphones can be leveraged to compensate this dearth of signs.

This can be achieved (a). if the point of interest can be identified as a stable virtual landmark and (b). if the combinations of sensors which identify the location is unique. One can then develop an app which would annotate the places of interest when people carrying the smartphone move by those particular locations. Repeated annotation by a lot of users can be an excellent crowd-sourcing method to perfect the signs. In previous sections, we have shown that we can find a satisfactory number of stable set of sensory landmarks, in both indoor and outdoor locations. A possibility is that we can use these stable landmarks to create virtual signboards. Here, as an initial proof of concept, we checked whether the stable landmarks formed in our experiments correspond to places of interest and whether semantically similar places (e.g. laboratory) have (near) identical set of sensor features. using the app developed for this purpose which we term as *SignFinder*.

Table 3.2: Signs with corresponding Sensor Signature Vectors

Signs	**Sensor Signature**
Toilet	High Sound, High Light, Low Wi-Fi, High GSM, High Relative Humidity
ATM	Low Sound, High Light, Low Wi-Fi, High Magnetic Signature, Low Relative Humidity
Lab	High Sound, High Light, Low Wi-Fi, Low GSM, High Magnetic Signature, Low Relative Humidity
Stair	High Sound, Low Light, Low Wi-Fi, Low GSM, High Gyroscope Signature Change
Shop	High Sound, Low Light, High Wi-Fi, High GSM, Low Magnetic Signature
Cycle Stand	High Sound, High Light, High Wi-Fi, High GSM, High Magnetic Signature

Experiments for finding Signs

SignFinder app consists of two modules, namely, previously discussed *Landmarker* module and *SignCreator* module. *SignCreator* module takes help of different sensory signature vector to identify generic places like ATM, Toilet, Laboratory etc. while the identified landmarks help to localize them. Some of the sensory signature vectors corresponding to different signs are given in table 3.2.

We have done experiments in the corridors of Department (*Department Experiment*) and in the outdoor market (*Market Experiment*). Users record sensory signatures of different signs and subsequently a small database of signs like *toilet*, *stairs*, *server room*, *office*, *shops*, *cycle stand*, *ATM* etc. as given in the table 3.2 are created. For example, in indoor scenario, signature vector of *toilet* sign looks approximately like *(High Sound, High Light, Low Wi-Fi, High GSM, High Rel-*

ative Humidity) and in outdoor scenario, *shop* sign shows a specific light-sound-gsm signature. Then, we consider stable landmarks created using *Landmarker* module and tag the discovered signs to database of local stable landmarks of a particular area. That is, stable landmarks help in pin-pointing signs.

Results and Research Issues

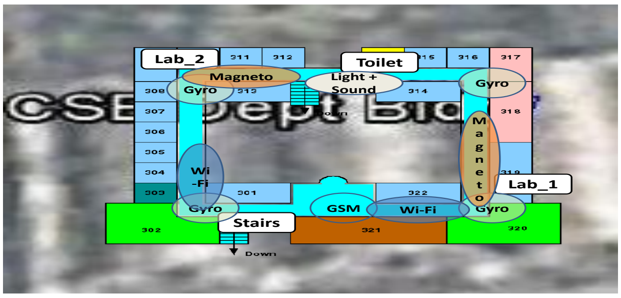


Figure 3.30: Overlay on Google Earth of the Indoor Area with built Signs and stable Landmarks

Figure 3.30 shows the dynamically generated signs in the indoor experimental area in white boxes. If we look into nature of signatures of different signs, we find interesting observations. For example, signature vector of *lab room* looks like *(High Sound, High Light, Low Wi-Fi, Low GSM, High Magnetic Signature, Low Relative Humidity)*, which is found almost similar in different user traces. On the other hand, Figure 3.31 reveals some signs found in our outdoor experimental area. For the outdoor case, the semantics of different signs are different. For example, *ATM* sign shows a strikingly different signature in *(wi-fi, magnetic, sound, relative humidity)* dimensions. But, in both cases, we found a different set of signs which will influence the further explorations.

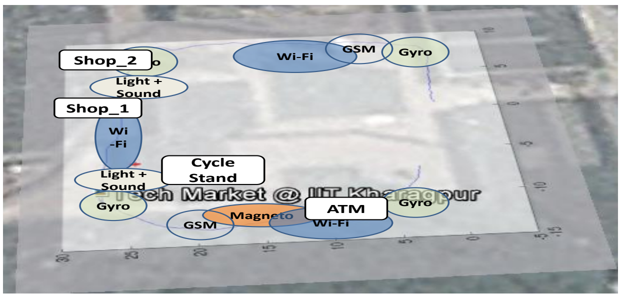


Figure 3.31: Overlay on Google Earth of the Outdoor Area with built Signs and stable Landmarks

We must comment that this is an initial study which illustrates the possibility of developing such an innovative and useful application. However, much detailed experiments need to be done to conclusively establish the feasibility of *SingFinder*.

3.8.2 Location Cheating Prevention in *LBS*

Cheating in location based services (LBS) is pervasive and is difficult to detect. Leader boards and monetary benefits to the toppers have lead to people doing replay attacks either by running a simulator or modifying the GPS driver of the phone [45]. Also, the current preventive measures do not provide fine-grained accuracy checks. For example, a person can "check-in" to a cafe by not buying or by even just standing outside the cafe. This inaccuracy has been a hindrance to the shop owners who are not able to attract enough users with discounts and offers for the user with the most check-ins.

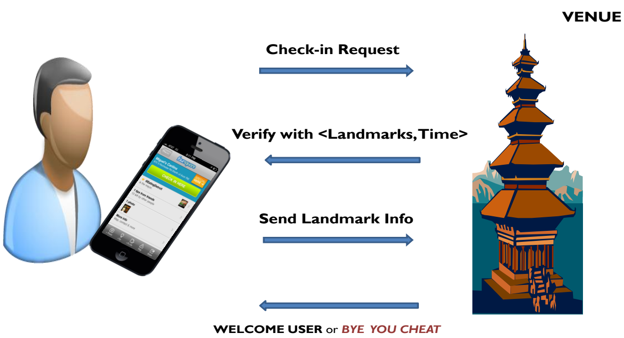


Figure 3.32: Architecture of Location Cheating Mechanism using Landmarks

As an initial idea, we can use stable virtual landmarks as identification tokens to check authenticity of a particular check-in, as shown in figure 3.32. Location verifier will use location specific landmark information to weed out the cheaters. If some of the stable landmarks from the smartphone sensor data of the users are identified, we can be sure that the user is actually in that location and not cheating.

If there is an advisory who has all possible sensor data to replay landmark information to fool such a location cheating prevention system, we argue that the cost of replaying data even for a single place will be huge. This can be quantified as the data cost in KB which has to be generated in order to cheat as shown in 3.33. This figure shows the difference between a simple GPS replay attack and landmark based replay attack in terms of data size in *BigBazaar Experiment* and *Department Experiment*. It can be seen that there are orders of difference between the cheating scheme for Landmark and GPS. In case of GPS, it is enough to just change the location using a rooted phone by giving the coordinates. However, in order to cheat in our landmark based system, it is imperative that they get the sensor data for these traces.

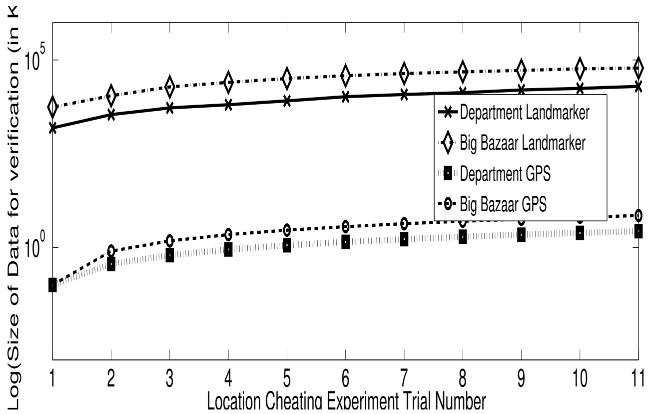


Figure 3.33: Cost Analysis of Location Cheating (Landmarker Vs GPS)

3.9 Points of Discussion

We have developed a working system which can successfully demonstrate the proof-of-concept of *Landmarker*. However, in order to deploy it in a commercial scale several further tests and fine-tuning needs to be done, we are listing some of them.

Stability of the clusters of K-means algorithm : The basic underlying clustering algorithm of our approach is k-means. So, the stability of clusters formed by k-means algorithm are very important to our approach. Several theoretical works have been done on the stability of clusters formed via k-means algorithm. [23] shows that existence of a unique minimizer cost function implies stability. If the data has multiple minimizing clusterings, then it is unstable. We observed that, in most of the cases of sensors' ensembles, we do not get multiple clusterings for the k determined by [64], as k plays an important role on the stability of the clusters of a dataset [23]. [53] shows that stability of clusters is fully determined by the behavior of the objective function which the clustering algorithm is aiming to minimize. If the objective function has a unique global minimizer, the algorithm is stable; otherwise it is unstable . Therefore, if we have an optimal minimizing objective function with properly chosen k and initial seed, we are mostly sure about the stable clusters from k-means algorithm. In this chapter, we have chosen one such procedure before applying k-means, which has given satisfactory results. But, it can be subjected to further investigation.

Effect of Phone orientation : Since our study was limited to identify the impact of changing devices, time and user, we have neglected the impact of the phone's orientation as held by the user. This assumption is fine if we assume the user's mobility pattern is unaffected by the phone's orientation. However, for example, a user might walk faster if his phone is in his pocket rather than held in his hand. But, we did not proceed further in this direction because a work by [54] has taken this into account to intelligently subtract the orientation effect on the dead-reckoning.

Experiment with more devices would help : One may ask we could have taken more devices to conduct an elaborate set of experiments. But, in this small scale experiment, we have considered the best case for devices by choosing same generation devices from same manufacturer. Even then, we have found that the device is the most prominent parameter affecting the stability of landmarks. A recent work [35] has shown that there are clusters in the sensors of different devices such as Nokia, HTC, iphone, LG handsets. As the effect of device is the most, we can have a set of landmarks belonging to each device class. This chapter strongly hints at this type higher level organization of database for future landmark supported co-operative applications.

Ensuring privacy of users : This kind of pervasive application generally suffer from privacy issues. In large scale deployment, this issue need to be tackled in detail.

Chapter 4

Cellular Radio Energy Reduction

4.1 Introduction

Mobile data traffic has been growing steadily driven by wider penetration of smartphones, and use of many network intensive applications [48]. As the adoption of cloud based services grows, the data traffic from smartphones will keep rising. Most of these apps, like mailing, online storage, or social networking apps, run as background services. The application thread wakes up intermittently to synchronize with the server.

As the landscape of smartphone based apps are evolving, several studies have investigated the traffic on smartphones [40]. One of the observations form the work is that most smartphone data transfers are small, with median size of 3KB only. Typically these small transfers are non-overlapping in time, and wake up the radio resource, like 3G network card, for every communication leading to high energy usage [68]. Several prior works have reported the power characteristics of 3G radio [21, 36]. The key observation is that every time a 3G radio transitions from sleep ($IDLE$) to wake up state ($CELL\_DCH$), the card will continue to stay up after the transmission for a threshold duration, called tail time ($CELL\_FACH$), thereby wasting energy. Also the transition energy from each state is high. Under-utilizing the bandwidth in $CELL\_DCH$ state is also detrimental towards the goal of energy optimization.

Reducing tail time energy wastage has been addressed primarily by (a) dynamic adjustment of the tail time timer by observing traffic patterns [70, 51], and (b) using the tail time for transmissions [55, 20]. In order to fill the tail time with transmissions, the classical approach is to aggregate packets from a single application either by delaying packets [21], or reorganizing computation and communication leading to higher batching efficiency [82, 87]. However, our observation is that aggregating packets from multiple applications can leverage even higher benefits. Second, the packets are typically of small size, specially for the background services. Hence batching packets from a single app may be insufficient to utilize the high bandwidth access links in emerging cellular networks, like LTE. Interleaving packets from multiple apps may lead to better utilization of the radio resource, specially in high bandwidth $CELL\_DCH$ state.

This is important to note that as the access links are reaching higher bandwidth, the bottleneck in the network is pushed back to the core. A request packet from an app, which waits for response due to a slow or congested link in the backbone, will force the radio to transition to idle state, and will be turned on when the response comes back. Instead the gap can be filled by increasing the efficiency of batching, preferably by interleaving packets from different apps.

In this chapter, our goal is to maximize the radio resource utilization in $CELL\_DCH$ state by aggregating packets across multiple apps. Higher packet aggregation requires that requests from different apps may need to be delayed to synchronize the transmissions. Each app is assigned a delay limit within which a packet transmission request from that app must be serviced. The delay limit for an app is calculated based on the user's interaction with the app. Foreground app is assigned the least delay budget (equal to the switching time from $IDLE$ to $CELL\_DCH$), while delay budgets of background apps are proportional to the time of user's last interaction with the app. This should minimally effect the user experience. Also background apps are more delay tolerant, as supported by Huang et al. in [47]. While modeling the overall traffic from a smartphone we apply variable delay limits to different apps.

We propose three deadline-aware online approaches for batching packets from multiple background apps and one foreground app, which represents a typical smartphone usage scenario. The goal of the algorithms is to maximally utilize the access link in $CELL\_DCH$ state by assigning app-specific variable transmission delay. We show that significant energy gains can be achieved by efficient utilization of the available bandwidth in $CELL\_DCH$ mode by batching packets across foreground and background applications without impacting the user experience. We use the standard energy model of 3G network card [18] to compare the energy gains by the online scheduling applied in Fast Dormancy, and Fast Dormancy with 5 sec tail time operating mode of the card. Through simulation experiments, we show that around 40% network card energy can be saved compared to transmission without any aggregation. We also present an offline scheduling approach for benchmarking and to compare performance of the proposed online algorithms. We have also compared our strategy with other competing strategies with gaining satisfactory results. Moreover, we have also gained around 20% in network energy savings also in real world traces also.

The rest of the chapter is organized as follows. In Section 4.2 we describe the models used in defining the problem. In Section 4.3 we present the problem formally followed by three strategies for online batch scheduling. Section 4.4 shows the results of simulation experiments and Section 4.5 shows results on real network traces.

4.2 Models and Assumptions

In this section, we describe various models that we use in formulating the problem and developing the scheduling algorithms. We start with the network model and explain the intuition behind the proposed approach. Next we characterize different apps that run on the smartphones, based on their network activity pattern. This is followed by a detailed description of the traffic model including the list of parameters and finally we conclude this section illustrating the energy model considered in this chapter.

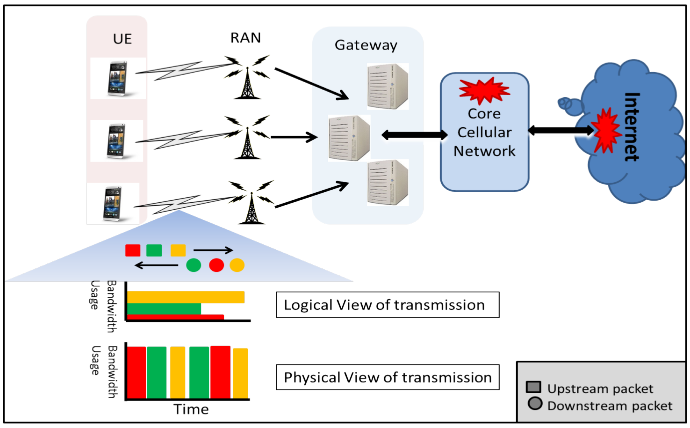


Figure 4.1: Simplified topology illustrating packet transfer over high bandwidth access link in cellular network. Multiple small packets can be aggregated during upload, which can reduce the need to switch off the 3G network card.

4.2.1 Network Model

First we describe the network model considered in this chapter along with its limitations and introduce the key concept behind the proposed methodology. We assume a cellular network model, similar to 3G UMTS discussed in Section 2.4.2, where the access links are supposed to be of high bandwidth [46]. Fig 4.1 illustrates the network model through a simplified diagram. As the access links are becoming faster, the bottleneck in the network is shifting towards the core cellular network, and the Internet backbone. As a result, although an upstream request packet from the smartphone to the Radio Access Network (RAN) may have low latency, the response packet corresponding to the request may take long to come back. If the gap between the request and response packets are longer than a few seconds, the radio network card will be put to sleep to save energy wastage on the smartphone. However, the problem is, for a 3G network card, each state transition wastes a significant amount of energy [69].

In this work, we propose that orchestrating the packet transmission, as shown in Fig 4.1, will prevent a state transition by scheduling other request packets during the interval between a request-response sequence. For a single app's request-response sequence, without card state transition, the goodput of the access link is the total bytes exchanged over the request-response completion time. Thus the goodput is much lower than the throughput of the access link. However, if packets from other apps are sent during the gap, then the goodput increases, as illustrated in the logical view of transmission. In the figure, the upstream and downstream packets from different apps are marked in different colors. The physical view of transmission shows the packet exchange at the physical layer of the card, while the logical view illustrates the idea of co-scheduling of the packets from different applications to increase the overall goodput by maximizing the access link bandwidth utilization.

4.2.2 Application Model

Mobile applications can be broadly categorized in following way, based on their unique network activity or usage pattern.

Background Services (A_B) : In general, these services run in the background on smart phones. These services sync with their corresponding servers to fetch periodic data updates. Some of the classic examples of such services include news service, weather updates, and software updater. On the other hand, some foreground applications spawn different background services to provide more engaging experience. For example, Facebook application spawns background processes for contact syncing service and message notification service, and Gmail app triggers mail notification service.

Data consumption and network request pattern of background services (A_B) generally follow a specific pattern, e.g. CNN news app pulls news snippets periodically.

Foreground Applications (A_F) : In general, this kind of application has a front-end and requires active user interaction to get and provide some meaningful service. Streaming apps like *Youtube*, gaming apps like *Free Online Games*, blog reading app like *Flipboard*, *Dolphin* browser app are some of the examples of A_F with network traffic.

Data consumption pattern of foreground service (A_F) heavily depends upon the behavior and interaction patterns of the users. Even user interaction pattern varies from application to application; e.g. mobile browser activity generally follows random pattern [76][62], whereas email app activity follows a power-law with $\alpha = 1$ [22]. We model these two distinct kinds of foreground applications with two different distributions (a) random ($A_F - Random$) and (b) power law

$(A_F - Normal)$ to accurately reflect the intrinsic difference.

4.2.3 Traffic Model

Network activity of a mobile application depends upon the sync timing (automatically set by the developer for an application like CNN news app or chosen by the user for an application like facebook), user interaction pattern, and data consumption during syncing or interaction. In order to synthetically generate the traffic trace, a proper specification of the parameters is essential. In the following, we illustrate the parameters regulating the properties of the network traffic for foreground and background applications.

Application Sync Timing (Ψ) : This parameter decides the data sync intervals for background services. In our experiments, we have considered two sync interval threshold for the experiments : 15 minutes and 30 minutes, as these are the typical default settings for mobile apps. For A_B we have considered a constant Ψ value, while for A_F this parameter does not apply.

User Interaction Timing (Υ) : This parameter emulates user interaction pattern with the foreground application. As [22] suggests that human triggered decision mainly follows the pattern of high activity for a small duration followed by inactivity for a longer duration, viz. a power-law distribution, therefore we emulate foreground application activity by choosing Υ as a power-law distribution with $\alpha = 1$. For background service A_B, Υ is constant since there is no user interaction.

We have also considered foreground applications where user's interaction cannot be modeled following a well-known distribution. Browsing behavior on smartphones has been reported recently to follow such random user interaction [76][62]. To mimic this behavior, Υ value is selected randomly within a range. For background services A_B, we choose a constant Υ value.

Data Transmission Size (Λ) : When an application gets hold of the network resource for a particular communication, amount of data transmission depends upon the application type. For applications, like YouTube it can be up to 5MB in a session [37] or for a background weather app, it can be around 3KB, or for a content heavy app like flickr it can be up to 128KB [37]. This parameter models the asymmetric behavior across application in transmitting data during a network session.

Bandwidth Demand (Δ) : The bandwidth demands for each application is also considered. We choose a higher value for streaming apps and lower value for background apps.

Flexibility (Φ) : This parameter denotes the maximum allowed delay limit for an application specific request. Φ takes lower value for packets from foreground app, A_F, and higher value for the packets from background apps, A_B.

We have used the energy model illustrated in Section 2.4.3 for the experiments. Moreover, we have adopted two different strategies [70][17] in 3G data transmission, namely *Fast Dormancy (FD)* and *Fast Dormancy with 5 second Tail Timer (TT)*. In FD, tail-time (t) is considered zero and for TT, tail-time (t) is considered as five seconds. We have assumed that, in FD, the demotion from $CELL\_DCH$ to $IDLE$ state happens instantly bypassing $CELL\_FACH$ state without losing any energy.

4.3 Scheduling Strategies

In this section, we present a suite of algorithms for scheduling network requests coming from different applications to optimize network device utilization. While scheduling the requests, two important points need to be taken care of: (a) two requests from the same application cannot be triggered simultaneously (b) total bandwidth consumption by all the scheduled requests should be bounded by the channel bandwidth. All the requests P_{ij} following these two constraints are termed as *compatible request*.

First we present three online algorithms. The design of each algorithm is geared towards a different class of foreground traffic, like interactive apps, browsing or streaming. Hence an implementation of the scheduler can switch to the appropriate algorithm to optimize the gains. Details of the implementation are beyond the scope of this work. Finally, we present an *Offline Scheduler* algorithm, which is a heuristic for co-scheduling with a priori knowledge of traffic. *Offline Scheduler* serves as a benchmark to compare the online versions.

4.3.1 Problem Description

Key concept: Let us assume that a set of applications, running in parallel, request for network resource intermittently. Moreover, depending on application (background/foreground), a flexibility or slack time is allowed to schedule each packet. The task of the scheduler is to correctly schedule each application request, exploiting the flexibility provided by the slack time, such that the maximum number of requests can be served simultaneously; this will eventually maximize the network card usage.
Formally, let us denote the j^{th} network request from the application A_i as P_{ij}. The arrival (release) time of the request P_{ij} is designated as r_{ij} and the actual triggering time of the request by the scheduler is x_{ij}. Each request P_{ij} has an allowed slack time (flexibility) f_{ij} which means that request P_{ij} can be scheduled latest by $r_{ij} + f_{ij}$ to avoid any deadline miss. Service duration of request P_{ij} is denoted as d_{ij} and bandwidth required to provide this service is represented as b_{ij}.

The communication channel has a finite bandwidth B which essentially implies that at any time instant, total bandwidth consumption of all the scheduled requests must be less than B. All the scheduled requests from same application must be sequential in nature; that means until a request P_{ij} is served, P_{ik} (where $k > j$) can not be scheduled, i.e. $x_{ik} \geq x_{ij} + d_{ij}$. Our goal is to batch and schedule requests from all running network applications in such a manner that network resource is maximally utilized when network radio is on so that radio idle time is maximized.

4.3.2 Complexity of the Problem

A simplified case of the above problem arises under infinite bandwidth assumption. An optimal solution to this problem can be derived, as shown later. However, imposing the bandwidth constraint turns the problem into a variant of bin packing optimization which is NP-hard.
The reduction is achieved through the following transformation of NP-hard version of bin-packing [42] optimization problem to this problem. Let us partition

the entire traffic duration into unit sized time intervals and also modify the bandwidth demand b_{ij} of request P_{ij} to the fraction b_{ij}/B $(= b'_{ij})$, where B is the bandwidth limit of the device. Now, we can map the items of bin-packing problem to the network requests and unit sized bins to the unit bandwidth demand. Hence, the goal of bin packing to find the minimum number of bins leads to our aim to achieve minimum duration of network usage.

4.3.3 Scheduling Algorithms

In this section, we present three scheduling algorithms - (a) *Lazy Scheduling*, (b) *Early Scheduling* (c) *Balanced Scheduling*, each of which can be selectively triggered by an implementation of traffic aware scheduler, as explained before. Implementation of traffic aware scheduling is beyond the scope of this work.
We begin with the online algorithms. The key elements behind these three online algorithms are two queues - (a) wait queue and (b) run queue. Wait queue maintains transmission requests sorted according to their deadlines (for application A_i and request j, deadline $r_{ij} + f_{ij}$). On arrival, a packet is placed in the wait queue. The run queue stores the set of requests being serviced by the network device; empty run queue implies that the device is not in $CELL\_DCH$ mode. Three algorithms introduced next differs functionally with respect to the placement of the requests from the wait queue to the run queue.

Lazy Scheduling

The wait queue sends a trigger to the scheduler when it has at least one request (say P_{ij} from application A_i) reaching its deadline[1]. On receiving the trigger, scheduler checks if the run queue is empty (i.e. network card is not in $CELL\_DCH$ mode). If empty, it removes the request P_{ij} from the wait queue, along with the sequence of successive compatible requests appearing behind P_{ij} and places all these requests in the run queue. On the other hand, if the run queue is not empty, it waits till it empties.

[1]The trigger is sent 2 seconds prior to the deadline in order to allow the radio to switch from $IDLE$ to $CELL\_DCH$ state.

Algorithm 4.1: Lazy Scheduling Algorithm

```
Input: Arrival of Requests;
Output: Scheduling of every request
Intialization: clock=1;
while True do
    if packet arrives then
        put it in Wait Queue;
    end
    if Trigger_From_Wait_Queue == True then
        if Run_Queue == Empty then
            Remove request(s) from Wait Queue;
            Put it in Run Queue;
            Trigger_From_Wait_Queue = False;
        end
    end
    clock++;
end
```

The two other scheduling algorithms function similar to *lazy scheduler*, except when the run queue is not empty.

Early Scheduling

If the run queue is not empty, scheduler checks wait queue to find any compatible request which can be scheduled. If it finds any such request in the wait queue, scheduler removes it from wait queue and places it in the run queue.

Algorithm 4.2: Early Scheduling Algorithm

```
if Run_Queue != Empty then
    Check Wait Queue for Compatible Request;
    Remove it from Wait Queue;
    Put it in Run Queue;
end
```

Balanced Scheduling

If there are some requests in the run queue, the scheduler first checks their compatibility in a way similar to *Early Scheduler*. If a request is present, *Balanced Scheduling* strategy computes a benefit function to decide placement of the compatible requests in the run queue. The key observation is that there are two competing factors bandwidth wastage and deadline miss which should be accommodated while defining F. More formally, we can combine them additively to reach a balance between the two factors. Therefore F can be defined as follows.

$$F = \beta \cdot Bandwidth\_wastage + (1 - \beta) \cdot Experience\_user \tag{4.1}$$

where β is a normalizing constant. Different elements of the equation are explained in detail below.

A. Bandwidth Wastage Let the maximum available bandwidth be B and let there be n requests denoted by P_i[2], $i = 1 \cdots n$, each starting at x_i time, running for a duration of d_i and requiring a bandwidth b_i. Then Bandwidth Wastage (BW) is the total available bandwidth during the active period of at least one of the requests minus the bandwidth utilized by these requests. It can be defined as,

$$BW = B \times T - \sum_{i=1}^{n} b_i \times d_i, \quad T = Max(x_i + d_i) - Min(x_j), \quad \forall (i, j)... \tag{4.2}$$

Assume a packet transmission is delayed using the available slack duration, which allows the scheduler to aggregate multiple packets together for transmission. If delaying makes better utilization of bandwidth compared to sending it as soon as it arrives, then the term $BW_1 - BW_2$ gives the bandwidth utilization efficiency, where BW_1 and BW_2 denotes bandwidth wastage by delaying and not delaying packets respectively. By normalizing the term, we get the term,

$$Bandwidth\_wastage = \frac{BW_1 - BW_2}{Max(BW_1, BW_2)} \tag{4.3}$$

B. User Experience Three factors need to be considered while optimizing user experience - (a) a deadline $(r_{ij} + f_{ij})$ should not be missed and (b) process the request not too late but (c) delay the request if it helps in aggregating a number of application requests together. Let us say the request P_{ij} appears at time instant r_{ij} and has slack duration f_{ij}. Let us also assume that all the network requests presently in the run

[2] without loss of generality we are dropping the j subscript for ease of understanding

queue will be served completely by E (finish time). We assume that if E is very close to r_{ij}, it is prudent to delay the request and wait for other requests to appear for better parallelization - the probability of which decreases as the value of E - r_{ij} increases. Consequently we formalize through the following formula,

$$Experience\_user = \frac{E - \frac{r_{ij}+(r_{ij}+f_{ij})}{2}}{Max(E, \frac{r_{ij}+(r_{ij}+f_{ij})}{2})} \tag{4.4}$$

C. Normalization factor (β) In the next set of empirical measurements, we carefully looked at the impact of each term in Eqn. 4.1 in determining the value of F such that we can select a value for the normalization parameter, β. The value of the normalization parameter, β, should be such that across all traces the impact of each term is roughly the same. With that aim we check the average number of times each term contributes in making the value of F positive. Since at $\beta = 0.9$, the contributions of the two terms are roughly equal, therefore, β value is set to 0.9.

Finally, for different traffic characteristics, one may need to give priority to either bandwidth utilization or deadline miss. This is incorporated by introducing a scaling parameter, α, to skew the importance of the two terms. It can be conceptualized with the help of the function F described below. The α value should be ideally chosen such that the bandwidth wastage and deadline can be optimally traded off.

$$F = \alpha \cdot \beta \cdot Bandwidth\_wastage + (1 - \alpha) \cdot (1 - \beta) \cdot Experience\_user \tag{4.5}$$

where α varies between 0 and 1.

Algorithm 4.3: Balanced Scheduling Algorithm

```
if Compatible Request found in Wait Queue then
    Evaluate F function;
    if F>0 then
        Remove it from Wait Queue;
        Put it in Run Queue;
    end
end
```

Observe that there are two factors involved in F and both ranges from -1 to 1. When F becomes positive for any request then it is removed from wait queue and placed in the run queue. In this way, this algorithm ensures that only those compatible requests are scheduled which ensures minimum bandwidth wastage and low deadline miss probability.

4.3.4 Offline Scheduler

In order to evaluate the performance of the aforementioned online algorithms, one needs to have a proper benchmark solution. Here we outline an approach which schedules a sequence of requests in a semi-optimal way such that the maximum number of requests can be served in parallel. It is assumed that the global information about the applications (request arrival time, service time, bandwidth requirement) are available a priori; in that aspect, this scheduling approach is offline. This algorithm is split into two components; first part drops the bandwidth constraint and only aims to maximize the number of applications which can be served in parallel adhering to the allowed slack time assigned to each application. Here, the only constraint we have is that two requests from the same application can not be scheduled at the same time. Subsequently, we incorporate the bandwidth constraint and finalize the solution.
Let us assume that the $n \times m$ (where n is the number of applications running in parallel and every application issues m requests) number of requests $P_{11}, P_{12}, ..., P_{nm}$ are sorted according to their arrival time. Here we try to batch different requests together for final scheduling by adjusting the allowed slack time. First we demonstrate the principle behind the approach for only two requests A_i and A_k and then generalize the solution for $n \times m$ requests. The concept here is to identify that whether two requests P_{ij} and P_{kl} (with arrival time r_{ij}, r_{kl}, allowed slack time f_{ij}, f_{kl}, service time d_{ij}, d_{kl} respectively), coming from two different applications A_i and A_k ($r_{kl} > r_{ij}$) can indeed be served in parallel. This problem essentially boils down to the problem of finding overlap in the service period between two requests P_{ij} and P_{kl}. The following three different cases may arise, for each of which offline scheduler takes appropriate action to properly schedule the request.

Mathematically, the action taken by the scheduler for scheduling two requests P_{ij} and P_{kl} is designated by the operator $\oplus$ and the outcome of the scheduler (which is essentially the actual triggering time x_{ij} and x_{kl} of the requests P_{ij} and P_{kl} respectively) is stored in function $BS(P_{ij}, P_{kl})$.

Case (1): No overlap is possible: This case can be identified by the condition

$$r_{ij} + f_{ij} + d_{ij} < r_{kl} \tag{4.6}$$

Action $\oplus$: This is the worst case. Since no overlap is possible, hence scheduler will trigger those requests as soon as they arrive (see Fig. 4.2).

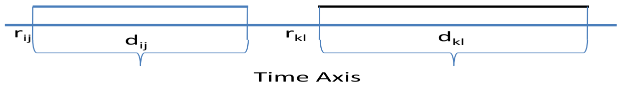


Figure 4.2: Schedule requests as they arrive when overlapping is not possible.

Case (2): Full overlap is possible: This is the best case. In this case, one network request can completely imbibe the other request within it and can be identified by the following condition

$$r_{ij} + f_{ij} + d_{ij} > r_{kl} \tag{4.7}$$

and

$$f_{ij} > (r_{kl} - r_{ij}) \tag{4.8}$$

Action ⊕ : In this case, starting time of one request (say P_{ij}) can be shifted forward by the amount $(r_{kl} - r_{ij})$ such that both the requests P_{ij} and P_{kl} can be scheduled at the same time.

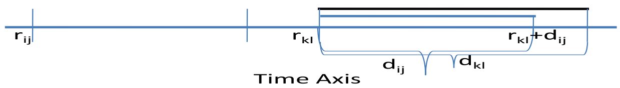


Figure 4.3: When slack of first request is more than the gap between arrival of two requests then requests can be fully overlapped.

Case (3): Partial overlap is possible: This case can be identified by the condition

$$r_{ij} + f_{ij} + d_{ij} > r_{kl} \tag{4.9}$$

and

$$f_{ij} < (r_{kl} - r_{ij}) \tag{4.10}$$

Action ⊕: In this case, the algorithm shifts the starting time of P_{ij} by f_{ij} (see Fig. 4.4) which would result in partial overlapped schedule.

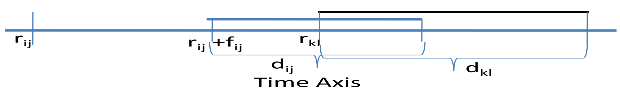


Figure 4.4: When slack of first request is less than the gap between arrival of two requests then requests can be partially overlapped.

Generalization for $n \times m$ requests
In a nutshell, $BS(P_{ij}, P_{kl}) = BS(P_{ij}) \oplus BS(P_{kl})$ provides us the optimal scheduling of two requests P_{ij} and P_{kl} based on the above three cases. Capitalizing on this observation, we generalize the scheduling action $\oplus$ for $n \times m$ requests $X_1, X_2, ..., X_{nm}$ using the optimal substructure property, which results in the following dynamic programming formulation

$$BS(X_1, X_2, ..., X_{nm}) = min(BS(X_1, ..., X_q) \oplus BS(X_{(q+1)}...X_{nm})) \quad (4.11)$$

where q varies from 1 to $n \times m - 1$.

We can easily find a suitable q minimizing the above function which eventually provides us the optimal scheduling time for all the $n \times m$ requests.
Incorporating bandwidth constraint
The formulation described above provides us an optimal schedule for a set of requests S_i at time t_i. Next, we select a (possibly proper) subset of requests S_u^i from this set S_i respecting the bandwidth constraint, which essentially means that the total bandwidth consumption of the requests in set S_u^i is limited to the total available bandwidth B. The leftover requests $(S_i - S_u^i)$, if any, will be suitably shifted (forward/backward) to a point where bandwidth violation can be avoided.

4.4 Evaluation

In this section, we evaluate the performance of batch scheduling techniques using simulation experiments. We describe the experimental setup, followed by analysis of the results.

4.4.1 Experimental Setup

The simulation experiments are driven by synthetic traces representing smartphone traffic. The traces are generated using the models defined in Section 4.2.3. We generate traffic from one foreground app which the user is using, and multiple background

Algorithm 4.4: Offline Scheduling Algorithm

Input: Number of requests:n;

Requests[1...n];

Output: Scheduling of every request

Intialization: size=1;

while $size < n$ **do**

  $start$ =1;

  end = n - size;

  while $start < end$ **do**

    Compute BS($Request[start]$,...,$Request[start + size - 1]$);

    if *violates bandwidth constraint* **then**

      Remove violation;

    end

    $start$++;

  end

  size++;

end

services. In order to mimic different usage scenarios in the foreground, such as interactive app, streaming app, and browsing, we generate traffic from three different combinations of apps.

Table 4.1: Foreground and Background App Parameters used in Synthetic Trace Generation

App Type	Sync Time (s) Ψ	UI Time (s) Υ	Data Tx Size (KB) Λ	Bandwidth Demand (KBps) Δ	Slack Duration (s) Φ
A_B	900,1800	5,10,15	3,5 [40]	10	5-7
A_F-Normal	NA	Power-law [22]	3,5 [37]	5,15,40	2
A_F-Random	NA	2-20 [76]	3-50 [37]	20	2

We set the total capacity of the access link to 50 KBps in $CELL\_DCH$ state. We model two different types of foreground apps, $A_F - Normal$ and $A_F - Random$ (see

Section 4.2.2) using the parameter values defined in Table 4.1. Background apps are assigned variable slack time values - 5, 10 and 15 seconds. Data transfer size (or packet size) for each app is also varied within a range as shown in the Table 4.1.

Different foreground user activity is captured by changing the foreground app bandwidth demands. The bandwidth demand for interactive gaming app is 5 KBps, for streaming app 40 KBps, and for browsing app 15 KBps. **Gaming** and **Streaming** scenarios comprise of one foreground app, modeled as $A_F - Normal$ traffic and bandwidth demand from Table 4.1, and three background apps. **Browsing** scenario comprises of one foreground app, modeled as $A_F - Random$ traffic with 15 KBps bandwidth requirement, and three background apps. We also evaluated the performance of foreground app in presence of an increasing number of background apps. Traffic trace for each scenario is of length 1 hr.

4.4.2 Evaluation Metrics

We use multiple metrics to analyze the energy performance and the impact on user experience.

Energy Consumption per KB (E_{KB}): This metric captures energy spent to transmit one KiloByte of data over the network interface. It considers ramp up energy, transmission energy and tail energy.

Average Slack Duration (μ_{SD}): It is defined as the average slack across all requests. Mathematically it can be expressed as $\frac{1}{n} \times \sum_{i=1}^{n} slack_i$. μ_{SD} is used to understand the slack experienced by the packets from foreground traffic.

Deadline Miss Frequency (D_F): It is expressed as, $\frac{Total\_Number\_of\_Deadline\_Miss}{Total\_Number\_of\_Request}$. This is also used to capture the impact on foreground traffic.

Two other metrics are used to gain insight into the source of energy gains. These are,

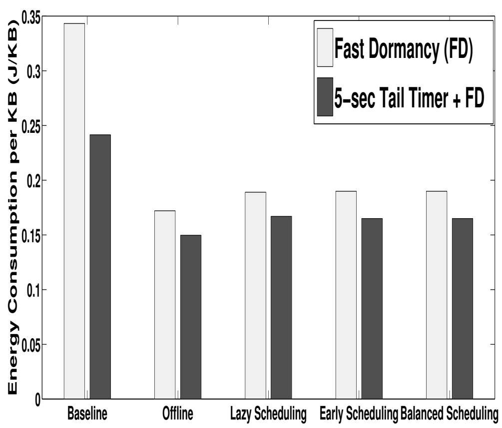
Percentage of Radio On Time (R_{ON}): This metric captures the radio on time as a percentage of total data transmission duration. We will use this metric to reason about the energy savings.

Switching Frequency (S_F): It is defined as number of switching ($IDLE \rightarrow CELL\_DCH$ and $CELL\_DCH/CELL\_FACH \rightarrow IDLE$) per unit time.

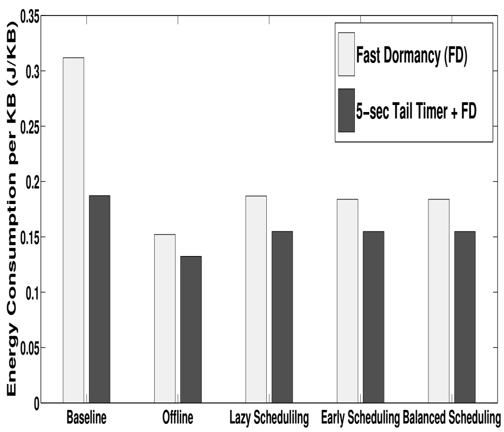
Finally, we also define a ***Baseline Scheme*** for data transmission, where a network packet is serviced immediately. If the radio is on, i.e. in $CELL\_DCH$ state, then the packet transmission is immediate. If the radio is in sleep ($IDLE$), then 2 seconds are required to switch to on state before transmission triggers. The *Baseline Scheme* provides a benchmark for understanding the gains over standard scheduling.

4.4.3 Energy Gains

We report the energy savings under different scenarios in terms of E_{KB}. First, we show the results for Gaming and Streaming scenarios, where the foreground app is modeled as $A_F - Normal$. Fig. 4.5(a) and Fig. 4.5(b) show the energy consumed to transmit per KiloByte of data across all the entire foreground and background traffic for Gaming and Streaming scenarios respectively. We compare three online schemes against Baseline Scheme and Offline Scheduling. We also show the results for the two modes - Fast Dormancy (FD) and Fast Dormancy with 5 sec tail timer (TT).



(a) Gaming Scenario



(b) Streaming Scenario

Figure 4.5: Comparison of Energy Consumption per KB of data transmission for different schemes in Gaming and Streaming scenario.

For the FD mode, Baseline consumes 0.34 J/KB for Gaming, and 0.32 J/KB for Streaming, whereas Offline consumes 0.17 J/KB and 0.15 J/KB respectively. Baseline and Offline depict the worst case and best case in terms of energy consumption. The online techniques are comparable in their energy consumption requiring between 0.18 J/KB to 0.2 J/KB. Although an ideal gain of 50% can be achieved, as shown by Offline scheduling, the online versions can reach 40% energy gain compared to the Baseline scheme. Note that for the Streaming scenario, power consumption is marginally lower than that of the Gaming scenario. Since the total data transmitted is much higher in streaming case, radio on time is utilized more effectively.

In the TT mode, the energy consumption is reduced further. For the Baseline case alone, there is 30% less energy requirement compared to the FD mode. The gain can be attributed to the reduction in the number of card state transition. Since a state transition from $IDLE$ to $CELL\_DCH$ consumes significant amount of energy, by saving the transition, Baseline TT mode gains over Baseline FD mode. However, the gains for the online, as well as, offline techniques are not as significant in TT mode. Tail time reduces total energy consumption by reducing switching. As the gain in TT mode is not significant compared to FD mode in our schemes, we can conclude that our schemes could not leverage the tail time to reduce enough number of switching. In Section 4.4.5 we take an in-depth look in switching frequency data. Still, the online techniques perform 10% to 30% better compared to Baseline scheme. In the next

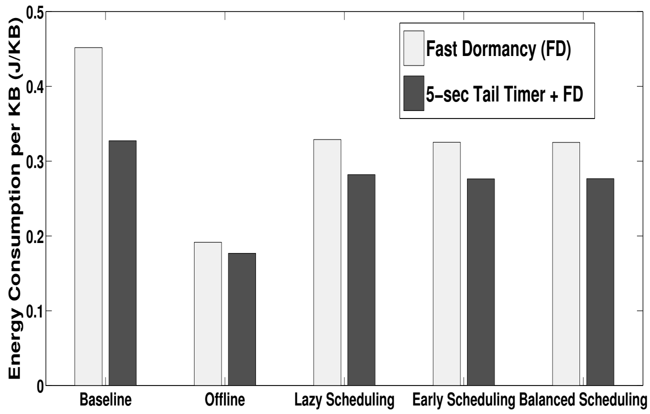


Figure 4.6: Comparison of Energy Consumption for different schemes in Browsing scenario.

experiment, we evaluate the energy consumption for the Browsing scenario, where the foreground browsing traffic is modeled as $A_F - Random$. Fig. 4.6 shows the results for energy consumption per KB of data transmission compared across all the scheduling schemes. Offline scheme shows 60% savings over Baseline scheme in FD mode. In FD mode, the online schemes save around 25% compared to Baseline. In TT mode, Baseline scheme again shows energy savings of 25% by reducing switching frequency, while the gains for the online schemes are between 5% − 10%. The key observation is that even when the foreground traffic pattern is randomly distributed, the online scheduling scheme can aggregate packets effectively to reduce energy consumption.

4.4.4 User Experience

We evaluate the user experience with respect to the impact of online batch scheduling on the foreground app traffic. We measure the slack suffered by the foreground traffic, and the fraction of packets that miss their deadlines, which will impact responsiveness of interactive applications. In this experiment, we increase the number of background app to 6. We also use the Streaming scenario which has the highest bandwidth demand, thereby presenting the most demanding case to handle during scheduling.

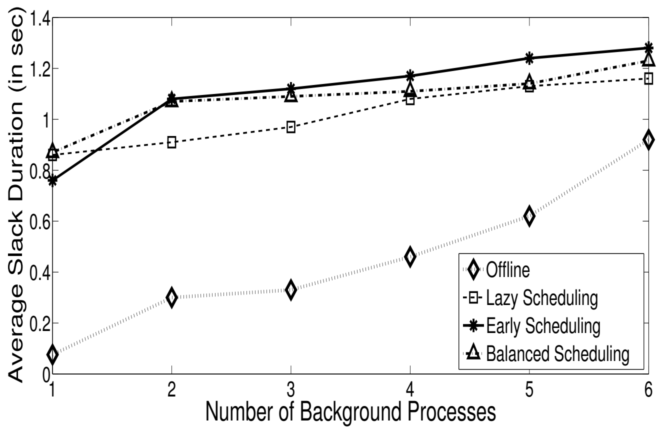


Figure 4.7: Average slack incurred by foreground app packets with different schemes in presence of varying number of background apps.

Fig. 4.7 shows the average slack incurred by the foreground packets. The Offline scheme confirms that it is possible to maintain a low average slack for the packets. The online schemes incur ≈ 0.8 second average slack, which increases to 1.2 second slack with 6 background apps. Even the worst value of standard deviation for the slack is 0.14 second indicating none of the packet incur a high slack. Though performance of online algorithms are little bit low compared to offline algorithm, degradation of performance with increasing load is very graceful.

The next experiment measures the number of foreground packets which missed their deadlines. Fig. 4.8 shows the fraction of foreground packets that missed their deadlines. Surprisingly, performance of *Lazy Scheduling* is better than that of *Early Scheduling* and *Balanced Scheduling*. Since *Early Scheduler*, as well as *Balanced Scheduler*, may schedule some compatible requests when there is any request in the run queue, therefore, any request which is not compatible and waiting in wait queue may remain incompatible and kept waiting, thereby missing its deadline. By design *Lazy scheduling* avoids this scenario and schedules requests in deadline priority basis, thereby reducing the number of deadline misses. *Balanced Scheduling* shows mixed characteristics of

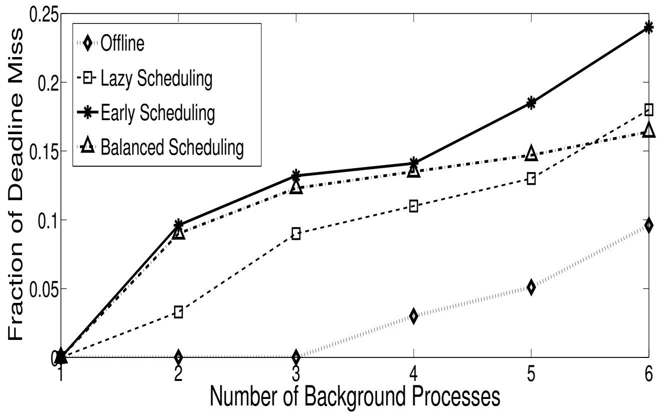
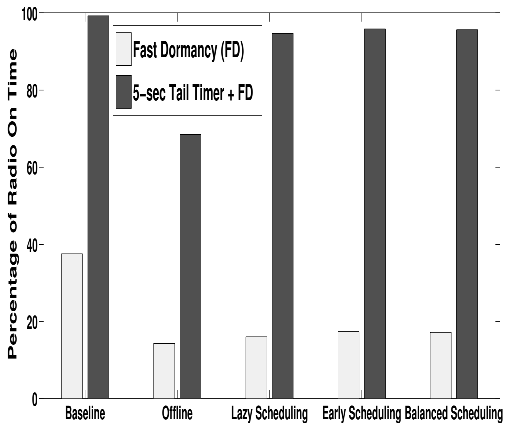


Figure 4.8: Fraction of overall packets from foreground app missing deadline with different schemes in presence of varying number of background apps.

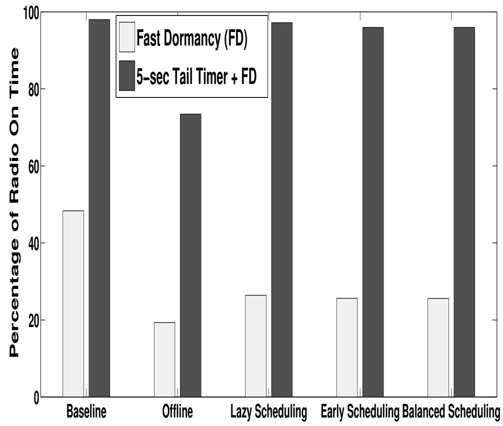
Early Scheduling and *Lazy Scheduling* as it does not schedule a compatible request unconditionally even when run queue is not empty (radio is on).

4.4.5 Insights into the Gains

In this section, we give an in-depth view of the reasons for the energy gains by measuring the radio on time, and switching frequency of the card.



(a) Streaming scenario



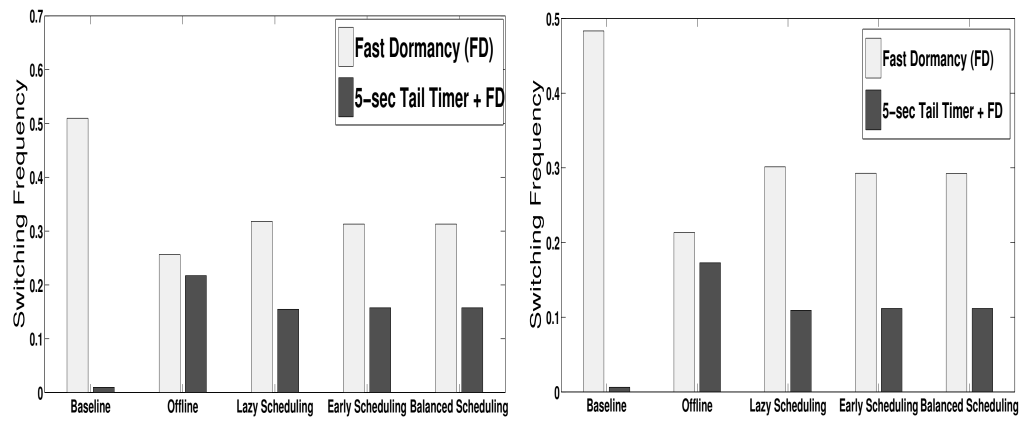
(b) Browsing in presence of 6 background apps

Figure 4.9: Percentage of duration when the radio was ON for the two energy models (FD and TT) and for different schemes in Streaming and Browsing scenarios.

In the first set of experiments, we measure the radio on time for each scheme. Fig. 4.9(a) shows the percentage of radio on time with different energy models (FD and TT) for the Streaming scenario. Experiment with FD energy model shows radio on time is $\approx$ 38% for Baseline scheme while it is around 15% for the Offline algorithm. Radio on time percentage for online algorithms ranges from 17%-19%. So in FD energy model, in terms of radio on time percentage, online algorithms perform far better than Baseline scheme.

With TT energy model, percentage of radio on time in $CELL\_DCH$ mode remains same for all the schemes. However, radio remains in $CELL\_FACH$ mode for a significant amount of time for every scheme. Result shows that Baseline scheme stays in $CELL\_FACH$ mode around 60% to 62% of time. Offline scheme stays in $CELL\_FACH$ mode for around 42% of time, whereas online algorithms stay in $CELL\_FACH$ mode for 80% of time. So, radio on time including both $CELL\_DCH$ and $CELL\_FACH$ modes are almost comparable for Baseline and all the online schemes. However, significantly low $CELL\_DCH$ time of online algorithms compared to Baseline scheme justifies low energy consumption of the online algorithms.

Fig. 4.9(b) shows the radio on time percentage with different energy model (FD and TT) and with various schemes for Browsing scenario. Here also we see similar characteristics of different schemes as in Streaming scenario. However, as Streaming utilizes available bandwidth more effectively than Browsing, radio on time percentage for Streaming is less than Browsing. In the next experiment, we study the impact



(a) Streaming scenario

(b) Browsing in presence of 6 background apps

Figure 4.10: Switching frequency comparison with two energy models (FD and TT) and for different schemes in Streaming and Browsing scenarios.

of the number of state transitions by the network card. Fig. 4.10 shows the switching frequency with different energy models (FD and TT) and with various schemes.

Experiments with FD energy model show, switching frequency is 0.48-0.5 for baseline scheme while it is around 0.22-0.25 for offline algorithm[3]. However, it is little bit higher ≈ 0.32 for all online schemes.

With TT energy model, switching frequency reduces drastically to 0.01 for baseline scheme. However, switching frequency of offline scheme does not change much compared to FD mode as $CELL\_DCH$ mode in offline algorithm is well separated. So even with tail timer on, Offline algorithm reduces switching marginally. Moreover, though all online algorithms stay in $CELL\_FACH$ mode for a significant fraction of overall time, it reduces switching only by 50%-70%. Similar to offline algorithm, $CELL\_DCH$ mode in online algorithms are also well separated, however, not as good as in the Offline case. So energy savings in proposed online algorithms, specially in TT mode cannot be attributed to lower switching frequency.

4.4.6 Competing Scheduling Techniques

As we find out that balanced scheduling works well for different scenarios, we compare it with other three different techniques - TailEnder, Tail Optimization Protocol (TOP), and Performance-aware Energy Scheduler (PerES). We present implementation details of these techniques.

TailEnder: $TailEnder$ [20] uses threshold based tail time prediction by considering deadlines of packets of an application. In principle, it delays packets as-long-as-possible without affecting user experience. We extend $TailEnder$ [20] to prevent tail energy wastage across multiple applications, instead of original design aimed at single application. This naive implementation introduces per application separate queues which are serviced according to the $TailEnder$ heuristics using a simple round robin order.

PerES: Performance-aware Energy Scheduler or PerES models cross application energy-delay tradeoff as an optimization problem and applies Lyapunov optimization framework. It assumes that the wireless channel bandwidth is variable.

The implementation depends on choice of several parameters. The chosen parameter values, based on [84], are as follows. δ value of SVA is 0.001, application preference weights are 1/10 (for foreground application) and 1/100 (for background application), θ value is taken as 10, wireless signal as taken uniform randomly from the range of

[3]Note that the average number of requests per second is 0.83.

$-50\ dBm$ to $-110\ dBm$, and application details are according to Table 4.1.

TOP: Tail Optimization Protocol (TOP) reduces tail energy wastage by predicting the application behavior [70]. Since te paper claims that tail time can be predicted with 60% accuracy, therefore, for 60% of the uniformly randomly chosen application traces, we assume that TOP is aware of the packets apriori. The remaining parameters are based on Table 4.1.

4.4.7 Results after comparing with competing schemes

In the following paragraphs, we will discuss different energy and deadline miss related results found after comparing our balanced schemes with different competing techniques.

1. Energy Gain In this experiment, we focus on how much energy is saved using our proposed balanced scheduling. We compare energy consumption of balanced scheduling against competing techniques, described in Section 4.4.6 in both Fast Dormancy (FD) and Tail Timer (TT) mode of operation. Fig. 4.11 shows the results for three different traffic traces. For gaming and browsing, Balanced scheduling performs similar to

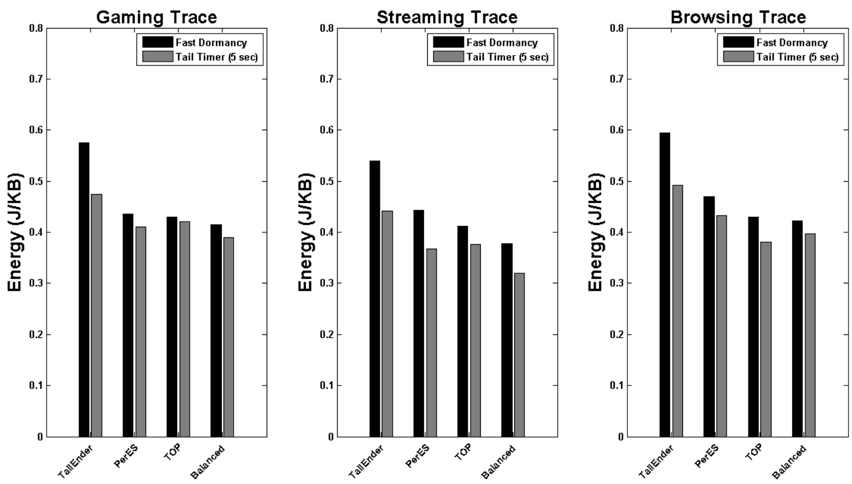


Figure 4.11: Comparing energy usage of competing scheduling strategies in Fast Dormancy and Tail Timer, with 5 sec tail time, modes. We compare three scenarios - gaming, streaming and browsing for the chosen α values.

PerES and TOP, but better than TailEnder. These two scenarios represent interactive foreground applications with relatively smaller request size, and reduced opportunity for delaying a request. Hence the gains are marginal compared to the other techniques.

Note that in streaming, as the packets are larger in size, and higher delay in transmission can be tolerated, there is more opportunity for aggregation. Therefore, Balanced Scheduling performs much better than the other techniques.

The energy consumption in TT mode is less than that of the FD mode. When network traffic is sufficiently high, then in TT mode the card transitions from $FACH$ to DCH state, thereby expending less energy compared to a transition from $IDLE$ to DCH in FD mode. The lower energy spent in each transition, and often the low number of transitions, as verified with more experimental results later, leads to better energy savings in TT mode.

2. Deadline Miss

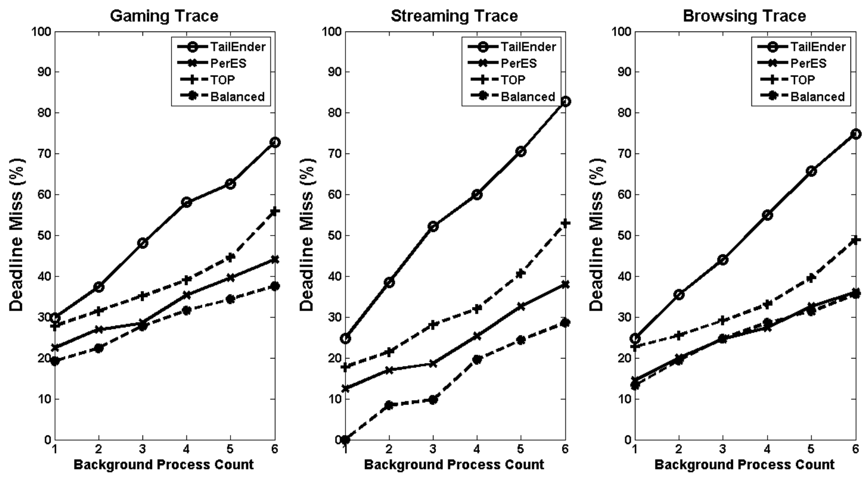


Figure 4.12: Comparison of percentage of deadline miss for competing scheduling strategies.

In our technique we introduce an application specific variable slack duration to each request. Ideally a packet should be transmitted within this slack such that user experience is not impacted. In this experiment, our goal is to study the impact of increasing number of background processes on the number of packets that miss the deadline. Lower deadline miss implies better user experience. The results record the deadlines missed by packets from foreground process since this number directly impacts the user experience. Fig. 4.12 shows the results of deadline miss percentage for all the competing techniques, and for three different scenarios, where the α values are same as the ones to obtain the energy consumption results.

There are several observations based on the results. First, Balanced scheduling performs better than the other schemes in terms of missed deadlines. PerES and TOP perform better than TailEnder since these two techniques are designed for cross application scheduling, unlike TailEnder which have been extended for cross application scenario. Second, with increasing background process count, deadline miss increases. The background processes start consuming the bandwidth although their slack duration is higher than foreground processes. This leads to more foreground requests missing their deadline. Third, in Browsing scenario PerES and Balanced scheduling has similar deadline miss, however, in other two scenarios, Balanced does better.

4.5 Real Trace based Evaluation

In this section, we firstly talk about the real trace collection methods employed and later we illustrate real trace based results.

4.5.1 App Specific Real Network Trace Collection

All the experiments in this work are done upon the synthetic data due to lack of availability of real app specific network data. To achieve that we have taken the following ways which will provide us with a rich set of app specific network data. Following are the two ways which we have explored:

Using *VpnService* in *tPacketCapture Pro* app on unrooted phone : We have used tPacketCapture Pro app which is available in Google Play to collect app specific network data in pcap file and analyzed in *WireShark*. It uses VpnService (available in Android Version $4.x$) to sniff the app specific packets and records it in a pcap file in phone's storage. The good part is that it can run on any phone but the bad part is it can not record data of multiple apps simultaneously dew to limitation in *VpnService* module.

Using netstat, tcpdump, and adb on rooted phone : On the other hand, we have also tried to collect app specific network trace data as hinted by Falaki et. al. [40] on rooted phones connected to PCs via USB. We have used android specific tcpdump implementation on ARM hardware to collect all network packets of a rooted HTC Desire phone running Android 2.3.5. We also used netstat via adb shell to get port to PID mapping on network packets and approximately able to classify app specific network data. As this method works only for rooted phones connected to PC, higher scale user data collection is not feasible. However, we have employed this method for data collection as it gives approximately accurate scenario of parallel network activities of apps.

4.5.2 Real Trace based Results

To test the practicality of our ideas, we have collected network traffic trace of 1 hour normal usage on a rooted android powered Samsung Galaxy S3 GT19300. To accomplish this, we have run tcpdump in usb connected phone and netstat tool for android through adb in laptop. By mapping the ports to process ids, we have differentiated application level traffics. For foreground apps and background apps, we have designated 5 seconds and 50 seconds of slack duration respectively. This collected network trace mostly follows the pattern of clearly separated grouped traffics of small duration requests, having 3 foreground apps and 10 background services. The grouped nature of the traffic stems from the reason that a single foreground application creates multiple connections for providing different contents to the users at a single time.

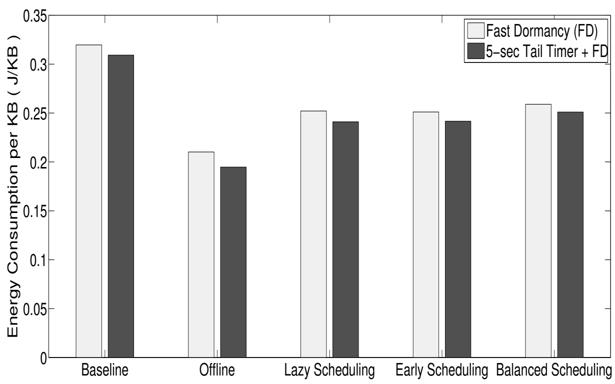


Figure 4.13: Comparing Energy Consumption of different schemes in Real Trace.

Fig. 4.13 shows that in FD or TT mode, our algorithms perform $\approx 20\%$ better than the baseline $Baseline$ system for this specific usage trace. In this scenario, energy consumption of $Baseline$ system is around 0.31 J/KB whereas our algorithms consume approximately 0.25 J/KB. As expected, Offline scheduler performs best with lowest energy footprint, i.e., around 0.2 J/KB. This gain is lesser than our simulated results due to the grouped traffic nature of the trace and lesser number of concurrent app requests, giving our techniques lesser opportunity to aggregate and to reduce switching. But, this small value will contribute to a significant energy benefit if the data transmission increases. However, deadline miss percentage or average slack duration is greater in *Baseline* compared to our techniques, which will enforce degraded user experience.

Comparing with Competing Techniques : Fig. 4.14 shows the energy usage of Balanced scheduling compared to the other techniques. Energy used by Balanced Scheduling is about 10% less than the competing techniques.

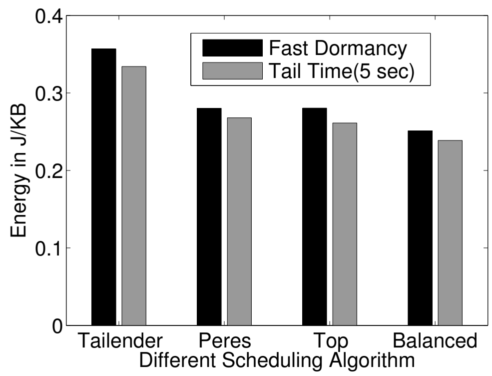


Figure 4.14: Showing energy required to transfer one KB data across different scheduling algorithms on collected real trace (browsing).

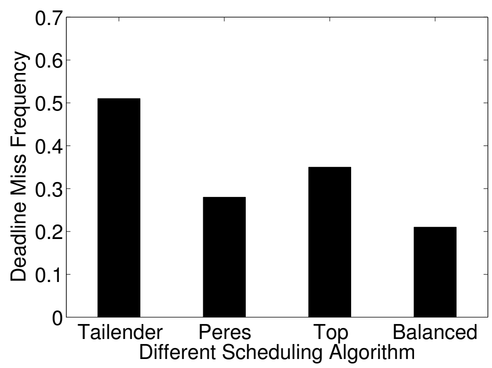


Figure 4.15: Showing deadline miss frequency across different scheduling algorithms on collected real trace (browsing).

The performance gain is lower compared to that of simulation results due to the nature of the real trace. The network traffic is grouped in nature since a foreground app spawns multiple background threads which trigger network requests at the same time. Such grouping limits the scope of aggregating traffic and reduce state transitions. Fig. 4.15 shows the corresponding deadline miss. Balanced Scheduling performs better than other techniques in this aspect.

Chapter 5

Conclusions & Future works

In this chapter, we summarize the main contributions of the thesis and outline our achievements in comparison to the objectives set up in the introductory chapter. Finally, we wrap up by pointing out some of the possible future directions of research that have been opened up by this thesis.

5.1 Summary of our contributions

In this thesis, our main contributions are the following.

1. Proposing a detailed framework for localization using *Virtual Landmarks* : We have proposed a detailed framework to find stable smartphone based virtual landmarks, which can help mainly in indoor localization. We have built a working system called *Landmarker* using that framework and have performed extensive testings in different scenarios with satisfactory results.

2. Analyzing stability of *Virtual Landmarks* : We have identified the factors that might affect the stability of *virtual landmarks* namely device, time and the phone's user. An extensive study has shown us that even though they affect the stability of landmarks, their level of impact is varied. Device heterogeneity affects the most (even when the phones are from the same manufacturer's and of the same series). Time heterogeneity also affects the stability of a landmark, which is expected as the surroundings change from time to time. However, the effect of user level heterogeneity is the least contributing factor in instability of a particular landmark. This result is assuring because modeling the heterogeneity of devices is easiest and one can build separate virtual landmark database pertaining to each class of device. On the other hand, if the result would have varied across users, identifying similar class of users and building database corresponding to each class would have been impossible.

3. Building Applications using *Virtual Landmark* based Localization Techniques : We have built a retail analytics cum shopping application in android platform called *RetailGuide* using stable virtual landmarks. We have found through various experiments that the stability of landmarks make *RetailGuide* application more robust and real-world ready. Through detailed experiments with recruited volunteers, we find not only the accuracy of the system but we also find varied amount of emotion/opinion about same products being posted by the volunteers thus bringing forward the importance of such app. Moreover, the initial idea of *SignFinder* app for creating virtual signboards seems quite promising. In future, we hope that virtual landmarks will form the basis of a suite of interesting applications.

4. Cellular Energy Consumption Reduction through Smart Scheduling of Network Traffic generated by various Applications : Research on energy optimization for 3G network card on smartphones has focused on tail time adaptation, and batch scheduling single app traffic. In this thesis, we extend the research by designing techniques for batch scheduling across multiple apps. We consider the typical smartphone usage scenario of one foreground app and multiple background services being operative and generating traffic. We introduce variable delay budget allocation per app based on user interaction, which helps in maintaining high user experience. We have proposed three deadline aware online algorithms for batch scheduling. Our simulation studies based on modeling of the overall smartphone traffic and 3G energy models show 40% energy savings in the best case and around 10% energy savings using real smartphone usage traces. Moreover, our approach has also performed competitively with other approaches of network energy savings. An important observation from this work is that better utilization of available bandwidth in $CELL\_DCH$ state leads to significant energy gains, both in Fast Dormancy and Fast Dormancy with Tail Time operating modes. The insight can augment other approaches in 3G energy optimization and help further in 4G energy optimization.

5.2 Future Directions

In this final section, we outline a few out of the many possible directions of future research that have been opened up by this thesis.

1. Augmenting *Virtual Landmark* based Indoor Localization : We have viewed virtual landmarks as stand-alone system for specifically indoor localization. However, there are also other indoor localization systems like FM-based or RFID-based as discussed in Section 2.2. So, if we intelligently include *virtual landmarks* in these localization systems, then it can give a more accurate and robust localization system. However, a detailed investigation is needed to look into the energy or deployment cost of such a system.

2. Using *Virtual Landmarks* in other Applications : We have shown the usability of virtual landmarks in localization through apps like *RetailGuide* and *SignFinder*. But, there are possibilities of numerous applications based on these smartphone based virtual landmarks. For example, in this age of Google Glass, augmented reality applications can benefit from this infrastructure-free localization system to reduce the search space of information for augmenting a particular location space. Moreover, we also can create many location based games like treasure hunt among users located at different places.

3. Developing a Network Activity Recording tool for Applications : While collecting real world usage traces from different smartphone users, we have faced a series of difficulties. There is no app in the market for unrooted phones which can help in smooth recording of application specific network activity. So, there is a great need of a good *app specific network activity* measuring tool. It will help the developers in building efficient applications and also help the researchers to properly evaluate their ideas.

4. Building a Middleware for Aggregating and Scheduling Application Requests : One can build a middleware in smartphones to aggregate the network requests across applications, which will reduce the energy footprint of applications.

5. Extending the idea of Aggregation for other resources : One can extend this idea of aggregation of different requests of similar resources to avoid redundancy and can save energy. For example, requests for energy hungry resources like GPS or sensors can be aggregated and consequently energy can be saved.

Bibliography

[1] 3G. http://en.wikipedia.org/wiki/3G.

[2] Activating Power Saving Mode in Android. http://androidwidgetcenter.com/android-tips/activating-power-saving-mode-in-android/.

[3] Apple AppStore. https://www.apple.com/in/osx/apps/app-store.html.

[4] BlackBerry service crash affects BBM messaging for millions. http://www.theguardian.com/technology/2011/oct/10/blackberry-outage-affects-bbm-services.

[5] Cloned banking app stealing usernames sneaks into Google Play. https://blog.lookout.com/blog/2014/06/24/bankmirage/.

[6] Error analysis for the Global Positioning System. http://en.wikipedia.org/wiki/Error_analysis_for_the_Global_Positioning_System.

[7] Foursquare. http://foursquare.com.

[8] FTC: Identity theft retains its throne as No.1 worst scourge in Top 10 consumer complaint list. http://www.networkworld.com/article/2163876/access-control/ftc--identity-theft-retains-its-throne-as-no-1-worst-scourge-in-top-10-consumer-compl.html.

[9] GitHub Repository of RetailGuide App. https://github.com/ananthbalashankar/landmarkerGSoC.

[10] Google Play. https://play.google.com/store?hl=en.

[11] Google Play Passes 50 Billion App Downloads. http://mashable.com/2013/07/18/google-play-50-billion-apps/.

[12] PRISM (surveillance program). http://en.wikipedia.org/wiki/PRISM_(surveillance_program).

[13] Survey shows battery life to be the single main gripe of today's mobile phone user. http://www.phonearena.com/news/Survey-shows-battery-life-to-be-the-single-main-gripe-of-todays-mobile-phone-user_id49818.

[14] WhatsApp Is Down, Confirms Server Issues. http://techcrunch.com/2014/02/22/whatsapp-is-down-facebooks-new-acquisition-confirms/.

[15] I. E. Akkus, R. Chen, M. Hardt, P. Francis, and J. Gehrke. Non-tracking web analytics. In *Proceedings of the 2012 ACM Conference on Computer and Communications Security (CCS)*, 2012.

[16] M. Anisetti and others. Advance localization of mobile terminal in cellular network. *I. J Communication, Network and System Sciences*, 2008.

[17] P. K. Athivarapu, R. Bhagwan, S. Guha, V. Navda, R. Ramjee, D. Arora, V. N. Padmanabhan, and G. Varghese. Radiojockey: Mining program execution to optimize cellular radio usage. In *Proceedings of the 18th Annual International Conference on Mobile Computing and Networking*, Mobicom, 2012.

[18] Comparing lte and 3g energy consumption. https://developer.att.com/developer/forward.jsp?passedItemId=11900006.

[19] M. Azizyan et al. Surroundsense: mobile phone localization via ambience fingerprinting. In *ACM MobiCom*, Beijing, China, 2009.

[20] N. Balasubramanian, A. Balasubramanian, and A. Venkataramani. Energy consumption in mobile phones: A measurement study and implications for network applications. In *Proceedings of the 9th ACM SIGCOMM Conference on Internet Measurement Conference*, 2009.

[21] N. Balasubramanian, A. Balasubramanian, and A. Venkataramani. Energy consumption in mobile phones: A measurement study and implications for network applications. In *Proceedings of the 9th ACM SIGCOMM Conference on Internet Measurement Conference*, IMC, 2009.

[22] A.-L. Barabasi. The origin of bursts and heavy tails in human dynamics. *Nature*, 435:207, 2005.

[23] S. Ben-david, D. Pl, and H. U. Simon. Stability of k-means clustering. In *Proceedings of the 20th Annual Conference on Learning Theory*, COLT, 2007.

[24] H. Blunck, N. O. Bouvin, T. Franke, K. Grønbæk, M. B. Kjaergaard, P. Lukowicz, and M. Wüstenberg. On heterogeneity in mobile sensing applications aiming at

representative data collection. In *Proceedings of the 2013 ACM Conference on Pervasive and Ubiquitous Computing Adjunct Publication*, UbiComp '13 Adjunct. ACM, 2013.

[25] P. Bradley and U. Fayyad. Refining initial points for k-means clustering. *International Confernce on Machine Learning*, 1998.

[26] C. Buehlmann et al. Desert ants learn vibration and magnetic landmarks. *PLoS One*, 7(3), 2012.

[27] M. Calder and M. K. Marina. Batch scheduling of recurrent applications for energy savings on mobile phones. In *SECON*. IEEE, 2010.

[28] A. Carroll and G. Heiser. An analysis of power consumption in a smartphone. In *Proceedings of the 2010 USENIX Conference on USENIX Annual Technical Conference*, USENIXATC'10, Berkeley, CA, USA, 2010.

[29] B. A. Cartwright and T. S. Collett. Landmark maps for honeybees. In *Biological Cybernetics*, 1987.

[30] A. Chakraborty, V. Navda, V. N. Padmanabhan, and R. Ramjee. Coordinating cellular background transfers using loadsense. In *Proceedings of the 19th Annual International Conference on Mobile Computing & Networking (MobiCom)*, 2013.

[31] Y. Chen, D. Lymberopoulos, J. Liu, and B. Priyantha. Fm-based indoor localization. In *Proceedings of the 10th International Conference on Mobile Systems, Applications, and Services*, ACM MobiSys '12, New York, NY, USA, 2012.

[32] K. Chintalapudi et al. Indoor localization without the pain. In *ACM MobiCom*, Chicago, Illinois, USA, 2010.

[33] Android phones steal market share. http://www.businessinsider.in/One-In-Every-5-People-In-The-World-Own-A-Smartphone-One-In-Every-17-Own-A-Tablet-CHART/articleshow/24353354.cms.

[34] S. K. Dandapat, S. Pradhan, and N. Ganguly. Offloading cellular traffic. In *Proceedings of the 5th IBM Collaborative Academia Research Exchange Workshop*, I-CARE '13, pages 4:1–4:4, New York, NY, USA, 2013. ACM.

[35] S. Dey, N. Roy, W. Xu, and S. Nelakuditi. Leveraging imperfections of sensors for fingerprinting smartphones. In *ACM Hotmobile*, Jekyll Island, Georgia, USA, 2013.

[36] N. Ding, D. Wagner, X. Chen, A. Pathak, Y. C. Hu, and A. Rice. Characterizing and modeling the impact of wireless signal strength on smartphone battery drain. In *Proceedings of the ACM SIGMETRICS/International Conference on Measurement and Modeling of Computer Systems*, SIGMETRICS, 2013.

[37] F. R. Dogar, P. Steenkiste, and K. Papagiannaki. Catnap: Exploiting high bandwidth wireless interfaces to save energy for mobile devices. In *Proceedings of the 8th International Conference on Mobile Systems, Applications, and Services*, MobiSys, 2010.

[38] H. Durrant-Whyte and T. Bailey. Simultaneous localisation and mapping (slam): Part i the essential algorithms. *IEEE Robotics and Automation Magazine*, 2006.

[39] H. Durrant-Whyte and T. Bailey. Simultaneous localisation and mapping (slam): Part i the essential algorithms. *IEEE ROBOTICS AND AUTOMATION MAGAZINE*, 2:2006, 2006.

[40] H. Falaki, D. Lymberopoulos, R. Mahajan, S. Kandula, and D. Estrin. A first look at traffic on smartphones. In *Proceedings of the 10th ACM SIGCOMM Conference on Internet Measurement*, IMC, 2010.

[41] R. Feliz et al. Pedestrian tracking using inertial sensors. *Journal of Physical Agents*, 3(1), 2005.

[42] M. R. Garey and D. S. Johnson. *Computers and Intractability; A Guide to the Theory of NP-Completeness.* W. H. Freeman & Co., 1990.

[43] Gps errors in indoor. http://gpsinformation.net/multipath.htm.

[44] S. Guha, M. Jain, and V. Padmanabhan. Koi: A Location-Privacy Platform for Smartphone Apps. In *Proceedings of the 9th Symposium on Networked Systems Design and Implementation (NSDI)*, San Jose, CA, Apr 2012.

[45] W. He, X. Liu, and M. Ren. Location cheating: A security challenge to location-based social network services. In *Distributed Computing Systems (ICDCS), 2011 31st International Conference on*, 2011.

[46] J. Huang, F. Qian, Y. Guo, Y. Zhou, Q. Xu, Z. M. Mao, S. Sen, and O. Spatscheck. An in-depth study of lte: Effect of network protocol and application behavior on performance. In *Proceedings of the ACM SIGCOMM 2013 Conference*, 2013.

[47] J. Huang, F. Qian, Z. M. Mao, S. Sen, and O. Spatscheck. Screen-off traffic characterization and optimization in 3g/4g networks. In *Proceedings of the 2012 ACM Conference on Internet Measurement Conference*, 2012.

[48] 2013 internet trends kleiner perkins caufield byers. http://www.kpcb.com/insights/2013-internet-trends.

[49] L. Jaulin. A nonlinear set membership approach for the localization and map building of underwater robots. *Trans. Rob.*, 25(1):88–98, 2009.

[50] L. Jaulin. Range-only SLAM with occupancy maps: A set-membership approach. *IEEE Transactions on Robotics*, 27(5):1004–1010, 2011.

[51] Y. Kim and J. Kim. Personalized diapause: Reducing radio energy consumption of smartphones by network-context aware dormancy predictions. In *Proceedings of the 2012 USENIX Conference on Power-Aware Computing and Systems*, HotPower, 2012.

[52] D. Kumaran and E. A. Maguire. The Human Hippocampus: Cognitive Maps or Relational Memory? *Journal of Neuroscience*, 25(31), 2005.

[53] L. I. Kuncheva and D. P. Vetrov. Evaluation of stability of k-means cluster ensembles with respect to random initialization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2006.

[54] F. Li et al. A reliable and accurate indoor localization method using phone interial sensors. In *ACM UbiComp*, Pittsburgh, PA, USA, 2012.

[55] H. Liu, Y. Zhang, and Y. Zhou. Tailtheft: Leveraging the wasted time for saving energy in cellular communications. In *Proceedings of the Sixth International Workshop on MobiArch*, MobiArch, 2011.

[56] Location aware apps on the rise. http://www.inc.com/best-industries-2011/location-aware-app-design.html.

[57] H. Lu et al. Soundsense: scalable sound sensing for people-centric applications on mobile phones. In *Proceedings of the 7th international conference on Mobile systems, applications, and services*, MobiSys '09, pages 165–178, 2009.

[58] R. Mittal, A. Kansal, and R. Chandra. Empowering developers to estimate app energy consumption. In *Proceedings of the 18th Annual International Conference on Mobile Computing and Networking*, Mobicom '12, New York, NY, USA, 2012.

[59] Mobile app internet recasts the software and services landscape. http://techie-buzz.com/mobile-news/forrester-mobile-app-internet-recasts-the-software-and-services-landscape.html.

[60] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit. Fastslam: A factored solution to the simultaneous localization and mapping problem. In *Eighteenth National Conference on Artificial Intelligence*, pages 593–598, Menlo Park, CA, USA, 2002. American Association for Artificial Intelligence.

[61] One in every 5 people in the world own a smartphone. http://www.businessinsider.in/One-In-Every-5-People-In-The-World-Own-A-Smartphone-One-In-Every-17-Own-A-Tablet-CHART/articleshow/24353354.cms.

[62] I. Papapanagiotou, E. M. Nahum, and V. Pappas. Smartphones vs. laptops: Comparing web browsing behavior and the implications for caching. In *Proceedings of the 12th ACM SIGMETRICS/PERFORMANCE Joint International Conference on Measurement and Modeling of Computer Systems*, SIGMETRICS, 2012.

[63] A. Pathak, Y. C. Hu, and M. Zhang. Where is the energy spent inside my app?: fine grained energy accounting on smartphones with eprof. In *EuroSys*, 2012.

[64] D. T. Pham et al. Selecting k in k-means clustering. *Proceedings of IMechE*, 2005.

[65] S. Pradhan, A. Balashankar, N. Ganguly, and B. Mitra. (stable) virtual landmarks: Spatial dropbox to enhance retail experience. In *2014 Sixth International Conference on Communication Systems and Networks (COMSNETS)*, pages 1–8, Jan 2014.

[66] S. Pradhan, S. K. Dandapat, N. Ganguly, B. Mitra, and P. De. Aggregating inter-app traffic to optimize cellular radio energy consumption on smartphones. In *2015 7th International Conference on Communication Systems and Networks (COMSNETS)*, pages 1–8, Jan 2015.

[67] N. B. Priyantha. The cricket indoor location system. Technical report, MIT, 2005.

[68] F. Qian, Z. Wang, Y. Gao, J. Huang, A. Gerber, Z. Mao, S. Sen, and O. Spatscheck. Periodic transfers in mobile applications: Network-wide origin, impact, and optimization. In *Proceedings of the 21st International Conference on World Wide Web*, WWW, 2012.

[69] F. Qian, Z. Wang, A. Gerber, Z. M. Mao, S. Sen, and O. Spatscheck. Characterizing radio resource allocation for 3g networks. In *Proceedings of the 10th ACM SIGCOMM Conference on Internet Measurement*, IMC, 2010.

[70] F. Qian, Z. Wang, A. Gerber, Z. M. Mao, S. Sen, and O. Spatscheck. Top: Tail optimization protocol for cellular radio resource allocation. *20th IEEE International Conference on Network Protocols (ICNP)*, 2010.

[71] L. S. Ravindranath et al. Code In The Air: Simplifying Sensing and Coordination Tasks on Smartphones. In *ACM HotMobile*, La Jolla, CA, USA, 2012.

[72] A. Schulman, V. Navda, R. Ramjee, N. Spring, P. Deshpande, C. Grunewald, K. Jain, and V. N. Padmanabhan. Bartendr: A practical approach to energy-aware cellular data scheduling. In *Proceedings of the 2010 International Conference on Mobile Computing and Networking*, 2010.

[73] I. Smith et al. Place lab: Device positioning using radio beacons in the wild. In *Pervasive*, Munich,Germany, 2005.

[74] S. P. Tarzia, P. A. Dinda, R. P. Dick, and G. Memik. Indoor localization without infrastructure using the acoustic background spectrum. In *Proceedings of the 9th International Conference on Mobile Systems, Applications, and Services*, MobiSys '11.

[75] A. Thiagarajan et al. Vtrack: accurate, energy-aware road traffic delay estimation using mobile phones. In *ACM SenSys*, Berkeley, California, 2009.

[76] C. Tossell, P. Kortum, A. Rahmati, C. Shepard, and L. Zhong. Characterizing web use on smartphones. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI, 2012.

[77] N. Vallina-Rodriguez, A. Auçinas, M. Almeida, Y. Grunenberger, K. Papagiannaki, and J. Crowcroft. Rilanalyzer: A comprehensive 3g monitor on your phone. In *Proceedings of the 2013 Conference on Internet Measurement Conference*, IMC '13, Barcelona, Spain, 2013.

[78] H. G. Walraff. *Avian Navigation: Pigeon Homing as a Paradigm*. Springer, 2005.

[79] H. Wang et al. No need to war-drive: unsupervised indoor localization. In *ACM MobiSys*, New York, NY, USA, 2012.

[80] Q. L. Xiaojun Zhu and G. Chen. Apt : Accurate outdoor pedestrian tracking with smartphones. In *IEEE INFOCOM*, Turin,Italy, 2013.

[81] C. Xu et al. Improving rf-based device-free passive localization in cluttered indoor environments through probabilistic classification methods. In *IPSN*, Beijing, China, 2012.

[82] F. Xu, Y. Liu, T. Moscibroda, R. Chandra, L. Jin, Y. Zhang, and Q. Li. Optimizing background email sync on smartphones. In *Proceeding of the 11th Annual International Conference on Mobile Systems, Applications, and Services*, MobiSys, 2013.

[83] J. Yang, A. Varshavsky, H. Liu, Y. Chen, and M. Gruteser. Accuracy characterization of cell tower localization. In *Proceedings of the 12th ACM International Conference on Ubiquitous Computing*, Ubicomp '10, New York, NY, USA, 2010. ACM.

[84] C. Yong, S. Xiao, X. Wang, M. Li, H. Wang, and Z. Lai. Performance-aware energy optimization on mobile devices. In *Proceedings of the 33rd IEEE International Conference on Computer Communications*, INFOCOM 2014, 2014.

[85] M. Youssef et al. Gac: Energy efficient gps-accelerometer-compass-gsm localization. In *IEEE GLOBECOM*, Miami,Florida,USA, 2010.

[86] M. Zhang and A. A. Sawchuk. A feature selection-based framework for human activity recognition using wearable multimodal sensors. In *BodyNets*, Beijing, China, 2011.

[87] B. Zhao, Q. Zheng, G. Cao, and S. Addepalli. Energy-aware web browsing in 3g based smartphones. In *ICDCS*. IEEE Computer Society, 2013.